

TRS-80[®]

Microcomputer News

Information published for TRS-80 users.

Word Processing With the Keiths

The Sieve of Eratosthenes

Color Text Editors

SuperSCRIPSI and

SCRIPSI 2.0 —

A Comparison



Fort Worth Scene



The May issue of 80-US contained the following article by Radio Shack's Ed Juge. Since there has been a lot of misinformation and since the policy has recently been restated to our own people, I thought you might like the information; here is Ed's May "Tandy Topics" article, reprinted with permission from 80-US:

SERVICE POLICIES

You've heard me discuss many times, the fact that we can't repair modified TRS-80s . . . we reserve the right to void your warranty if a unit has been opened or modified . . . we can't be responsible for non-Radio Shack parts installed, etc. Officially, our policies have been pretty much as stated. Unofficially, we've allowed our service center personnel to determine when special cases warranted special handling. We ended up with some inconsistencies, though, where one repair facility would agree to work on a unit after another facility had refused. The good news is that we think we've learned quite a bit about what we can do effectively, and what we can't.

So about the first of March, a letter went out to all of our Service Centers, giving them specific guidelines on what to do with TRS-80's which aren't completely "stock." And since it will undoubtedly make a lot of you very happy . . . and since I think it shows a real effort on the part of our Support Operations management to increase our level of service, I'd like to tell you about these guidelines.

Up front, let me say that there are some "maximum parts/labor charges" for certain completely stock equipment. Those charges will not apply to units which are "non-stock" in any way. With that caveat, we'll proceed . . .

First of all, let's look at the case of RAM chips. Those of you with "stock" TRS-80's containing Radio Shack chips . . . we will replace only the defective chips we find, not a whole set. If you've installed "foreign" RAMs, and we find one or more of them defective, the service center is still required to replace a complete set or none. They will, however, call and give you the option of doing it yourself or having us do it. There will be a checkout fee charged for our time, even though no service is performed.

FOREIGN DISK DRIVES

We have said many times that we can't work on Model III's which contain non-Radio Shack drives. Well, our folks tell us that they will accept such units for service . . . partly! What that means is that if the drives work properly, we will simply overlook them. If the drives are at fault, we will disconnect them, check or repair the "stock TRS-80" portion of your computer, and let you handle any drive problem with the vendor or the dealer who installed them.

I still issue a strong warning to the non-technical reader: This leaves you in the dangerous position I've described before, that Radio Shack will guarantee the operation of our equipment, as the drive vendor may guar-

antee his drives. But you'd better be sure someone guarantees that they will work together, unless you have the knowledge and equipment to do it yourself. Now my sales pitch . . . a few freight bills or "figure it out as we go" repair bills by a third party can more than eat up your savings.

MODIFICATIONS

This one's a bit harder. If a modification doesn't appear to be hurting anything, we'll disregard it. If we can't properly check out the unit, or if the modification seems to be causing the problem, we'll ask your permission to remove it. If you decline, you'll pay only the checkout fee. Otherwise, we'll remove it, return the parts to you, and effect repairs.

Now, there is one big "gotcha" . . . when there are "cuts and jumpers," or damage to the board such that our repair people back here in Fort Worth wouldn't accept it for rebuilding on an exchange basis. If your computer qualifies, then your only options may be to repair it yourself, or authorize us to install a new circuit board on a non-exchange price basis.

Our repair people will always return any non-Radio Shack parts, whether you request them or not. We've always done this if you asked when you brought the unit to us, but some of you asked only when you picked your computer up . . . after the parts had been thrown away or mixed up with other defectives to the point they could not be identified.

There are a couple of other clauses in the new policy that you should be aware of. If our people feel that the computer poses any kind of safety hazard to the service technician, they may refuse to service it at all. And our after-service warranty will specifically exclude the operation of any non-Radio Shack parts or modifications which we leave installed. Also, TRS-80 equipment containing non-Radio Shack parts or modifications cannot be covered by our Maintenance Agreements.

So having opened, modified, or installed your own parts no longer automatically excludes Radio Shack service on your TRS-80. It can, however, cause that service to cost you more; and a trip to our Repair Center will not necessarily assure a working computer if your non-stock parts are faulty.

* * * * *

DATE CHANGE

As you may have noticed, this issue of TRS-80 Microcomputer News is dated July/August rather than just July. This change was made to make our dating method conform more closely with your expectations. We have received several complaints that the News always arrives at the end of the month rather than the beginning. This change will result in the issues getting to you at the beginning of the month they are dated.

This change will not affect the actual number of issues (free or paid) which you will receive (this issue counts as one issue, not two). Our mailing list system counts the number of issues we send you, without regard for when they are sent. So, if you are entitled to six free issues, you will still get six free issues. If you have paid for twelve issues, you will get twelve issues.

TRS-80[®] Microcomputer News

Information Published for TRS-80 Users

Reproduction or use, without express written permission from Tandy Corporation of any portion of the Microcomputer News is prohibited. Permission is specifically granted to individuals to use or reproduce material for their personal, non-commercial use. Reprint permission for all material (other than William Barden's article), with notice of source, is also specifically granted to non-profit clubs, organizations, educational institutions, and newsletters.

TRS-80 Microcomputer News is published monthly by Radio Shack, a division of Tandy Corporation. A single six month subscription is available free to purchasers of new TRS-80 Microcomputer systems with addresses in the United States, Puerto Rico, Canada and APO or IPO addresses. Subscriptions to other addresses are not available.

The subscription rate for renewals and other interested persons with U.S., APO or FPO addresses is twelve dollars (\$12.00) per year, check or money order. Single copies of the Microcomputer News may be purchased from Radio Shack Computer Centers or Computer Departments for \$1.50 suggested retail each. The subscription rate for renewals and other interested persons with Canadian addresses is Fifteen dollars (\$15.00) per year, check or money order in U.S. funds. All correspondence related to subscriptions should be sent to: Microcomputer News, P.O. Box 2910, Fort Worth, Texas 76113-2910.

Retail Prices in this newsletter may vary at individual stores and dealers. The company cannot be liable for pictorial and typographical inaccuracies.

Back issues of Microcomputer News prior to January, 1981 are available through your local Radio Shack store as stock number 26-2115 (Suggested Retail Price \$4.95 for the set). Back issues of 1981 copies are not available.

The TRS-80 Newsletter welcomes the receipt of computer programs, or other material which you would like to make available to users of TRS-80 Microcomputer systems. In order for us to reprint your submission, you must specifically request that your material be considered for reprinting in the newsletter and provide no notice that you retain copyrights or other exclusive rights in the material. This assures that our readers may be permitted to recopy and use your material without creating any legal hassles.

Material for publication should be submitted on magnetic media (tape, disk, or CompuServe). If you submit material on tape or disk, and it is accepted for publication, we will send you two cassettes or diskettes for each one you sent us. Cassettes will come from our box of mixed blank cassettes. If you submit material on CompuServe, and we think we may use the material, we will extend your Microcomputer News subscription by six months for each article accepted.

If you are submitting material over CompuServe, please include your name and address or your subscription number so we can find you.

Material may be submitted by mail to P.O. Box 2910, Fort Worth, Texas 76113-2910, or through CompuServe. The Microcomputer News CompuServe user ID number is 70007.535.

Notes to Program Users:

Programs published in the Microcomputer News are provided as is, for your information. While we make reasonable efforts to ensure that the programs we publish here work as specified, Radio Shack can not assume any liability for the accuracy either of the programs themselves, or of the results provided by the programs.

Further, while Microcomputer News is a product of Radio Shack, the programs and much of the information published here are not Radio Shack products, and as such can not be supported by our Computer Customer Service group. If you have questions about a program in the Microcomputer News, your first option is to write directly to the author of the program. When possible, we are now including author's addresses to facilitate communications. If the address is not published, or if you are not happy with the response you get, please write us here at Microcomputer News.

Comments on Our Program Listing Style:

In order to make the program listings we publish easier to read, we have adopted a style of inserting spaces to enhance readability, and we separate each program statement onto a separate line. While these techniques increase program readability, they also require more memory, and may execute more slowly than the original program did.

Trademark Credits

CompuServe*	CompuServe, Inc.
Dow Jones**	
NEWS RETRIEVAL	
Service*	Dow Jones & Co., Inc.
Time Manager**	Image Producers
VisiCalc*	VisiCorp, Inc.
Program Pak**	Tandy Corporation
SCRIPSIT*	Tandy Corporation
TRSDOS**	Tandy Corporation
TRS-80*	Tandy Corporation

Contents:		
Assembly Language Programming	8
by William Barden, Jr.		
Color Computer		
Bugs, Errors, and Fixes		
Scripsit (26-3105)	46	Spectaculator (26-3104) 46
Programs		
Asteroids by Richard Zepp	46	Super Text by Matthew McGovern 43
Fire Works by Franklin Pigounias	47	Text Editor by Brian Lievalen 43
Scripsit and the LP VIII		
by Garrett McGowan	45	
Computer Customer Service		12
<i>More about Scripsit 2.0</i>		
Data Bases		
CompuServe	14
New Menu System		
Racing Information	15
Dow Jones	16
Movie Reviews		
Profile	19
Merging Profile with Scripsit		
VisiCalc	23
Additional Use of a LOOKUP by Lawrence Scanlon		
Educational Products		
Compress Graphics Using DEBUG by Dennis Tanner	34
Computers for Neophytes	15
Network 3 — New Capabilities for the Classroom	32
Feature Story		5
<i>Word Processing with the Keiths</i>		
Fort Worth Scene		2
General Interest		
Micro Research Problem	17
Model I/III		
Bugs, Errors, and Fixes		
Accounts Payable (26-1554)	30	Disk Scripsit (26-1563) 30
Accounts Receivable (26-1555)	30	Manufacturing Inventory
Advanced Statistical Analysis (26-1705)	31	Control (26-1559) 30
Cassette Portfolio (26-1506)	29	Medical Office Systems (26-1568) 30
COBOL (26-2203)	31	Stockpak (26-1507) 30
Disk Payroll (26-1556)	30	
Product Line Manager's Page	25
<i>SuperSCRIPSIT and SCRIPSIT 2.0—A Comparison</i>		
Programs		
Poor Man's Editor by Dallas Powell	35
Model II		
Bugs, Errors, and Fixes		
Accounts Payable (26-4505)	38	Statistical Analysis (26-4540) 39
Inventory Control (26-4502)	38	TRSDOS (26-4910) 39
Scripsit 2.0 (26-4531)	38	TRSDOS-HD (26-4150) 38
Product Line Manager's Page	36
<i>Scripsit Ideas (XERCH by Peter Beeson, Underscore by J.G. Lowder)</i>		
Medical Office System	37
Programs		
More Scripsit Ideas by C. M. Mendenhall	37
More Computer Clubs		15
Notes on Previous Newsletters		
March 1982	22
Peripherals		
Product Line Manager's Page	
Using the Digitizer to Calculate Area Measurement and Curve Integration	18
Pocket Computer		
Programs		
Convert 35MM to 16MM by Robert Edwards	11
Cost Comparison by Vernon Scally	41
Factorials by Thomas Cox	41
Measurement Conversion by Frederic Bishop	40
Mileage Calculations by Alan Leiwant	42
MPG on PC by Bruce Taub	42
Projectile Trajectory by Larry Mermelstein	40
Resistor Values by Robert Edwards	41
View From the 7th Floor by Jon Shirley	4

Prices shown in TRS-80 MICROCOMPUTER NEWS are in U.S. Funds.

View from the 7th Floor

by Jon Shirley,
Vice President,
Radio Shack Computer Merchandising

I am told that this is a word processing issue. I am a big user of Scripsit both at home and in the office, and if any of you readers have a TRS-80 without any word processing, you are missing a great tool. (All right, all you Pocket Computer owners, I know there is no Scripsit for you.) With the issuance of Scripsit for the Color Computer all of you can now enjoy WP. If you have not ever tried WP, run to our nearest store and try a little hands-on.

At its most simple, WP will let you hack away at the keyboard with one or two fingers, make lots of errors and still produce a perfect printed document. At its most complex (which is still easy to use), it will let you create form letters, fill in the blanks automatically, and print them out unattended. WP is absolutely the best thing that ever happened since the typewriter; may it fade away.

One of the most useful features of Model II Scripsit is user-defined keys. This lets you define a key stroke sequence (escape 1 through 0 and shifted 1 through 0) to do anything that you could do from the keyboard. I use this feature a great deal to hold words or phrases that I use often. For example, "Radio Shack TRS-80 Model II" is escape 3 on some of my disks. But you can also store commands in the user key sequence. For example, a user key could move the cursor a certain number of spaces or insert blanks.

The user key can insert both text and commands, so a key could insert an often-used word and the commands to make it underlined. The keys can also call other user keys to nest operations. One example of this use would be to do a phrase built up of smaller phrases stored in other keys. Each key is limited to 255 keystrokes.

One application for the user key is for the secretary who handles a lot of correspondence. A single user key could hold a sequence such as **(ENTER) (ENTER) (ENTER) (TAB) Sincerely, (ENTER) (ENTER) (TAB) Mr. Phil Alphabet (ENTER) (TAB) Vice President (ENTER)**. That sequence will finish a letter. The sequence could be longer and finish with the initials of the secretary at the lower left of the letter. If one individual does letters for several different people each one would be assigned a different user key.

Another application is in editing a letter. If you had something to insert in a lot of places that could not be accessed by the global search command, you could have a sequence that invoked insert, put in the new text and a space, then hit delete to end the sequence. All this is pretty simple stuff. Experimentation can lead you to some very complex routines.

OFFICE AUTOMATION SHOW AND CONFERENCE

I recently attended the Office Automation Show and Conference in San Francisco. All the big names were there all with their answers for the computerized, paper-less office of the future. They all seemed to be offering the same stuff, and they all seemed to have the same problem—their stuff is not selling. Offices are getting automation, but it's personal computers that are doing the automating. The reason for this is that personal computers are relatively cheap, that they are reasonably user friendly, and that there is enough good software to make them cost justified.

It's a little crazy to see all those big companies with their XXXnet (everyone has one), when the hookup alone costs more than most microcomputers. To make matters worse, some require workstations that sell for over \$15,000 and printers that sell at over \$40,000. So I will go out on a limb and predict that the revolution in the office will arrive through the back door with zillions of little computers sneaking in while the data processing department tries to decide which "net" to buy that won't be obsolete next year.

A GLIMPSE OF THE FUTURE?

I did see one really neat product and while I usually do not talk about other equipment, I will pass this one on to you. It is a small portable computer with all this: 16 bit processor, 256K RAM, 256K bubble memory, 320x240 flat screen display, nice 57-key keyboard, built-in 300/1200 baud direct connect auto answer/auto dial modem, and IEEE 488, RS232 interfaces. When I said small I meant it, 15" by 11 1/2" by 2"! (Yes, really, 2" thick). Is it a dream? No, it's very real and has only one small drawback; it's about \$8,500 without software. I must admit that this is the first computer of another brand that I really wanted to buy. I have no idea why, other than it looked so neat. However . . . at that price I think I'd better wait.

What this little product does show is the future. Flat screen displays with speed and resolution will come down in cost. I do not think bubble memory will, but CMOS RAM will and solve that problem. One thing about this business, it is not boring. Evolution and revolution are the name of the game, and we can all look forward to some real science fiction stuff in the years ahead. Just think about having VisiCalc on your 32 bit wrist watch computer. On the other hand, my old Model I has gone well over a year now with no failures and no CRC errors, despite its frequent use so I guess I can wait a little longer for the future.

Until next month.

Word Processing with the Keiths

Kathleen Keith of Rineyville, Kentucky, is a professional writer who thinks that Scripsit is fantastic. There are two signs in her computer room. The first hangs over a souvenir Bible page that she obtained in 1955 on the 500th anniversary of Gutenberg's development of movable type. It says:

**THE FIRST BREAKTHROUGH FOR LITERACY
AND PRINTING — MOVABLE TYPE**

The second sign, which hangs over her 64K two disk Model II and Daisy Wheel printer, says:

**THE SECOND BREAKTHROUGH FOR LITERACY
AND WRITING — WORD PROCESSOR**

HER WRITING PROCEDURE

"As a writer I have been following this procedure in using the word processor. I CREATE a document and type this episode for the six to ten pages I can do that day, usually stopping in the middle of some action. When I am finished for the day I COPY it onto another diskette and I also PRINT it out on the Daisy Wheel II single spaced, to file away. I guess writers are used to the hard copy and can't yet do without it. The next day I go back through what I had typed and correct all the typographical and spelling errors and grammatical boo-boos and continue writing.

"Many of the publishers say that because of increasing inflation, the money offered to writers is being whittled down. With the opportunities provided by the marvels of the data that the writer has put onto the magnetic disk, this can eliminate a highly paid human typesetter, and the necessity for me to proofread the galley proofs that he or she has produced . . ."

WHAT SCRIPSIT HAS MEANT TO ME — A SUMMARY

"Not many of my writer friends around here have heard of word processors for writers. At a recent writers conference held in Kentucky, I offered this summary of what my word processor has meant to me.

"This spring my husband and I had returned from an extensive trip through Mexico gathering background material for me to prepare my novel on the culture of the Mayan peoples before the Spanish came. In my research I had read a number of theses on the subject. I found that many times in the first pages of the thesis, after giving recognition to all the experts who had assisted the writer, there were many dedication pages which read in essence: 'To my Wife, the Typist.' Well, I told the group that I haven't got a wife to be a typist. (Laughter from the audience) While I don't mind typing the first draft of my work in the flush of creation, what I hate is retyping everything after I've edited it, especially the good parts that really didn't need any reworking. (General nods of agreement) I told them that it was only after considerable soul searching that we finally

decided to get the Daisy Wheel II Printer despite the additional cost. This was simply because it was the only printer that produced typewriter quality print. This is absolutely necessary to have in order to present a salable package to a publisher. Our philosophy of purchasing the entire word processing system was that no one would blink an eye if we purchased a new car today. The word processing system (computer and all!) cost less than a new car so I am continuing to drive my car with 111,000 miles on it.

"I went on to explain that my thought processes have been freed from thinking about the mechanics of what's happening when I type, and my fingers can almost keep up with what I am thinking. My husband says I type three times as fast as the electric typewriter and it's reassuring not to hear that typewriter noisily banging away on the paper. Trying to explain why I needed the additional disc drive to copy the discs was more difficult but everyone was curious about the whole word processing idea."

KATHY KEITH'S COMMANDMENTS FOR A HOME COMPUTER ROOM

- I Thou shalt not set thy cup of coffee on thy Computer Desk nor any glass with any liquid in it.
- II Thou shalt not bring anything with a magnetic field into thy Computer Room. Thy bulk eraser shall be kept in another room or 50 feet away from all disks or computer.
- III Thou shalt make two BACKUP copies of any software thou dost purchase before thou shalt start to play with it.
- IV Thou shalt not press the Reset switch whilst thou hast a SCRIPSIT document open in thy computer's drive.
- V Thou shalt not let thy cat take a nap with thee in thy computer room, in particular if the cat hast longhair.
- VI Thou shalt keep thy windows and door closed for temperature and humidity control. Thou shalt vacuum thy computer room regularly and immediately spray sensitive areas with anti-static spray. (Nylon rug and video screen.)
- VII Thou shalt take all measures to keep dust out of thy computer, vacuuming out floor vents for thy heating system, including return air vents. Thou shalt examine the filter on thy furnace and replace at the slightest shadow. And if thou hast a wood burning stove in thy house, thou shalt filter incoming air to thy computer room and use such additional mechanical devices as an air cleaner near thy computer's air vents.
- VIII Thou shalt not let any Ham operate his Amateur Radio Walkie Talkie in thy computer room.
- IX Thou shalt cover thy computer and printer with dust covers when not in use, every day!
- X Thou shalt not hit the switch with thy knee on thine additional disk drive, for if thou dost turn it off, thy disk

will be afflicted with glitches from which it cannot be cured.

A TWO COMPUTER FAMILY!

A few weeks after we received Mrs. Keith's letter her equally ardent TRS-80 user husband, Gilbert Keith, amused and delighted us with the following.

HAPPINESS IS HAVING YOUR OWN PRINTER

"Happy day when the little LP VIII arrived. Waiting for it had been a trauma and an eternity. The joy and pleasure it has given me since then is not counted in dollars and cents, as the pleasure it affords me and the fact it talks to me is worth all the anxious days spent in waiting for it.

"After my wife Kathy had installed her Model II, I knew I just had to have the Model III so I started looking for a printer. Checking out all the different ones on the market was a chore. None seemed to be just what I wanted. Maybe being spoiled by Kathy's Daisy Wheel II did it. The only solution was to purchase the LP VII and wait to see what would be on the market later that I could be happy with. With Scripsit, Mailing List and VisiCalc, the limitations of the LP VII became apparent as the days passed by."

ANOTHER ADDITION TO THE FAMILY

"I don't know how I missed the LP VIII in all of my checking out of the printers. At the Radcliff, Kentucky, Radio Shack store one day, the LP VIII was called to my attention and you know what? The order was placed that day. Just reading about it in the catalog was enough to convince me. I later found out my gut feeling was right. Two days later the Line Printer VII was sold, leaving me without a printer. I was in misery. Being told the waiting time for the LP VIII was to be one month, I called several stores, even the Tandy store in Ft. Worth, to see if a LP VIII was available. One day, at the local store I saw the UPS truck stopping and I went out to see what was in it. I was ecstatic as there were three LP VIII's on the truck designated for the local store. With no further ado I grabbed one, and with coattails flying, headed for my car. 'BABY' had found a new home."

AN INCOMPATIBLE BABY?

"First, I'm not a programmer and understand very little about DOS and BASIC. But I have spent many happy hours seeing what I could do with 'BABY.' My main use for the computer is Scripsit. When the local computer center in Louisville was asked if 'Baby's' different type styles could be used with Scripsit, I was informed that they had not had time to check it out yet. The more I thought about it, the more it bugged me. One night while watching a movie on TV, my mind was in a whirl. There had to be a way! Surely Scripsit and 'BABY' were compatible. Suddenly, it came to me and I left Kathy watching the movie. I rushed to my computer room and started to work.

"It wasn't long before my old letters and writings were coming in all different shapes and forms. I was able to print two columns on one paper. Weird, same letter on the same page side by side. I'm sure SuperSCRIPSIT will be able to do all this and much more. (Editor's note: How right he is!)

"I will show the way it is accomplished. Now all I have to do is to wait for SuperSCRIPSIT and the dictionary. It will not be as hard, as I have the 'best' computer and printer I could ever want. . . ."

Using Scripsit And The LP VIII Type Styles

Gilbert Keith
Rineyville, Kentucky

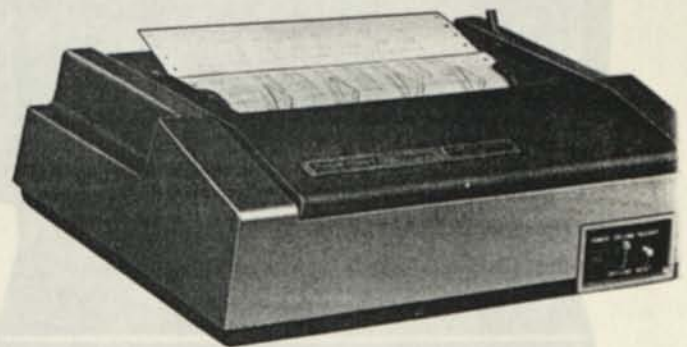
First, we will use the SCRIPSIT program entirely. After you have written and saved your document on Scripsit decide what style of type you want to print it in. There are four you can use (Monospaced, Proportional, Condensed, Monospaced, Elongated).

Boot up the SCRIPSIT diskette and at TRSDOS Ready type BASIC (ENTER). When READY appears, type the following:

```
10 LPRINT CHR$(27);CHR$(N);"!"
```

N is the number of the type you want. Then press (ENTER). With the printer on-line, type RUN. The printer should make a little grunt and do nothing. Next type CMD"S" and press (ENTER). You should be back to TRSDOS. Type: SCRIPSIT (ENTER) and you are ready to load any document you want to print. Remember, you must set the format line for the type style you are printing. The "LM" and the "RM" must be set. For printing condense with two columns on a page, first I set the format line like this:)PL=66,LM=15,TM=3,BM=60,RM=60 (ENTER. For printing, I roll back the paper to the top of the page and reformat thus)PL=66,LM=70,TM=3,BM=60,RM=120. When printed you should have two columns on the same page. Try this out on some scratch paper and when you find out the right format line, SAVE it for that type of printer. You should have a lot of fun experimenting with SCRIPSIT and the LP VIII. I know I have . . .

```
10 LPRINT CHR$(27);(20);"!"
30 LPRINT CHR$(27);CHR$(14);"LARGE"
50 LPRINT CHR$(27);CHR$(15);"SMALL"
70 LPRINT CHR$(27);CHR$(17);"PROPORTIONAL"
90 LPRINT CHR$(27);CHR$(20);"CONDENSED"
110 LPRINT CHR$(27);CHR$(19);"ORDINARY"
130 LPRINT CHR$(08);CHR$(143);"BOLD LETTERS"
150 LPRINT CHR$(15);"UNDERLINE"
170 LPRINT CHR$(14);"WITHOUT UNDERLINE"
190 LPRINT "BOLD LETTERS"; CHR$(08); CHR$(143)
    "BOLD LETTERS"
```



Kathleen Keith's Scripsit Backup System

These four steps are required to set up an effective Word Processing System to give peace of mind to users and insure against loss of documents along the way.

This set of 14 (or 16) disks will be enough Backup copies of SCRIPSIT and the Dictionary. Give Scripsit disks a name for their intended purpose.

A writer uses 2 disks called NOVEL. Day to day LETTERS are a second set. (It may also be necessary to create a set of BUSINESS disks).

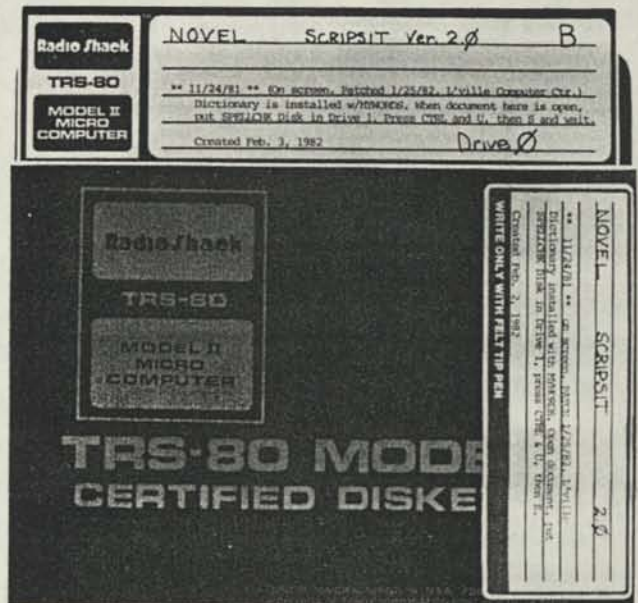
4. When Dictionary (now called SPELLCHK) has been installed on NOVEL Scripsit disk B, having chosen the MYWORDS Option, make a Backup of this NOVEL disk. (Before wiping NOVEL A, check that all documents are copied onto NOVEL B. Then it can be reused in this system.)

Label SPELLCHK disk to be used with NOVEL Scripsit ONLY.

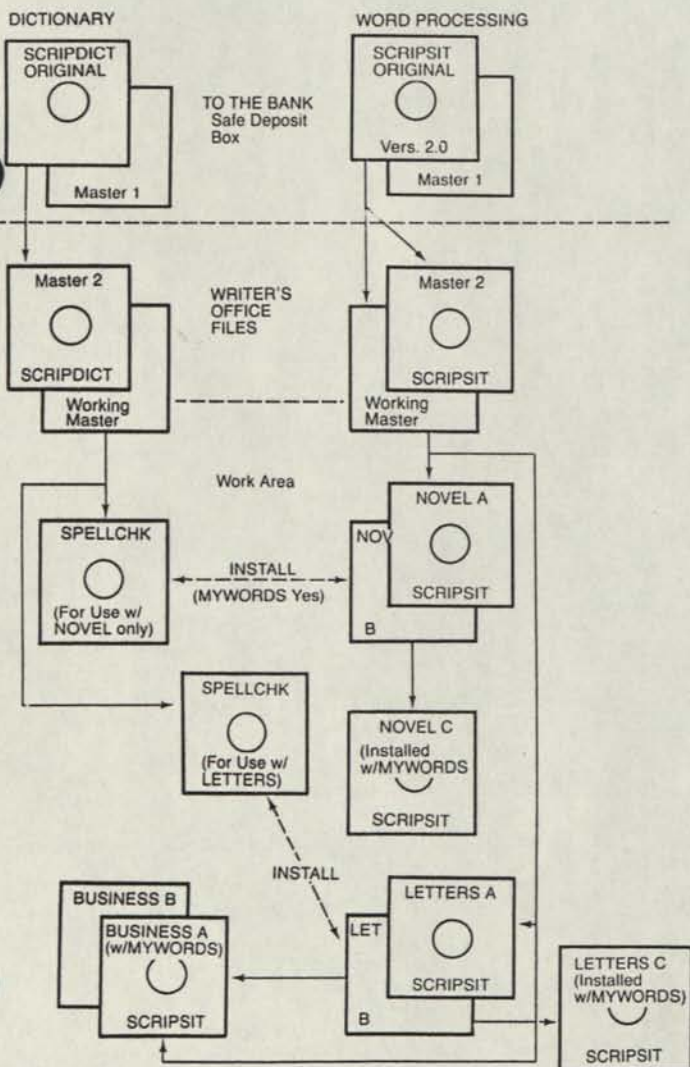
Make separate SPELLCHK Dictionary for use with LETTERS Scripsit disks. Make backup of LETTERS B, and check that all documents on LETTERS A disk have been copied onto LETTERS B.

If deemed necessary, make up Scripsit disks for BUSINESS also, but SPELLCHK Dictionary for LETTERS can be used with it.

Label all disks carefully!



A set of disks are put into a soft-sided binder to protect them. This binder includes two SCRIPSIT disks which were named NOVEL when they were backed up but carry written labels of A and B also. The other disk in this binder is the Spelling and Hyphenation Dictionary called SPELLCHK which was installed on a NOVEL Scripsit Disk. This SPELLCHK will have the MYWORDS Option which will include the words unique to this Novel. (These words include geographic names and foreign words.)



will be afflicted with glitches from which it cannot be cured.

A TWO COMPUTER FAMILY!

A few weeks after we received Mrs. Keith's letter her equally ardent TRS-80 user husband, Gilbert Keith, amused and delighted us with the following.

HAPPINESS IS HAVING YOUR OWN PRINTER

"Happy day when the little LP VIII arrived. Waiting for it had been a trauma and an eternity. The joy and pleasure it has given me since then is not counted in dollars and cents, as the pleasure it affords me and the fact it talks to me is worth all the anxious days spent in waiting for it.

"After my wife Kathy had installed her Model II, I knew I just had to have the Model III so I started looking for a printer. Checking out all the different ones on the market was a chore. None seemed to be just what I wanted. Maybe being spoiled by Kathy's Daisy Wheel II did it. The only solution was to purchase the LP VII and wait to see what would be on the market later that I could be happy with. With Scripsit, Mailing List and VisiCalc, the limitations of the LP VII became apparent as the days passed by."

ANOTHER ADDITION TO THE FAMILY

"I don't know how I missed the LP VIII in all of my checking out of the printers. At the Radcliff, Kentucky, Radio Shack store one day, the LP VIII was called to my attention and you know what? The order was placed that day. Just reading about it in the catalog was enough to convince me. I later found out my gut feeling was right. Two days later the Line Printer VII was sold, leaving me without a printer. I was in misery. Being told the waiting time for the LP VIII was to be one month, I called several stores, even the Tandy store in Ft. Worth, to see if a LP VIII was available. One day, at the local store I saw the UPS truck stopping and I went out to see what was in it. I was ecstatic as there were three LP VIII's on the truck designated for the local store. With no further ado I grabbed one, and with coattails flying, headed for my car. 'BABY' had found a new home."

AN INCOMPATIBLE BABY?

"First, I'm not a programmer and understand very little about DOS and BASIC. But I have spent many happy hours seeing what I could do with 'BABY.' My main use for the computer is Scripsit. When the local computer center in Louisville was asked if 'Baby's' different type styles could be used with Scripsit, I was informed that they had not had time to check it out yet. The more I thought about it, the more it bugged me. One night while watching a movie on TV, my mind was in a whirl. There had to be a way! Surely Scripsit and 'BABY' were compatible. Suddenly, it came to me and I left Kathy watching the movie. I rushed to my computer room and started to work.

"It wasn't long before my old letters and writings were coming in all different shapes and forms. I was able to print two columns on one paper. Weird, same letter on the same page side by side. I'm sure SuperSCRIPSIT will be able to do all this and much more. (Editor's note: How right he is!)

"I will show the way it is accomplished. Now all I have to do is to wait for SuperSCRIPSIT and the dictionary. It will not be as hard, as I have the 'best' computer and printer I could ever want. . . ."

Using Scripsit And The LP VIII Type Styles

Gilbert Keith
Rineyville, Kentucky

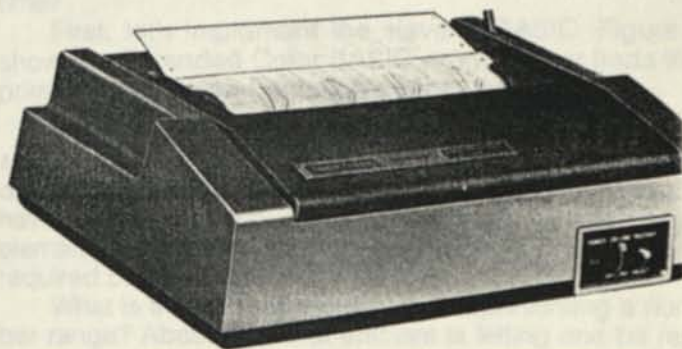
First, we will use the SCRIPSIT program entirely. After you have written and saved your document on Scripsit, decide what style of type you want to print it in. There are four you can use (Monospaced, Proportional, Condensed Monospaced, Elongated).

Boot up the SCRIPSIT diskette and at TRSDOS Ready type BASIC (ENTER). When READY appears, type the following:

```
10 LPRINT CHR$(27);CHR$(N);"!"
```

N is the number of the type you want. Then press (ENTER). With the printer on-line, type RUN. The printer should make a little grunt and do nothing. Next type CMD"S" and press (ENTER). You should be back to TRSDOS. Type: SCRIPSIT (ENTER) and you are ready to load any document you want to print. Remember, you must set the format line for the type style you are printing. The "LM" and the "RM" must be set. For printing condensed with two columns on a page, first I set the format line like this:)PL=66,LM=15,TM=3,BM=60,RM=60 (ENTER). For printing, I roll back the paper to the top of the page and reformat thus)PL=66,LM=70,TM=3,BM=60,RM=120. When printed you should have two columns on the same page. Try this out on some scratch paper and when you find out the right format line, SAVE it for that type of print. You should have a lot of fun experimenting with SCRIPSIT and the LP VIII. I know I have . . .

```
10 LPRINT CHR$(27);(20);"! "  
30 LPRINT CHR$(27);CHR$(14);"LARGE "  
50 LPRINT CHR $(27);CHR$(15);"SMALL "  
70 LPRINT CHR$(27);CHR$(17);"PROPORTIONAL "  
90 LPRINT CHR$(27);CHR$(20);"CONDENSED "  
110 LPRINT CHR$(27);CHR$(19);"ORDINARY "  
130 LPRINT CHR$(08);CHR$(143);"BOLD LETTERS "  
150 LPRINT CHR$(15);"UNDERLINE "  
170 LPRINT CHR$(14);"WITHOUT UNDERLINE "  
190 LPRINT "BOLD LETTERS"; CHR$(08); CHR$(143)  
"BOLD LETTERS"
```



Sieve of Eratosthenes

by William Barden, Jr.
© William Barden, Jr. 1982

What do the ancient Greeks, a hack computer writer, and the Radio Shack Color Computer have in common? Give it some thought before answering, and please, no bad jokes about "Euripides? I mend-a these . . .".

Oh, oh. There's the buzzer. I'm sorry, Mr. Smith, you lost the chance for a fun-filled weekend in Fort Worth, but we're giving you two fun-filled weekends as a consolation prize . . . The answer to the question, of course, is prime numbers.

Now wait. I know you might not like mathematics. However, you don't need to know much about math for this topic, and it has some fascinating aspects. We're going to use a 2000-year old algorithm to generate prime numbers in the wink of an eye by assembly-language on the Color Computer. Furthermore, we'll use the high-resolution graphics of the CC to display the numbers. Interested? Read on . . .

THE PRIME SUBJECT

Prime numbers have interested numerologists through the ages. A prime number is any number that can't be divided evenly by any number except itself and the number one. For example, 5 is a prime number, because 2, 3, or 4 can't be divided into 5 without remainder; only 5 and 1 can be divided into 5 evenly. The number 11 is also a prime number; divisors of 2, 3, 4, 5, 6, 7, 8, 9, and 10 result in remainders, and numbers over 11 result in fractions. The number 12 is not prime because it can be divided evenly by 2, 3, 4, or 6.

The first few prime numbers are 1, 2, 3, 5, 7, and 11.

How would you go about finding all the prime numbers up to a given number? There's been a great deal of recent research into prime number theory — such topics as the total number of primes in a given range of numbers, tests for primes, and special classes of prime numbers, such as "Mersenne primes." One old, but effective method for finding primes, however, was developed over 2,000 years ago by Eratosthenes, a Greek mathematician and astronomer (who, among other things, accurately calculated the size of the moon!).

THE "SIEVE OF ERATOSTHENES"

Eratosthenes' method is shown in Figure 1. All the numbers to infinity are listed in a row. (I gave Bruce an infinitely long list, but he may cut it down to manageable proportions for the newsletter . . .)

Now a "multiplicand," starting with 2, is multiplied by 2, 3, 4, 5, 6, and all integral multipliers. The result is "struck out" of the list of all numbers. In the case of a multiplicand of 2, for example, 4, 6, 8, 10, 12, and so forth would be

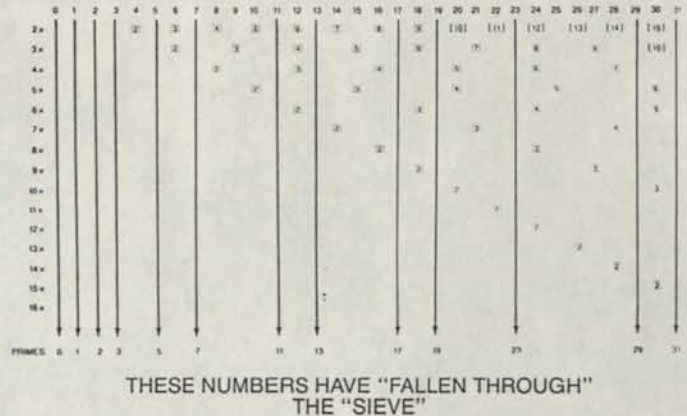


Figure 1. The "Sieve of Eratosthenes"

struck out or deleted from the list of all numbers. This process is repeated for multiples of 3 (6, 9, 12, etc.), multiple of 4 (8, 12, 16, 20, etc.), multiples of 5 (10, 15, 20, 25, etc.) and all numbers up to the last number in the list.

After the multiplicand has been processed, the numbers remaining in the list "fall through" the "sieve" and are primes, as all multiples have been considered.

This is a "brute force" method, as many numbers are struck out more than once. The number 24, for example, is deleted because it is a multiple of 2, 4, 6, 8, and 12. On the other hand, it is guaranteed to work.

ENTER THE COLOR COMPUTER

Eratosthenes would probably have given up stuffing grape leaves for a Color Computer. Early mathematicians performed astonishing calculations by laborious manual methods; imagine what they could have done with a modern small computer that can perform hundreds of thousands of additions per second! Can we use Eratosthenes' method on the CC? If so, will it do the job in a reasonable time?

First, let's implement the sieve in BASIC. Figure 2 shows an Extended Color BASIC program that finds the prime numbers from 0 through a given number.

We've used a special way of compressing the number range in this routine. We could have used a one-dimensional array to hold the number range, but this would have used an enormous amount of space, as each array element is 5 bytes. A number range of 1000 would have required 5000 bytes.

What is the most efficient way of representing a number range? About the most efficient is letting one bit represent each number. The first bit would represent a number of 0, the next a number of 1, and so forth. This scheme

```

100 * PRIME GENERATOR
110 S=1536
120 INPUT "N OF BYTES":E
130 E=E*8-1
140 PRINT "NUMBERS FROM 0 THROUGH " E;" WILL BE TESTED FOR PRIMESS"
150 PMODE 4,1:SCREEN 1,0
160 PCLS
170 FOR A=S TO S+E/8
180 POKE A,170
190 NEXT A
200 POKE S,10
210 FOR I=3 TO (E+1)/2 STEP 2
220 FOR J=2 TO (E+1)/2
230 IF I#J THEN GOTO 290
240 A=PEEK(S+INT((I*J)/8))
250 B=INT(2^(7-(I*J-INT((I*J)/8))/8))
260 A=(A OR B)
270 POKE S+INT((I*J)/8),A
280 NEXT J
290 NEXT I
300 IF INKEY="" THEN GOTO 300
310 FOR A=S TO S+E/8
320 B=PEEK(A)
330 FOR I=7 TO 0 STEP -1
340 C=INT(2^I)
350 IF (B AND C)=C THEN GOTO 370
360 PRINT (A-S)*8+(7-I);
370 NEXT I
380 NEXT A

```

Figure 2.

is shown in Figure 3. You can see that each byte represents eight numbers, and that 125 bytes would represent numbers from 0 through 999, a reduction of 40 to 1 over an array. The bytes representing the number range can be anywhere in RAM.

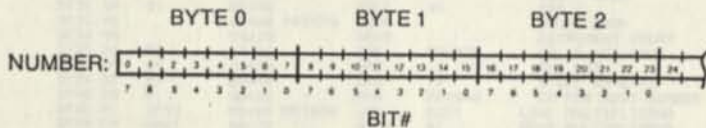


Figure 3. Mapping Numbers by Bits

We'll start off with all bits set to 0, representing numbers that are possible primes. We'll then find multiples and set the corresponding bits to 1, representing "non-primes." The numbers set to 0 after all processing is done will be primes. We should see something like Figure 4 when we're done.

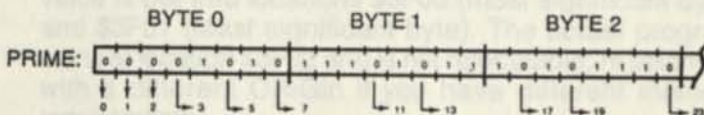


Figure 4. Bit Map After Primes Have Been Found

When I say, "we should see," I mean that literally. Why not put the bytes representing the numbers into a graphics page and display the entire number range? We should then be able to see the entire number range, the whole sieving process, and the resulting primes on the screen!

The program first sets variable S to 1536. This number is the start of the graphics screen area in RAM.

Next, variable E is INPUT. E represents the number of bytes used, which will represent a number range of 0 through 8*E, as there are eight bits per byte. The last number in the range will be E*8-1, and E is reset to this number in line 130.

Line 150 sets the highest resolution graphics mode in Extended Color BASIC, and switches to a graphics screen. The screen is then cleared by a PCLS, setting all bits to 0. This clear may not be obvious, but in a two-color mode, each graphics element is represented by one bit, and the PCLS sets all elements to the background color or 0.

The loop at lines 170 through 190 performs a somewhat dishonest maneuver. It sets all even numbers to 1, indicating "non-prime." We know that all even numbers will be non-prime, and this reduces the overall processing time.

At the end of the loop, we store a 00001010 into the numbers 0 through 7, setting 0, 1, 2, 3, 5, and 7 to primes, another dishonest tactic. From here on, however, we'll behave ourselves.

The heart of this routine is in lines 210 through 290. This is the code that finds all multiples of 3, 5, 7, 9, 11, and so forth, and sets the bit to a "non-prime." Variable I is the "multiplicand" in the loop, and varies by 2. Variable J is the multiple, and varies by 1. The product of I*J is the number to be "struck out." The first time through the loop we'd have the non-primes 6, 9, 12, 15, etc., the second time the non-primes 10, 15, 20, etc., and so on. Another time-saving trick we're using here is to step only through one-half the range ((E+1)/2), to avoid unnecessary processing of large numbers.

The only complication here is the "bit-finagling" we have to perform to find the proper bit and byte in RAM. The location in RAM (in the graphics page) is given by:

$$(S + \text{INT}((I*J)/8))$$

The term I*J is the number. The integer value of this product divided by 8 is the byte location of the byte containing the number. Variable A in line 240 holds the eight bits of the byte from the PEEK of the location in the graphics page.

We now have the byte containing the bit, but what about the bit itself? The term:

$$(I*J - \text{INT}((I*J)/8)*8)$$

finds the "bit location" within the byte. This number will be 0 through 7 representing the leftmost to rightmost bit. If we subtract this value from 7, we now have 7 through 0, representing the power of two of the bit location. This power of two is then used in 2^n to yield 128, 64, 32, 16, 8, 4, 2, or 1, which is the actual bit value involved. This bit value (variable B) is "ORed" with the PEEK value to set the bit to indicate a non-prime condition. A POKE at line 270 restores the byte in RAM.

At the end of the loop in line 300, we've gone through all multiplicands and multiples, and have "struck out" all non-primes.

Don't forget that while we've been doing this, we've seen the whole sieving process on the graphics screen, thanks to the magic of the Color Computer and dolmadakia (stuffed grape leaves). The resulting display represents all remaining prime numbers.

Line 300 looks for a keypress before printing out the primes. The loop at lines 310 through 380 goes through all bytes a bit at a time and checks for a bit value of 0. If the bit value is a 0, the number is indeed prime and is PRINTed out by:

$$360 \text{ PRINT } (A-S)*8 + (7-I);$$

MAKING ERATOSTHENES EVEN HAPPIER

The sieving process works fine, but has one drawback—it takes hours for large number ranges. Although the old Greek would have been ecstatic over this display, I suspect he would have given up even his baklava for the

assembly-language version of his sieve. Would you believe 32 seconds to find all primes from 0 through 49,151? Read on . . .

The assembly-language version of this program is shown in Figure 5. It is called from BASIC by a `USR` call and assumes that the RAM area used for storage is graphics page 0 starting at 1536. It also assumes that all even numbers have been cleared to 1.

```

3F00      00100      ORG      $3F00
00110      *****
00120      * PRIME NUMBERS BY "SIEVE" METHOD *
00130      * INPUT: LAST CONTAINS LAST NUMBER TESTED *
00140      * OUTPUT: PRIMES IN 1536 AREA *
00150      *****
3F00      0000      00160      LAST      FCB      0          LAST NUMBER TESTED
3F02      CC      0003      00170      PRIME      LDD      M3          START NUMBER
3F05      FD      3F53      00180      STD      CURI          INITIALIZE
3F08      FC      3F53      00190      PRI030     LDD      CURI          GET CURRENT MULTIPLICAND
3F0B      FD      3F55      00200      STD      CURN          INITIALIZE CURRENT NUMBER
3F0E      FC      3F55      00210      PRI040     LDD      CURN          GET CURRENT NUMBER
3F11      F3      3F53      00220      ADDD    CURI          FIND NEXT NUMBER
3F14      25      27        00230      BCS     PRI090     GO IF OVER 64K
3F16      FD      3F55      00240      STD      CURN          STORE
3F19      10B3     3F00     00250      CMPO   LAST       TEST FOR LAST
3F1D      22      1E        00260      BHI     PRI090     GO IF CURRENT GT LAST
3F1F      44          00270      LSR    LSR        LSR
3F20      56          00280      ROR    ROR        ROR
3F21      44          00290      LSR    LSR        LSR
3F22      56          00300      POR    POR        POR
3F23      44          00310      LSR    LSR        LSR
3F24      56          00320      ROR    ROR        ROR
3F25      C3      0600     00330      ADDD   #1536      POINT TO ADDRESS
3F28      1F      01        00340      TFR    D,X        HOW IN X
3F2A      FC      3F55     00350      LDD    CURN       CURRENT NUMBER
3F2D      C4      07        00360      ANCB  #7          GET BIT # 0-7
3F2F      5C          00370      INCB  CLEAR MASK
3F30      1A      01        00380      CLRA  #1          NOW 1 THROUGH 8
3F31      1F      01        00390      ORCA  #1          SET C
3F33      46          00400      PRI070  RORA          SHIFT MASK
3F34      5A          00410      DECB  DECREMENT COUNT
3F35      26      FC        00420      BNE   PRI070     GO IF NOT DONE
3F37      AA      94        00430      ORA   ,X          SET BIT
3F39      A7      84        00440      STA   ,X          STORE IN BYTE
3F3B      20      D1        00450      BPA   PRI040     GO FOR NEXT NUMBER
3F3C      FC      3F53     00460      LDD    CURI       LOAD MULTIPLICAND
3F40      C3      0001     00470      ADDD  #1          BUMP MULTIPLICAND
3F43      C3      0001     00480      ADDD  #1          THICE FOR ODD #
3F46      FD      3F53     00490      STD    CURI       STORE
3F49      F3      3F53     00500      CUI   #          THICE # FOR LIMIT CHK
3F4C      10B3     3F00     00510      CMPO  LAST       TEST FOR DONE
3F50      23      86        00520      BLS   PRI030     GO IF CURRENT LE LAST
3F52      39          00530      RTS   RETURN
3F53          0000     00540      CURI  FCB      0          MULTIPLICAND
3F55          0000     00550      CURN  FCB      0          CURRENT NUMBER
0000          0000     00560      END
0000      TOTAL ERRORS

```

Figure 5.

One parameter is passed to the assembly-language program, the value of the last number to be tested. This value is put into locations `$3F00` (most significant byte) and `$3F01` (least significant byte). The actual program starts at location `$3F02` and is not relocatable; reassemble with a different `ORiGin` if you have different memory requirements.

The algorithm used for this program is very similar to the algorithm discussed above. Multiplicands from 3—3, 5, 7, and so forth, are used in the outer loop from edit line 190 through 520 (notice the indentations in the listing comments). The inner loop from lines 210 through 450, multiplies the multiplicand by 2, 3, 4, etc., to find the number to be "struck out."

The byte address is found as in the BASIC program — by dividing the number by 8. A bit mask is established as in the BASIC counterpart and used to set the appropriate bit to a 1 value.

DETAILED DESCRIPTION OF THE PROGRAM

An immediate value of 3 (`#3`) is loaded into the D register (A and B taken together to form one 16-bit register) and stored into location `CURI`. Location `CURI` holds the current multiplicand.

The `LDD` at instruction `PRI030` loads the current multiplicand and stores it in location `CURN`. `CURN` holds the current number for the "strike out." The inner loop at `PRI040` loads `CURN`, adds `CURI`, and stores it in `CURN` for

the next strike out number. This effectively multiplies the multiplicand by 2, 3, 4, 5, etc., each pass through the loop. The `BCS` instruction branches to the next multiplicand at `PRI080` if a carry occurred. A carry will only occur if `CURN` is incremented above 65,535, and this ensures that the loop terminates properly for large multiplicands.

The loop also terminates for a current multiple greater than `LAST`, the last number to be tested.

The three sets of shifts (`LSRA/RORB`) finds the byte displacement by shifting the D register 3 bits to the right and dividing by 8. A value of 1536 added to the displacement points to the actual byte address containing the bit for the number. This pointer value is transferred to the X register in preparation for changing the bit value in the byte.

`CURN` is now `ANDed` with 7 to find the number of the bit that represents the number from 0 (leftmost) to 7 (rightmost). This number is incremented by one (1 through 8); the A register is cleared, and the Carry condition code is set by `ORCC #1`. All of this is in preparation for finding the "mask" for an OR operation that will be similar to the BASIC `OR`.

The loop at `PRI070` shifts the mask (in the Carry flag and the A register) right until it is aligned at the proper bit position. The count is the B register, and is decremented until it is 0.

After the shift, the byte containing the bit is `ORed` with the A register and restored into the byte address to strike out the non-prime number.

The outer loop at `PRI080` loads `CURI` and adds 2 to the value. `CURI` is initially 3, and will increment to the next odd number each time through the outer loop. The outer loop is terminated when the multiplicand is greater than half of the `LAST` value; all numbers have been struck out in this case.

USING THE ASSEMBLY-LANGUAGE PROGRAM

Figure 6 shows the assembly-language program in the form of `DATA` statements embedded in the BASIC driver. The assembly-language code is moved to the `$3F00` area before execution. Before loading the program, be certain to perform a "`CLEAR 200,&H3EFF`" to protect the `$3F00` area.

The program first asks for the # of bytes; enter 1/8 of the number range to be tested. For example, if you want to test 49,152 bytes, enter "6144." This, by the way, is the maximum number range that you'll be able to see on the screen.

The program will then strike out all even numbers; this will be a minute or so of slow BASIC processing on the screen. After the initial pass, however, things move like Greek folk dance. You'll see the multiples of 3 zap diagonally across the screen, followed by multiples of 5, followed by multiples of 7, and so forth. As numbers are struck out, you'll see the remaining primes come into focus. If only Eratosthenes were present . . .

After about 32 seconds, only primes remain on the screen. Press any key to get a listing of all primes; you may change line 360 to an "`PRINT #-2`" if desired, for a hard copy.

This is only one minor topic of many fascinating mathematical recreations that can be investigated on the Col

```

30 * PRIME GENERATOR
40 DATA 0,0,204,0,2,252,63,63,252,63
41 DATA 83,252,63,63,252,63,63,243,63,63
42 DATA 37,39,252,63,63,16,179,63,0,34
43 DATA 38,69,66,68,66,63,66,195,6,0
44 DATA 31,1,252,63,63,196,7,92,75,26
45 DATA 1,70,90,39,252,170,132,167,132,32
46 DATA 209,252,63,63,195,0,1,195,0,1
47 DATA 253,63,63,243,63,63,16,179,63,0
48 DATA 35,182,57,0,0,0,0
50 FOR A=LH3F00 TO LH3F56
60 READ B
70 POKE A,B
80 NEXT A
110 S=1536
120 INPUT "N OF BYTES":E
130 E=EB-1
140 PRINT "NUMBERS FROM 0 THROUGH " ; E;" WILL BE TESTED FOR PRIMENESS"
150 POKE 4,1:SCREEN 1,0
160 PCLS
170 FOR A=S TO 3+E*8
180 POKE A,170
190 NEXT A
200 POKE 9,10
210 DEFUSR0=LH3F02
220 POKE LH3F00,INT(E/256)
230 POKE LH3F01,E-INT(E/256)*256
240 PUSHR0:G
300 IF INKEY="" THEN GOTO 300
310 FOR A=S TO 3+E*8
320 D=PEEK(A)
330 FOR I=7 TO 0 STEP -1
340 C=INT(2/I)
350 IF (B AND C)≠C THEN GOTO 370
360 PRINT (A-9)/8+(7-I);
370 NEXT I
380 NEXT A

```

Figure 6.

Computer and for which assembly-language is desirable. A good place to get a background on topics in recreational mathematics is in Martin Gardner's column in "Scientific American"; Mr. Gardner is semi-retired now, and his "Mathematical Games" column has now become Hofstadter's "Metamagical Themas" (Hofstadter is the author of the best-selling "Gödel, Escher, Bach"), but past and present columns contain much valuable information on mathematical topics that can be implemented on small computers.

(Not to put myself in the same esteemed company, but I might mention another text on microcomputer math topics — "Microcomputer Math" from Howard W. Sams, and Co. by yours truly.)

Try the above program. You may be pleasantly surprised by the results and it may spur your interest in this assembly-language area. If all else fails, find a Greek mathematician and take him out to lunch. . . .



Convert 35 mm to 16 mm

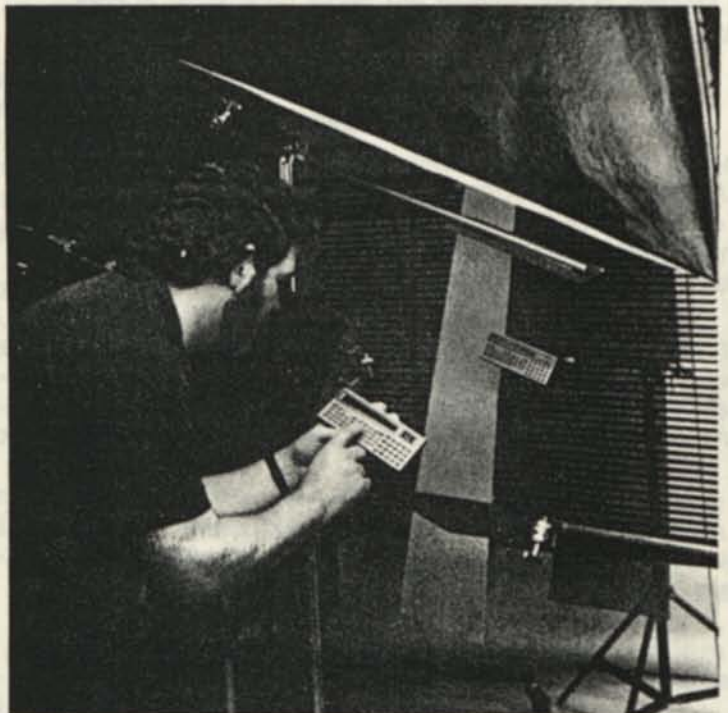
Robert Edwards
 Steed Rd. Box 581-A
 Chenango Forks, NY 13746

A Program for Pocket Computer 1

```

10: "A"
   : PAUSE "CONVERT 35 mm TO 16 mm"
   : BEEP 1
   : CLEAR
20: INPUT "FOOTAGE AND FRAMES "; A
30: B=INT(A)
   : C=(A-B)*10
40: D=B*16+C
   : E=D/40
50: F=INT(E)
   : G=E-F
60: H=C*40
70: PRINT F; "FEET "; H; "FRAMES "
80: GOTO 20
90: "B"
   : PAUSE "CONVERT 16 MM TO 35 MM"
   : BEEP 1
   : CLEAR
100: INPUT "FOOTAGE AND FRAMES "; A
110: B=INT(A)
   : C=(A-B)*100
120: D=B*40+C
   : E=D/16
130: F=INT(E)
   : G=E-F
140: H=C*16
150: PRINT F; " FEET ";H;" FRAMES "
160: GOTO 100
170: END

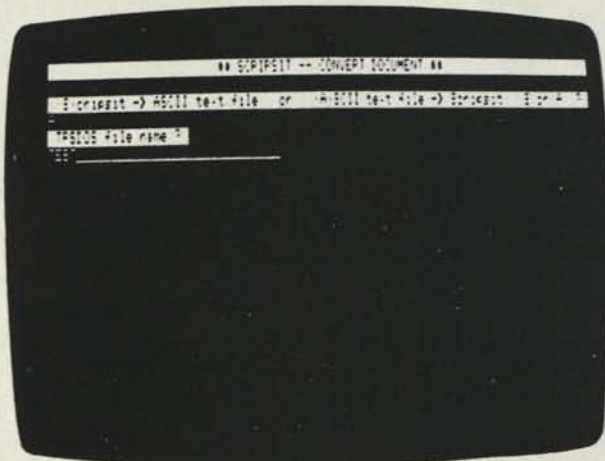
```



More about SCRIPSIT 2.0

This article is for all of you SCRIPSIT 2.0 users who know you have a wonderful word-processing program and want to learn more about its finer points. In this article we will try to explain the use of some of the features that might be unfamiliar to you. We will assume that you have some skill with SCRIPSIT 2.0, but if you haven't used SCRIPSIT 2.0 very much, or are unsure of your abilities, you will want to practice what you learn here before trying these techniques on actual documents. It will also be helpful if you read the sections on the CONVERT, ASSEMBLE, and COPY utilities in the SCRIPSIT 2.0 reference manual. We are going to cover these utilities using SCRIPSIT 2.0 with VisiCalc and Profile.

First a little instruction on a utility that is very important to this article, the Convert utility. The Convert utility allows you to convert an ASCII TRSDOS file to a SCRIPSIT document or a Scripsit document to an ASCII TRSDOS file. This feature allows us to use data or text created by other software packages such as VisiCalc, Profile, or even your own BASIC programs. To convert an ASCII disk file into a SCRIPSIT 2.0 document, or a SCRIPSIT 2.0 document into a ASCII disk file do the following. From an open document press **(ESC) (U) (ESC)** which displays the Convert prompt or press **(ESC) (U) (C)** to go directly to the utility. The Convert utility screen will be displayed.



To convert an ASCII disk file into a Scripsit document, press **(A)**, or to convert a SCRIPSIT 2.0 document into an ASCII file, press **(S)**. The second prompt asks for a "TRSDOS file name?" This is either the name of a file already on a disk or the disk file name under which the current document will be saved. If you are unsure of the requirements for a TRSDOS file name, be sure to read the TRSDOS section about FILE NAMES in your Model II owner's manual. To use the convert utility you must either have extra drives with a TRSDOS formatted disk in one of

them, or a "Minimum Scripsit Disk" (refer to the Scripsit reference manual to create this minimum disk) on which to put files and from which to take files.

VISICALC® PRF FILES

One of the print options of VisiCalc is the ability to "print" a sheet to a disk file. This file is called a PRF file and is an ASCII text file, which can be easily converted to a Scripsit document. To create a PRF file refer to the print commands section of your VisiCalc manual (/PF). Be sure that the width of the margins of the VisiCalc sheet are no wider than the margins of your original document. If they are, don't worry, you can fix them later with Scripsit. After you have created and printed your sheet to a disk file, you can use the information in a report, contract, list of specifications, or any other document without having to retype anything. To use the PRF file with a document you have created, open the document and place the cursor at the very end of the document. From the open document use the convert utility as described earlier (use **A** and your file name). After you enter the file name and press **(ENTER)** you will see the information appear on your screen as if it is being added to your document. When the PRF file has been loaded into your Scripsit document and the cursor reappears you can use SCRIPSIT 2.0 to edit this information so it will appear exactly as you want. You can change the margins, delete information or insert print control codes, etc.

Another method (although more complicated) of using the PRF files with Scripsit is to create a new document. Convert the PRF file into the empty document, and use the Assemble utility of SCRIPSIT 2.0 to combine the two documents. Pressing **(ESC) (U) (A)** will display the Assemble Document menu. Enter the document name (file name) of the one to be added to the document that is currently open. Press **(ENTER)** and it will be included in this document at the end of the document. You can use this process to put as many different sheets of information together as needed. By editing the information, you can create a custom document with a minimum of effort.

PROFILE MERGE FILES

We will refer you to the Profile article elsewhere in the issue of Microcomputer News on how to create a regular merge file with PROFILE to use with SCRIPSIT 2.0. Here we will go over some of the less obvious uses of the Merge utility. Sometimes you may find that a file you have already created is just what you want for a new document except for a couple of items. A file might contain, for example, the first and last names for a mailing list but you don't want titles such as Dr., Mrs., Mr., etc. Using the Convert and Assemble utilities and the advanced editing features of SCRIPSIT 2.0, you can fix that problem. Also, you can combine two files that have the same variable names and the

same number of variables in each grouping. This would be a convenient way to combine two mailing lists. To do this, first open a new document and use the Convert utility discussed before to get the file into SCRIPSIT. Answer (A) to the first prompt. The TRSDOS file name you will use when you convert your PROFILE created file is the same one you would use if you were going to do a merge instead. (Refer to the section in the PROFILE manual on "Merging with SCRIPSIT 2.0"). Now look at your new document. It will look much the same as a merge file for SCRIPSIT 2.0. If you want to append a file so that you don't have to do a merge twice, open a new document, follow the same procedure, and then use the Assemble utility to add the other document to the first one. In looking at the assembled document you will see the variable names from the second merge document. Delete the second set of merge variable names using your normal editing functions to modify the information. Make sure there is an (ENTER) after each variable and a second (ENTER) after the last variable in each group of variables. If you don't have the correct (ENTER)'s then the document will not print. You are now ready to print your merge document.

LABELS, ENVELOPES, AND COMMON FORMS

Have you ever used SCRIPSIT 2.0 to write a business letter then manually (or with a typewriter) addressed the envelope? Didn't that seem like a waste of time? A better way is to set up a document for both a large envelope and a small envelope with the return address and X in the address block for the recipient's name, street address, city, state, and zip code. Then after you type your document, and before you are ready to print, you can assemble your envelope document with this one. Remember, that when you assemble a document, the second document is on a separate page and will print as a separate page. Fill in the name and address in place of the x's (you can define a user key to do this automatically.), and after the letter is printed you can slip an envelope into the printer, print the next page, and the envelope will be addressed and ready to send.

This also holds true for forms. You can set up a document (again with Xs) and then fill in the correct information on each form, print the form and you're ready to fill in the next form. Or if you have the desired information on Profile you can make the form your base document and use the MERGE utility to print your forms. This may sound complicated, but if you have ever had to fill out the same form over and over, with only the information that fills in the blanks changing each time, you will appreciate how much easier this method is.

Labels are also easy to print using a common form with Xs where the information goes. You could even set up a mailing list with a base document having the merge variables positioned correctly and another document with the names. You can't sort or select any specific information, but if you only need to print the labels then this procedure will work well. If you need to add information to the merge document, add the information, convert the merge file and then print the labels using the label base document and the Scripsit merge utility.

CONCLUSION

We hope these ideas on different ways to use SCRIPSIT 2.0 will be helpful in your every day business world. Some of the secretaries here at Radio Shack are just finally getting used to the incredible features of Scripsit. Some of us have fairly poor handwriting and occasionally our writing efforts may require several revisions. With Scripsit our handwriting becomes legible (because we now use a printer instead of scribbling) and as many times as we've revised this article, we sure are glad we have Scripsit so we don't have to retype the entire article each time we make a change. If you employers think Scripsit 2.0 is just for the secretarial pool, try using it yourself. Type a couple of memos instead of writing them down and having your secretary type them. Instead of having to do everything your secretary will only have to proof, do the final print, and distribute the memo allowing her (him) to make more productive use of the few hours in a work day. You might consider doing the first draft of a letter yourself, also. With this program the sometimes boring task of writing becomes fun.



**Computer Customer Services
Address and Phone Number**

8AM to 5PM Central Time

Computer Customer Services
400 Atrium, One Tandy Center
Fort Worth, Texas 76102

- Model I/III Business Group 817-870-2041
- Model II/16 Business Group 817-870-2042
- Languages and Compilers 817-870-2044
- Color and Pocket Computer Group 817-870-2150
- Hardware and Communications Group ... 817-870-2571
- Educational Software 817-390-3302
- Games, Books, and New Products 817-870-2271

New Menu System

Menu Structure Takes on New Look, Promotes Ease of Use

Editor's Note: The CompuServe Information Service is one of the largest information and entertainment services available to owners of personal computers and computer terminals. With each issue of TRS-80 Microcomputer News, various features of CompuServe will be discussed.

The CompuServe Information Service is sold at Radio Shack stores nationwide and in Canada.

As part of CompuServe's continuing effort to make its information service responsive to customers needs, a new menu structure has been put into effect.

All of the information contained on CompuServe has been organized under 6 primary interest sections:

1. Newspapers
2. Home Services
3. Business and Financial Services
4. Personal Computing Services
5. User Information
6. Index.

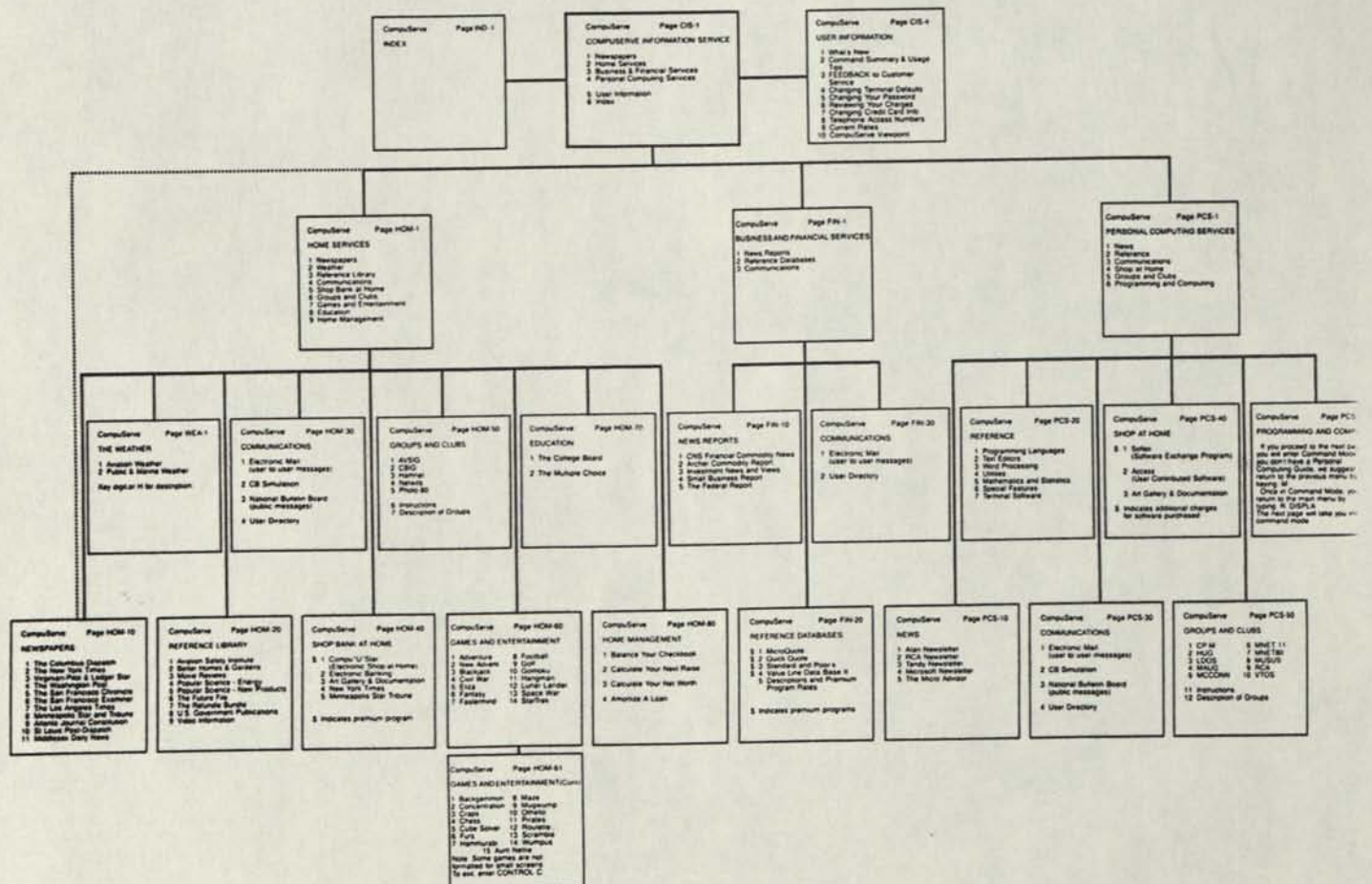
Newspapers.

Our electronic newspapers are now directly accessible from the new main menu as selection 1. All eleven newspapers can also be reached from the Home Services menu.

Home Services.

The Home Services sections contains general and family-related information and services. Currently, these include:

1. Newspapers
2. Weather: This includes both aviation and public marine weather.
3. Reference Library: Under this menu you can find general interest newsletters from such information providers as the Aviation Safety Institute, magazines such as Better Homes & Gardens, and the Refund Bundle.



4. Communications: A communications area has been placed in each of the three main service sections to enable you to have access to these often-used offerings as easily as possible. This communications area includes Electronic Mail, CB, the National Bulletin Board and the on-line User Directory.
5. Shop/Bank at Home: Electronic banking and shopping offerings can be found under this choice including, among others, Comp-U-Star and Bank-at-Home.
6. Groups and Clubs: The groups and clubs available under Home Services are general interest groups that share an interest in non-technical subjects.
7. Games and Entertainment: The games menu contains your old favorites, like Adventure and Football, in addition to games that were previously available on other areas of the service. Interactive games, such as DecWars and Space War, can be reached through this menu.
8. Education: The education section contains the College Board that is a must for any prospective college student and the Multiple Choice where you can test your trivia skills and be challenged by many types of self development tests.
9. Home Management: In this section are grouped several very handy home-management programs for doing such otherwise tedious jobs as balancing your checkbook or figuring out a loan schedule.

Business and Financial Services.

The Business and Financial Services section includes all business and financial news reports, updated stock quotes and business reference databases including, among others, MicroQuote and Standard and Poor's.

Personal Computing Services.

The Personal Computing Services section includes news, programming languages, a word processor, a Software Exchange program for purchasing software, computer related groups and clubs and an area for programming activities.

User Information.

The User Information section contains material pertinent to your general use of the information service including Feedback, documentation and product ordering, What's New, a general command summary as well as instructions for verifying and/or altering your terminal defaults, password and credit card information. Detailed billing information, current access rates and telephone numbers are located here.

Index.

The Index section contains the most up-to-date version of our Subject Index.

* * * * *

Questions and comments about the CompuServe Information Service can be sent to Richard A. Baker, editorial director, CompuServe Information Service, P.O. Box 20212, Columbus, Ohio 43220 or through Feedback on the User Information menu.

Racing Information

The method of accessing Racing Information has changed: RIS may be accessed by going to page HOM-110 on CompuServe. When "Command" appears, type "SS4." Then type "RF." Then type the lowest message number that is on the system.

Michael F. Hollander
 Director of Operations
 Racing Information Systems
 User ID: 70001,557

Microcomputers for Neophytes

The University of Oklahoma, known for excellence in computing sciences and computer related mathematics, and its Saturday workshops in computer usage for secondary and elementary teachers, has introduced a new course in *microcomputer programming for the complete neophyte*. The course is a hands-on course using 30 TRS-80 computers for 40 students. Class meets for two hour-long lecture-demonstrations each week, with students expected to put in at least another four hours a week of lab plus homework. (Lab is open 9:00 a.m.-9:00 p.m. M Tu W Th F with breaks for class, which are held in the lab room.) The text used is *EXPLORE COMPUTING with TRS-80 and COMMON SENSE* (Prentice Hall). Grades are on an S/U basis, with either one or two hours of strictly undergraduate credit available, depending upon amount and difficulty of work undertaken.

Dr. Richard V. Andree said the course was created in 1981 primarily to help future teachers, but the course attracted twice as many students as could be accommodated, including three Ph.D.s and numerous graduate students who could not receive credit, but worked hard anyway. Only four of the forty students who enrolled were Education majors, the others came from all over the University.

Highly successful, the course was enlarged in Spring 1982 and will be further enlarged in Fall 1982 if additional microcomputers can be obtained.

New Computer Clubs

The BASICS
 Frederick Town High School
 % Gary Mayton
 117 Columbus Rd.
 Fredericktown, OH 43019

Johnson Amateur Computer Society
 34 South Fir Street
 Medford, OR 97501
 Computer Bulletin Board System
 phone number 503-535-6883

● Movie Reviews

Into horror movies? According to Jay Brown, a popular movie review columnist who can now be read on the Dow Jones News/Retrieval service, those of you who are could belong to a beastly class of human beings who delight in potent doses of blood and gore.

In one of his latest movie reviews, the veteran columnist says: "This blood drenched horror number faithfully delivers a full ration of gore that should delight fans of this genre."

Brown recently wrote this in his review of the "The Beast Within" — a movie about a teenager who turns into a grotesque monster.

And all you thought Brown wrote about had to do with dramatic effects and the intricacies of theatrical style.

His writing is of a feisty sort and has most recently been added to the Dow Jones News/Retrieval service.

Another sample of his writing on this latest consumer oriented data base is his review of the movie, "Amin — The Rise and Fall" — which he rates as (FAIR).

He wrote: "... Joseph Olita, in the title role, is amazingly similar in appearance to the brutal Amin who is shown eating human flesh, cutting up his wife's corpse and preserving the heads of his enemies in his refrigerator."

But Brown does not always focus on the gore. In fact, there are times when he goes to as many as eight different movies in one week, when there is a glut of new releases.

For example, his review of the movie "Reds," begins with: "A brilliant and passionate thick slice of history with Warren Beatty splendidly performing as radical journalist John Reed who chronicled the Russian revolution."

And there's Brown's review of the movie, "On Golden Pond," in which he wrote: "The main reason to see this sentimental drama is the magnificent casting of Henry Fonda and Katherine Hepburn as an elderly couple trying to enjoy perhaps their last summer together while embroiled as usual in family conflicts."

Not to be forgotten is his review of the movie, "Absence of Malice," in which he wrote: "A brisk, intelligent drama that takes a penetrating look at the ethics of journalism in view of the quest for truth and justice."

After spending several years as a newspaper reporter and editor for several publications, Brown decided to branch off on his own and formed the Cineman Syndicates in 1975.

His reviews are being published by more than 70 newspapers and news services, including publications in Saudi Arabia, Caracas, and the Phillipines.

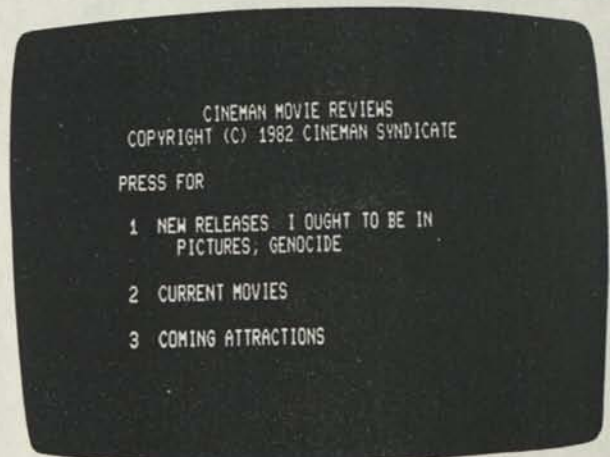
"This started as a newspaper syndicate publication," Brown said recently. "I had noticed that nobody had a cap-sulized movie review . . . It was something that I sort of needed myself and I found it lacking," he said.

What was lacking was easy, quick and up-to-date access to movie reviews that are interesting, informative, and concise.

Brown said he sees approximately 100 films a year. On the Dow Jones News/Retrieval service, Cinema reviews can be accessed by typing:

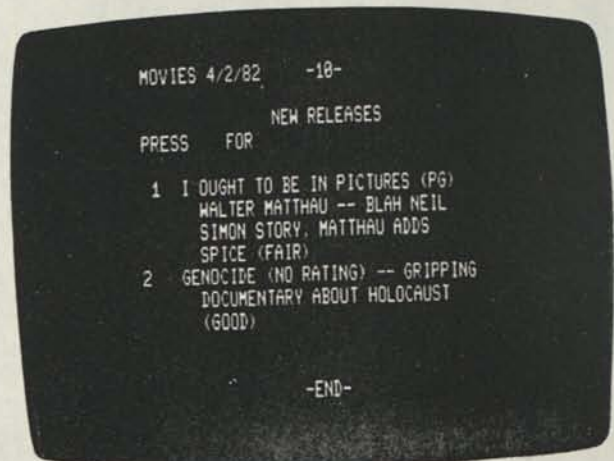
```
//MOVIES <ENTER>
```

and you will see the menu listed on screen 1, below.



Screen 1.

If you press Number 1, for New Releases, another menu similar to the one on screen 2, will be displayed.



Screen 2.

At the bottom of the screen, notice the word: -E- That tells you that there are no more movies in that category.

To call up one of the reviews, press a number (<ENTER>). In this case, if you press number 1, the review shown in screen 3 is displayed.

MOVIES 4/2/82 P100 ENDS AT 101

I OUGHT TO BE IN PICTURES
(FAIR)

WALTER MATTHAU'S DELIVERY OF CANTAN-
KEROUS WISECRACKS ADDS SOME SPICE TO
THIS OTHERWISE BLAH NEIL SIMON FILM
BASED ON HIS BROADWAY PLAY.

MATTHAU PLAYS A DOWN-AND-OUT HOLLY-
WOOD SCREENWRITER WHO IS REUNITED WITH
HIS TEEN-AGE DAUGHTER (DINAH MANOFF)
AFTER A 16-YEAR SEPARATION.

A LOT OF SARCASTIC SPARRING ENSUES IN
TYPICAL SIMON MANNER. BUT THE RELENT-
?

Screen 3.

If you want the remainder of the review, press (ENTER).

The Current Releases category on the initial menu works the same way. And the Coming Attractions category is even more simple to access. If you press number 3 on the main menu screen (see Screen 1), you will immediately see a screen showing coming attractions, similar to screen 4 below:

MOVIES 4/2/82 P196 ENDS AT 197

COMING ATTRACTIONS

SYLVESTER STALLONE WILL WRITE AND
STAR IN "PALS", A CONTEMPORARY COMEDY
TO BE DISTRIBUTED BY MGM.

|||
BARNARD HUGHES, WHO PLAYS THE TITLE
ROLE ON THE "MR. MERLIN" TV SERIES,
WILL JOIN BURT REYNOLDS AND GOLDIE HAWN
IN THE CAST OF WARNER BROTHERS' "BEST
FRIENDS," A ROMANTIC COMEDY.
|||

Screen 4.

If you are not a Dow Jones News/Retrieval service subscriber but are interested in this or other data bases, call the Dow Jones Customer Service hotline number: 257-5114. It's toll free.

DOW JONES
NEWS/RETRIEVAL®

micro-RESEARCH Problems for Computer Assisted Solution

Richard V. Andree
micro-RESEARCH Problems
P.O. 2910
Fort Worth, TX 76113-2910

This is our first micro-RESEARCH column in what we hope will be a long series.

A micro-RESEARCH problem is an open-ended problem which does not require advanced mathematics to *understand*—most are understandable by 7th grade students, all by students taking 9th grade algebra. Yet the problems are *not easy* to *solve* without the assistance of a microcomputer. Most first attempt computer solutions can be speeded up considerably by the judicious use of *common sense* (frequently called "computer sophistication"). The micro-RESEARCH problems are present to challenge *your* programming ability. I hope you enjoy them.

I will provide you with a sample solution to each month's problem, but you will have to wait a month or two to get it (after all, I want you to work out your own solution, not copy mine!) If I get some really unique written responses, I may publish one or two of them also. There is one important thought here—in most, if not all cases, there is no one correct answer or method to solve a particular problem. Be creative, and generate your solution. The problems are open-ended. Once the stated problem is solved, numerous possible extensions will occur to the solver. The extensions justify the use of the adjective "RESEARCH" in the title, since the full answers are not available in standard publications, and frequently are actually unknown. Yet the exploration of these problems is well within the ability of secondary students who can program in BASIC.

micro-RESEARCH Problem #1.

Find all integer squares $S = N * N$ which contain each of the ten digits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 *exactly once*.

You should find this problem easy to understand, and it can be attempted using a pocket calculator or a large table of squares. It takes little sophistication to recognize that only those squares containing exactly 10-digits need be considered. Furthermore, as soon as a repeated digit is discovered in a 10-digit square, it can be discarded without further checking. It takes some understanding of computer programming to display only the solutions. Can you do it?

Some obvious *extensions* (which are not as easy) include:

For "squares" read "cubes" or "fourth powers" or "factorials."

For "exactly once" read "exactly twice" or "exactly 3 times."

The problem may also be altered by using only the nine non-zero digits, 1, 2, . . . , 9.

What results can *you* come up with?

Using the Digitizer to Calculate Area Measurement and Curve Integration

The basic approach used here is to sum up strips of area equal to $\Delta X \cdot Y$. To measure the area inside a figure such as Figure 1, the area under the upper portion of the figure is considered positive and the area under the lower portion negative.

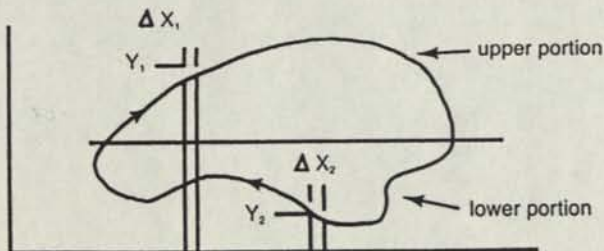


Figure 1

The Model II program below has the advantage of automatically closing the figure. This technique involves inserting a test for a specific set of coordinates. Line 398 checks to see if the value of the coordinates X and Y both equal zero. This should only be true when the first two coordinates are read in and when the program is terminated by pressing RESET on the digitizer.

Before running the program, press RESET on the digitizer so the first values for X and Y read from the Digitizer will be 0's. When all the coordinates have been input, press RESET again and the area of the figure will be printed.

```

100 CLS
   : CLEAR
150 DEFINI I-N
160 AR=0
240 DIM A$(8)
250 FOR I=0 TO 8
260 READ A,B
270 D=D+A*B
280 C=A+256*B
290 IF C>32767 THEN C=C-65536
300 A$(I)=C
310 NEXT I
320 IF D<>2050 THEN PRINT
   : PRINT"DATA ERROR"
   : GOTO 4030
330 REM
340 REM DATA LIST
350 REM
360 DATA 229,62,96,207,22,32,249,120,230,127,119
370 DATA 35,54,0,62,2,201,0
375 A=0
380 GOSUB 8000

```

```

390 XO=X
   : XP=X
   : YO=Y
   : YP=Y
395 GOSUB 8000
398 IF X=0 AND Y=0 THEN GOTO 450
400 DX=X-XP
401 AY=(Y+YP)/2-YO
410 AR=AR+(DX*AY)
420 XP=X
   : YP=Y
425 PRINT @160,AR;"
440 GOTO 395
450 SA=AR
   : X=XO
   : Y=YO
460 AR=((X-XP)*(Y+YO)/2)
470 AR=AR+SA
475 CLS
480 PRINT"AREA = ";AR
490 END
5000 REM READ ONE CHARACTER FROM THE RS-232
5010 REM
5020 J=0
5030 DEFUSR0=VARPTR(A$(0))
5040 J=USR0(J)
5050 RETURN
5060 REM
6000 REM GET AN 8-DIGIT STRING OF COORDINATES
6010 REM
6020 GOSUB 5000
   : IF J<>88 THEN 6020
6025 PRINT@100,A
   : A=A+1 'COUNTER
6030 D$=""
   : FOR I=1 TO 8
   : GOSUB 5040
   : D$=D$+CHR$(J)
   : NEXT I
6040 RETURN
6050 REM
7000 REM CONVERT 4 HEX DIGITS TO DECIMAL
7010 REM
7020 DE=0
7030 FOR I=0 TO 3
7040 J=ASC(MID$(HXJ$,I+1,1))-48
7050 IF J>9 THEN J=J-7
7060 DE=DE+J*16^(3-I)
7070 NEXT I
7080 IF DE>32767 THEN DE=DE-65536
7090 DE=.01*DE
7100 RETURN
7110 REM
8000 REM GET ONE SET OF (X,Y) COORDINATES
8010 REM
8020 GOSUB 6000
8030 HX$=LEFT$(D$,4)
8040 GOSUB 7000
8050 X=DE
8060 HX$=RIGHT$(D$,4)
8065 IF X=0 AND Y=0 AND AR<>0 GOTO 450
8070 Y=DE
8080 Y=DE

```

(Continued on Page

Merging Profile Files With Scripsit

The Small Computer Company
Written for the users of PROFILE II,
PROFILE+ and PROFILE III+

PROFILE Editor
P.O. Box 2910
Fort Worth, TX 76113-2910

We're often asked how, exactly, one merges PROFILE files with SCRIPSIT to create letters to one's accounts or mailing list. The merge process is remarkably simple, since PROFILE II, II+ (also called PROFILE+) and III+ all have built-in merge creation operations. These operations each build up to 5 special merge files compatible with SCRIPSIT, indicated in the directory by the extensions SR1, SR2, SR3, SR4 and SR5. Your merge document, therefore, will be "FILENAME/SR1" (or SR2, et cetera).

In the process of creating a merge file, the computer will usually write the file onto the Drive 0 diskette. Check that you have enough room on Drive 0—the calculation is given below; if not, we'll explain how to set up the file on another diskette with more space.

Let's assume that you own a small solar-powered toy company, SUN CHILD, INC. You have 3 salespeople, a 20-item catalog and 100 customers. You would like to give your best customers a special discount on a new solar calculator and want to send out personalized letters based on your PROFILE customer list.

PROFILE II & II+: STEP BY STEP



Screen 1: Profile II Menu

1. At the menu, press 5, "Define Selections." For the SUN CHILD company, you'd type in the filename SUN-SALE. Then press 1 for the selector number, and Y[es] to indicate that this is a new format.

2. You will see a "Define Selection Formats" screen. You can type in up to 16 names and numbers of fields you wish to use in your merge document. These names do not have to be the same as the names of your data fields. However, since they will be used as codes in the SCRIPSIT letter, you should make a note of the codes for later reference (this operation doesn't include a hardcopy option). For SUN CHILD, you would type in SALES (field 1), CUSTM (field 2), STREET (5), CITY (6), STATE (7) and ZIP (3). Press (ESC) to record the screen, and answer N[o] for password protection.

3. You'll be returned to the menu, where you will press selection "A" (or "4" if you're using the Limited Menu). The key (segment 1) fields will appear:



Screen 2: Profile II Select Records

You are asked for a sort field number. Since you want to sort by customer, press 2 for Field 2, and 25 for the number of characters.

4. A 16-entry selection format appears at the bottom of the screen. You want to select only those customers with total balances greater than \$100, so type in field 4, YTD (Year To Date) SALES. Press (ESC).

5. When the "Select Record" screen appears, type in GT for "greater than" and 100.00:

YTD SALES GT 100.00

Press **(ESC)**. The computer will select and sort your records according to your criteria and return you to the menu. If you want to see that the new file is on Drive 0, you'd e[X]it from the menu, type "DIR" (Drive 0 is assumed), and press ENTER. The file would be SUNSALE0/SR1, and unlike any other non-merge file in the directory, it would have logical record lengths (LRL) of only 1 byte. It would, however, have many records. This is the only format SCRIPSIT can accept.

You can also see the field names and data in that file if you wish (provided that you have the program SELECT/BAS somewhere on your diskettes). At TRSDOS READY, type BASIC SELECT/BAS -F:1 and press **(ENTER)**. The screen will show:

```
ENTER FILE NAME? (type in SUNSALE)
and then
ENTER SELECTION NUMBER? Type in "1."
```

The program will display the field names and data to be used in the SCRIPSIT files.

SPECIAL INSTRUCTIONS: To calculate the amount of room you'll need for the merge file, take the lengths of all fields you want to pass to SCRIPSIT, multiply by the number of records you expect to pass, then divide by 1280 bytes per gran (granule). The result, in grans, is the maximum amount of room you'll need. (Numbers of grans available appear at the bottom of the DIRECTORY screen.) DISK SPACE FULL or Error 27 appears if you attempt to put the merge file on a diskette with too little room.

If you don't have enough room on Drive 0, you can create a file on a diskette that does have room; the computer will write the information to it rather than trying to set up a new file on Drive 0. At TRSDOS READY, type:

```
CREATE filename/SRn:drive# LRECL=1,NRECS=#
```

For instance:

```
CREATE SUNSALE0/SR1:1 LRECL=1,NRECS=#
```

This preallocates a merge file with no records on the drive you want. Just make sure you have the right PROFILE diskette in the system when you start the SCRIPSIT merge operation.

6. Move the Drive 0 PROFILE diskette to Drive 1. (According to the 2.0 SCRIPSIT manual, you're supposed to extract the file from the PROFILE diskette and store it on another diskette. This isn't necessary as long as the merge file is on the PROFILE Drive 0 diskette. If, however, you have only 1 drive, you can copy the PROFILE file onto the SCRIPSIT diskette:

At TRSDOS READY type:

```
COPY SUNSALE0/SR1:0 :0
```

You will have to switch the PROFILE and SCRIPSIT diskettes at the appropriate prompts. Put your SCRIPSIT diskette in Drive 0. Create a base letter, embedding the codes you set up in the PROFILE program in braces. For example:

```
{CUSTM}
{STREET}
{CITY}, {STATE} {ZIP}
```

Dear {CUSTM}:

We've just bought a truckload of the brand-new SPACE AGE.T solar-powered calculator, and we're excited about it that we wanted to offer it to you, one of our most valued customers, at a pre-catalog discount.

Our catalog price — if we have any left to sell! — will be \$21.95. Your price is \$17.50 — more than 20% off!

I'm looking forward to your order. Call me at (214) 444-5555.

Best regards,

```
{SALES}
SUN CHILD, INC.
```

7. Next, get the merge utility. If you're using SCRIPSIT 1.0, press CONTROL U, then M[erge]. For SCRIPSIT 2.0, do the same. ("Merge" is listed on the second utility screen, not the first, in the 2.0 version. Press any arrow to see it.)

8. The program will ask you: Convert this document into merge information or use Base document? B

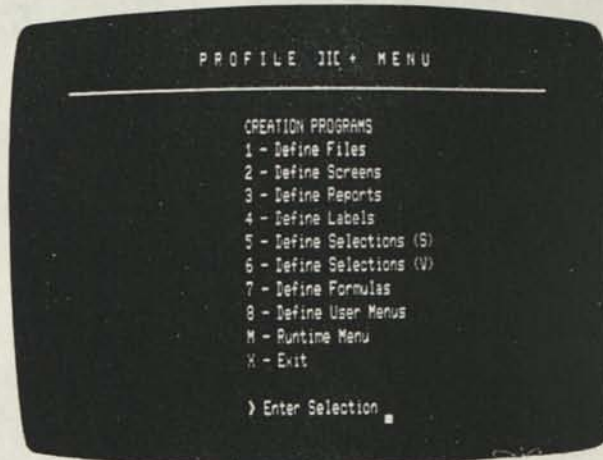
Leave the B — press **(ENTER)**.

Name of merge document (enter TRSDOS file name)?

For our purposes, SUNSALE0/SR1 and press **(ENTER)**.

9. You will be prompted for printing information. When you've filled in all the blanks, the computer will bring up the SCRIPSIT letter up to the screen, fill in a customer name and address, and print out the letter while you watch. It can do the same for as many customer names as appear in the merge file.

PROFILE III+ : STEP BY STEP



Screen 3: Profile III+ Creation Menu

1. The menu choice for "Define Selection" for SCRIPSIT is 5 on the CREATION diskette. Follow the prompts using the same information as in the PROFILE II+ instructions above. You can make a hardcopy of the codes in this program.

The instruction for creating a file on a diskette in a drive other than 0 is slightly different:

```
CREATE filename/SRn:d (LRL=1,REC=0)
```

where "n" is the file number and "d" is the drive number.



Screen 4: Profile III+ Runtime Menu

2. The "Select Records (S)" menu choice is 6 on the RUNTIME menu. Follow the prompts, using the same information as above. The merge files are FILENAME/SR1, SR2, SR3, SR4, SR5 (and they too have record lengths of 1).

3. Move the PROFILE III+ RUNTIME Drive 0 diskette in Drive 1 and put your SuperSCRIPSIT diskette in Drive 0. Open or type out the base letter, but in this case, embed the codes in @s; however, DO NOT use the @ key on the keyboard to make the character—press SHIFT/0 instead: This does work as unlikely as it sounds, and is necessary because the regular @ is used to designate "record number" in the merge format.

4. Press CONTROL F (the real @ key and the letter F) for the merge operation. When prompted for the name of the merge document, type in SUNSALE0/SR1. The computer will read the merge file off the PROFILE diskette, prompt you for print parameters, and type the letters.

LETTERS

As you are probably aware, the new PROFILE+ enhancement package will be available from Radio Shack by the time you read this column. This package includes the programs most often requested by the large base of PROFILE II users.

For example, Mr. Robert P. Fischer of Lancaster, Pennsylvania, writes: "One problem with printing labels from PROFILE II is that two or more fields cannot be printed together without the blank spaces being printed between them. For instance, a last name field length of 20 and a first name field length of 30 requires a label 50 characters wide to print the complete name, even when the names may be much shorter than the space allotted."

PROFILE+ now includes a push-left field indicator in the label format operation, which moves names and other data together when there are unused spaces in the fields.

Another example: Mr. Sammy Robinett of Gorman, Texas, writes, "I have encountered a small problem. When using more than one Total Field in a report, I get a line feed

after each column is totaled so that the totals are skewed down the page instead of being on one line. Can you recommend a patch or anything else that I can do to correct this, or is this something that I will have to learn to live with?"

The problem in this case is that PROFILE II always puts new totals onto separate lines as:

```
100.00
                250.00
                        23.00
```

However, if you have or buy the PROFILE+ enhancement, PROFILE+ will put the totals on one line if it sees that the fields won't overlap.

PROFILE II users can add these new operations and fields to their program with the PROFILE+ upgrade. Radio Shack catalog numbers are 26-4515 (\$299.00) for PROFILE II+, and 26-4517 (\$120.00) for the PROFILE+ upgrade for PROFILE II.

Mr. Al Kessler of Riverbank, California, sent us a printout that explains how he squeezed 10 report formats out of PROFILE II, and could do even more—by renaming QUALITY2/MAP, KEY, DAT, DA2, DA3, PMP, PM2 and PM3 to QUALITY1/MAP, et cetera, before printing, then renaming the files to their original names afterwards. In the program, after all, you can only choose report formats 1 to 5!

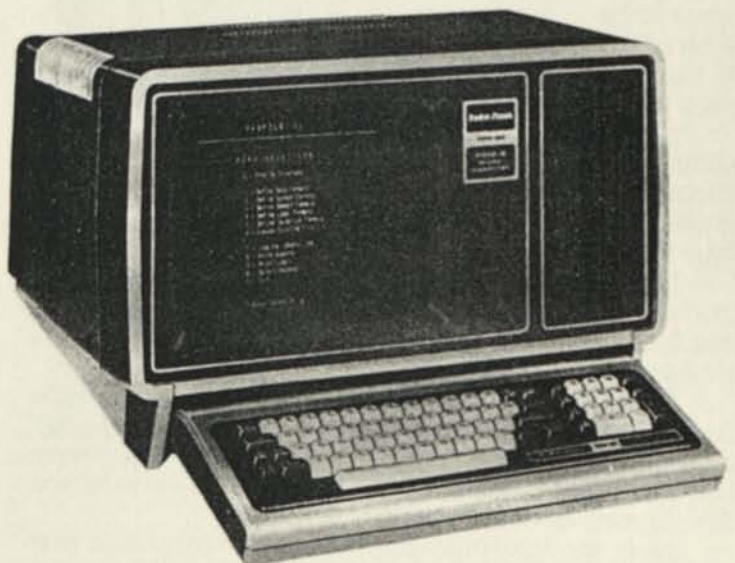
There is an easier way to do it. Set up report format number 5, and rename it to number 6 or A or whatever. Set up another format 5 and rename it number 7, B, et cetera. Then enter your user's menu—or write one—and in writing up the menu entry, list report format #6, #7, et cetera, instead of #1 through #5.

For instance:

```
1          PRINT SALES REPORT
PRINT/EFC (SALESMEN,6,PRINT SALES REPORT)
```

This bypasses PROFILE's normal 5 report limitation.

Write us at the address at the head of this column and let us know how you like this material and how useful it is to you.



Notes on Previous Issues

March, 1982

Double Dump to Tape (Page 5)

Serge Tremblay of Chateauguay, Quebec, Canada sent his version of "Double Dump":

```
20000 FOR N=1 TO 3: OUT 255,4: FOR M=1 TO 7000:  
NEXT M: CSAVE "A": NEXT N
```

The OUT 255,4 controls the remote control for the cassette.

Mark Young of Niangua, MO sent this version:

```
For X = 1 TO number of copies wanted:  
CSAVE "filename": NEXT X
```

Deltasin (Page 10)

Mr. David Sligar, author of Deltasin wrote (among others) to point out three errors in Deltasin. Please change these lines:

```
20 PI=3.14159  
41 IF Q>=1 THEN R=R+Q  
70 PSET ((X+140),(80+Y),1)
```

Gary Nance of Vallejo, California became so frustrated trying to correct Deltasin that he submitted his own version:

```
10 PHODE4,1:PCLS:SCREEN1,1  
20 FOR S=0 TO 12.56 STEP .07  
30 CIRCLE(S*20,96-SIN(S)*85),4,5,2,.13,.90  
40 NEXT  
50 GOTO 50
```

Relocating Machine Language Program (Page 11)

We received the following from Jerel Peterson:

Since I had just bought a disk-based system to replace my cassette drive Model III, I was happy to see an article on transferring machine language programs from tape to disk. I had tape versions of Scripsit and Eliza, as well as several other non-Radio Shack machine language tapes that I wanted on disk. When I saw the article, I figured on getting Scripsit and Eliza transferred, but not the others.

Following the DEBUG routine was easier said than done for someone who has trouble following directions. After many trials and errors, I finally got Scripsit to transfer to disk, but I was always getting an error message when I tried to do the same with Eliza. Finally, I gave up.

Later, as I was paging through the TRSDOS manual, I came across the TAPE command, which allows for direct transfer of machine language tape programs to disk. This looked much easier than going through the long DEBUG routine listed in Microcomputer News, so I decided to try it.

I wasn't really expecting it to work (if it would, why bother with a DEBUG routine) but it did! The program loaded right on to the disk as soon as the cassette stopped.

Getting the program to run caused a few problems. In BASIC, when I typed RUN"ELIZA", I got a "file not found" message. In TRSDOS, I also got an error message. Finally, I simply typed the filename, ELIZA, at TRSDOS Ready, and the program loaded up.

All of my non-Radio Shack machine language programs also loaded up and ran using this procedure, something that could not have been done by using published

procedure because the hexadecimal values were not available.

I also found that I could transfer Eliza and Scripsit (which were recorded at low baud rate) from disk to tape at the high baud rate. This saves quite a bit of time when loading by cassette.

I now have all my machine language tapes recorded on one disk. Rather than switching tapes and waiting several minutes for some of these programs to load up, I can switch from one program to another in about ten seconds.

CLEAR (Page 13)

Mr. Sheldon J. Kramer of Schaumburg, IL sent us the following:

The article on "CLEAR" in the March 1982 issue did not mention that when CLEAR is executed, it also does a RESTORE on both Models I and III. I have not seen this function of CLEAR described previously.

This feature was very useful when I combined several small programs into a larger menu driven program so that each section could sequentially process the same data. Since each section required different dimension (DIM) statements, the use of one CLEAR statement at the beginning of the menu routine eliminated the need for any RESTORE statements.

The following short program demonstrates this function:

```
10 CLEAR  
20 READ A$  
30 IF A$="END" THEN GOTO 10  
40 PRINT A$  
50 GOTO 20  
60 DATA ONE, TWO, END
```

When RUN, this program generates an endless series of ONE, TWO, ONE, TWO . . . on the screen. However, if in line 30, the transfer is changed to line 20 instead of line 10, then an out of data (OD) error occurs after the first TWO is displayed.

Model II Bugs, Errors and Fixes (Page 35)

Under the PATCHes for TRSDOS 1.2a, an O was inadvertently inserted into the second IODVRS/SYS patch. The O should have been a zero (0).

Pocket Computer Programs (Page 36)

Fred Nachbaur sent the following corrections to programs we ran in March:

TECH-3

```
12 F=1/(2pi*SQR(LC)): B$=" HZ"  
14 IF F>Exp3 LET F=F/Exp3: LET B$=" MHZ"  
22 L=1/((2piF)^2*C): LET B$=" H"  
30 INPUT "F=" :F  
52 F=1/(2piRC): B$=" HZ"
```

TRF-1

```
10 "A" PAUSE "TRF-1 PENTODE": D=60: J=1  
30 INPUT "R(P)=":O=OJJ  
140 PRINT "Q=":K: END  
150 "B" J=1: D=70: PAUSE "PENTODE BW": GOTO 20  
180 S=1/(1/O+1/N): IF D=160 LET  
R=1/(1/S+1/JJQWL): ...  
190 R=1/(1/S+1/(L/(CP)))  
195 B=JJ/(2piRC): ...
```

Sub Destroyer (Page 47)

We had numerous people point out an error in line 290 of Sub Destroyer. Line 290 should read:

```
290 IF A$="H" THEN K=K+2
```

Our thanks to the many people who wrote.

Additional Use of "@LOOKUP" in VisiCalc

Larry Scanlon
230 Lorraine Drive
Berkeley Heights, N.J.

I read with great interest the article in the March issue of MICROCOMPUTER NEWS entitled "New Commands and Functions for VisiCalc." Normally, when I read about enhancements to software that I already own I get very frustrated, as I wish I had these enhancements. I was not as frustrated as usual this time as I had already figured a way to incorporate "if" and "and/or" statements into my VisiCalc worksheets. I have version 1.20Z of VisiCalc, and run it on a Model I. This version contains the "@LOOKUP" function, and I have always been able to construct a relatively simple decision table to be used in combination with arithmetic logic that I build into the formulae to simulate IF, AND and OR statements that I wish to use. I have also always used tables to simulate the functions that "@CHOOSE" would allow. Obviously, these methods are a little cumbersome to set up, and cause the worksheet to grow (which can sometimes be a real problem), but they do give a great deal of flexibility to us users of older versions of VisiCalc.

COMMISSION

To illustrate the decision table and arithmetic logic I utilize, the following is how to use them in the same examples you gave in your article on the new functions: "A salesman's commission is 10% of sales; but if one month he sells over \$10,000, he gets a bonus of \$200.

	A	B	C
1) Sales	(Your Input)		
2) Commission	(B1 * .1)		
3) Bonus	/@LOOKUP(B1-10000,A8..C8)*200		
4) Income	+B2 + B3		
5)			
6)			
7)			
8) -10000	0		1
9) 0	0		1

Thus if Sales is \$10,000 or less the Bonus will be (0*200) or 0. If Sales is \$10,001 or more the Bonus will be (1*200) or 200. It should also be noted that the value of A8 should be smaller than the lowest possible value of B1-10000 or it is possible for "@LOOKUP" to return a value of "NA."

QUOTA

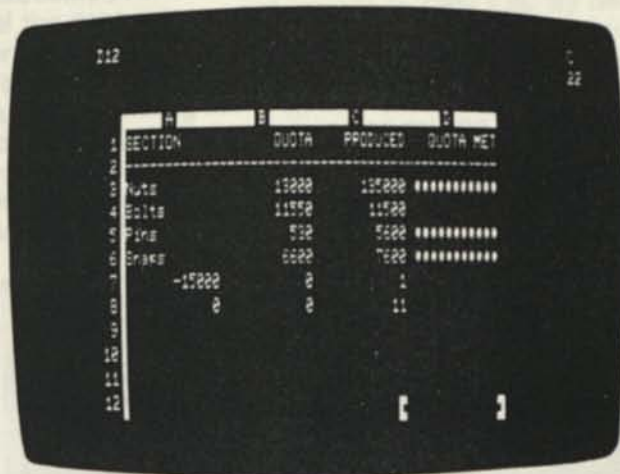
The example of each department exceeding its production quota is a little trickier. I use the "/F*" format command to print a row of asterisks when the statement is true rather than a number vs. 0 to indicate if the decision is true:

CORRECTIONS FOR THE MODEL II DIGITIZER PROGRAM

Some Model II program lines in the Digitizer manual need clarification. In the introductory paragraphs for Appendix A in the Digitizer manual, it is indicated that the shaded areas of the sample programs include the driver and initialization routines and would therefore have to be edited in a program editor to use the Digitizer in the program.

	A	B	C	D
1) Section	Quota	Produced	Quota Met	
2)				
3) Nuts	13000	135000	/F* @LOOKUP(C3-B3,A7...C7)	
4) Bolts	11550	11500	/F* @LOOKUP(C4-B4,A7...C7)	
5) Pins	530	5600	/F* @LOOKUP(C5-B5,A7...C7)	
6) Snaps	6600	7600	/F* @LOOKUP(C6-B6,A7...C7)	
7) -1500	0	1		
8) 0	0	11		

Note that the number 11 at C8 will cause 11 asterisks to be printed if quota is exceeded. Use of number 1 at C8 would result in one asterisk; 2 would yield two asterisks; etc. Simulation of "@OR" and "@AND" requires some slightly different arithmetic logic. Again, I use a field of asterisks to indicate a positive result.



BONUS

A salesman gets a bonus if his sales are over \$10,000 or if the number of units he sells is over 35 and his total sales are over \$6,000.

	A	B	C
1) Sales	(Your Input)		
2) Units	(Your Input)		
3) Bonus	/F* 15*(A6 + (A7*A8))		
4) -10000	0		1
5) 0	0		1
6) @LOOKUP(B1-10000,A4..C4)			
7) @LOOKUP(B1-6000,A4..C4)			
8) @LOOKUP(B2-35,A4..C4)			

Note that B3 is multiplied by 15 in order to assure that the column is all asterisks when the answer is true.

JULIAN

The example giving the calculation of a Julian Date as complicated by the function for leap years; however, was still relatively easy to do:

	A	B	C	D
1) Day =		(Your Input)	1	0
2) Month =		(Your Input)	2	31
3) Year =		(Your Input)	3	59
4)			4	90
5) Day of the			5	120
6) Year		(A18*A20) + A17	6	151
7)			7	181
8)			8	212
9)			9	243
10)			10	273
11)			11	304
12)			12	334
13) 0		1	3	
14) 0		0	1	
15) -5		0		
16) 0		1		
17) @LOOKUP(B2,C1...C12)+B1				
18) @LOOKUP(B2,A13...C13)				
19) (@INT(B3/4)*4)-B3				
20) @LOOKUP(A19,A15...B15)				

When this is calculated, it will return the Julian Date at B6. Note the placement of the first table in columns C and D rather than in rows as in the other examples. This allows the use of a lengthy decision table without causing the worksheet to use as much memory as if it were set up as two rows. Thus, a judiciously placed decision table and some arithmetic logic can simulate the more powerful of the new functions in VisiCalc without causing the worksheet to expand beyond the capacity of the system.



Digitizer

(Continued from Page 18)

```
8090 CLS
: PRINT@180, USING "LOCATION:
(###.##,###.##)"; X;Y
8100 RETURN
```

CORRECTIONS FOR THE MODEL II DIGITIZER PROGRAM

Some Model II program lines in the Digitizer manual need clarification. In the introductory paragraphs for Appendix A in the Digitizer manual, it is indicated that the shaded areas of the sample programs include the driver and initialization routines and would therefore have to be included in any program written to use the Digitizer. In the Model II program, line 150 and lines 360-370 should be included in the shaded area. In order to get a proper value back into the USR, the DEFINT I-N in line 150 has to be included and lines 360 and 370 are DATA lines which contain the code for the Machine Language routine that allows the data to be read from the Digitizer.

Additionally, there is an error in line 7060. The right arrow should be replaced with a ^ symbol for exponentiation. Line 7060 should read:

```
7060 DE=DE+J*16^(3-1)
```

Because in TRSDOS 2.0a SETCOM is a library command, after line 200 or 4030 (SYSTEM"SETCOM"...) is executed you will not return to BASIC to continue execution of the program but will remain in TRSDOS. By deleting lines 200 and 4030 from the BASIC program, initializing the RS-232 port by doing the SETCOM while still in TRSDOS, and then running the BASIC program, this problem should be eliminated. The error trapping routine to check RS-232 initialization in lines 190 and 9000-9030 will never be utilized because as soon as line 200 is executed you will be left in TRSDOS.



SuperSCRIPSIT and SCRIPSIT 2.0 — A Comparison

SuperSCRIPSIT for the Models I/III and SCRIPSIT 2.0 for the Model II are powerful word processing programs. There are many similarities in the two programs and some notable differences.

For Openers . . .



Screen 1—An example of a Model III SuperSCRIPSIT main menu.

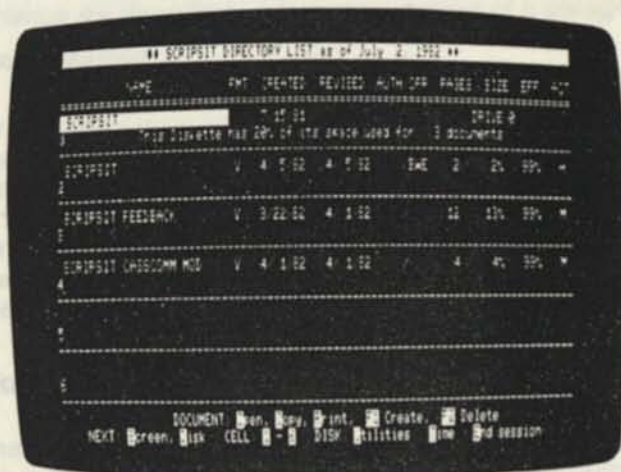
The first option (O) is selected to open an existing document or to create a new document. To get a disk directory select (D). The System setup utility will be accessed by pressing (S). System defaults for other menus (e.g., Printer options) can be changed by choosing one of the options provided through the system setup utility. (P)roof reading is used to check the spelling of a document. (C)ompress rewrites a document to a file using the least possible space. ASCII text conversion (converting an ASCII file to SuperSCRIPSIT or a SuperSCRIPSIT document to ASCII format) is done through option (A). (E) is self explanatory. (R) will return the user to the current document at the current page/cursor position.

Screen 2 shows the combined disk directory and main menu of SCRIPSIT 2.0. A lot of information about the document files including date created and last worked on, pages in length is displayed in this directory. At the bottom of the screen additional operations are indicated that can be initiated from the directory such as Open (open a file), Copy (copy a document to another file), Print (print a document), F1 Create (create a new document).

WRITING AND EDITING FEATURES

Cursor Movement

Both programs make it possible to move the cursor freely through text. Pressing any of the four arrow keys will



Screen 2—An example of a SCRIPSIT 2.0 directory or main menu

move the cursor in the direction of the arrow. The arrow keys are repeating in SuperSCRIPSIT while in SCRIPSIT 2.0 holding down the (REPEAT) key with the arrow key will cause the cursor to continue moving through text in the direction of the arrow until it reaches the beginning or end of the document.

SuperSCRIPSIT has the ability to define cursor movement by the next or previous word, paragraph, page, or occurrence of a search string. Cursor movement can be used to access the header (O or E) or footer (O or E) pages.

SCRIPSIT 2.0 uses (CTRL) (N) for next page, (CTRL) (P) to get the previous page, and the global utility to search for a word or string. Any page can be "gotten" or displayed on the screen by doing a (CTRL) (G) and entering the number of the desired page.

Editing Existing Text and Text Block Manipulation

Characters can be typed over, and text can be opened to allow insertions of characters, words, sentences, or blocks of text.

Both SuperSCRIPSIT and SCRIPSIT 2.0 include the ability to define a single character, a word, a sentence, a paragraph, or define text below the cursor as a block of text. In SCRIPSIT 2.0 a block of text can be defined above the cursor. In SuperSCRIPSIT a block can be defined across pages of text (more than one page of text at a time in a single block) while SCRIPSIT 2.0 allows a maximum of one page to be defined in a block of text.

Once the block of text is defined it is possible to:

- Reformat the block
- Move the block

- Delete the block
- Hyphenate the block
- Copy (duplicate) the block
- Lock the block (SCRIPSIT 2.0)
- Freeze the block (SuperSCRIPSIT)

By reformatting a block, the margins, indent tab, line spacing, tabs, etc. can all be changed. Moving the block means that a defined block can be moved from where it is to another document position either in the same or a different document. The copy block command is useful for duplicating text.

When a block of text is locked or frozen any reformatting that is done does not affect that block until it is unlocked or unfrozen. In SCRIPSIT 2.0 the block lock feature is used primarily to prevent a block of text, such as a table or chart, from being split up during a repaginate. If the block of text containing the table or chart is locked, then it will be printed in a block of text on one page. In SuperSCRIPSIT the block is literally frozen so that it cannot be edited. Printing will break at the usual page ending.

SuperSCRIPSIT has the option of defining a single block of text within a document and printing it without printing the rest of the document. In SCRIPSIT 2.0 a range of pages to be printed may be specified. In a 10-page document, page five only can be printed by entering the number five as the beginning and ending page number.

Both SuperSCRIPSIT and SCRIPSIT 2.0 allow one to delete a page, reorder or move pages. SCRIPSIT 2.0 has a utility to renumber pages. SuperSCRIPSIT is document oriented, which means that it looks at the document as one continuous block, while SCRIPSIT 2.0 is page oriented and looks at the document a page at a time.

Line Spacing

In SuperSCRIPSIT line spacing may be defined when the document is opened if it is a new document (one not already created), or if it is an existing document, then line spacing is changed by defining a block and changing the line spacing of the block. SCRIPSIT 2.0 document line spacing may be changed during a repaginate or in a defined block.

The line spacings available in the two programs are illustrated by the following chart.

Line spacing	SuperSCRIPSIT	SCRIPSIT 2.0	Blank lines b/t text
Single space	1	1	No blank lines
Double space	2	2	One blank line
Triple space	3	3	Two blank lines
Fourth line	*NA	4	Three blank lines
1½	1+	6	Half line
2½	2+	7	1½ lines
3½	3+	8	2½ lines
No space	NA	5	No space at all between lines

*NA — Not available in this program

Global Commands

Globally finding, deleting, or replacing a string of characters can be done very quickly in both programs. (Globally refers to every occurrence of the string in the document.) In SCRIPSIT 2.0 the global find, delete, and replace instructions are executed in a forward direction only. This means that in a global search for a string, the search begins at the current cursor position and searches

through to the end of the document for the string. SuperSCRIPSIT can also perform these functions in a reverse direction which means that the search can begin at the current cursor position and go back through text to the beginning of the document.

Paragraph Centering, Cascading Margins, and a Margin Command

SCRIPSIT 2.0 will center one line paragraphs where SuperSCRIPSIT will center each line in a multi-line paragraph. Cascading margins is a handy feature of SCRIPSIT 2.0 for entering text in an outline form. The margin command in SuperSCRIPSIT provides the ability to instantly change the indent tab, left or right margin of a paragraph.

Repagination

SuperSCRIPSIT automatically repaginates. The program keeps track of the line spacing and lines per page that are set in the open document options by either the default value or a value entered when the menu is on the screen. As soon as the last line of the page is typed in, the program begins a new page. In the status line at the bottom of the screen, the program advances the page number indicator by 1 and resets the line number indicator to 1. In SCRIPSIT 2.0 repaginate is a utility.

Format (Tab) Line

In SCRIPSIT 2.0 twelve format or tab lines (1-11 plus the default format line) can be pre-defined. Ten tab lines plus a default or system tab line can be pre-defined in SuperSCRIPSIT. There is a major difference between these two programs regarding the format line that is displayed when moving through a document. In SCRIPSIT 2.0 the last format line recalled or stored with the document is the one displayed. In SuperSCRIPSIT the format line for the current paragraph is displayed. Therefore, in SuperSCRIPSIT if the paragraph format changes from one paragraph to another then the format line displayed changes accordingly. The format line can be edited and tabs are set in the format line. The tabs remain active in the line for as long as it exists.

Align Tab in the Format (Tab) Line

An align tab is used to align data by a defined character. The decimal point is normally used as the align character. This means that a list of decimal numbers (or any other characters) can be aligned by a decimal point as they are entered. Consider the three numbers below for an example.

```

12.44
3000.11
    .12

```

The default align tab character is a decimal point (period) in both programs, but can be changed to another character (e.g., a space).

User-defined Keys

SuperSCRIPSIT allows up to 10 user-defined keys with up to 127 key strokes stored in each key. User-defined keys are very useful for storing a sequence of characters and/or instructions that are used frequently. Refer to View from the 7th Floor page 4 for an example of a typical letter signature block that might be stored in a user key. SCRIPSIT 2.0 allows up to 20 user-defined keys each containing

up to 256 key strokes. User-defined keys can be edited easily in SuperSCRIPSIT or SCRIPSIT 2.0.

Scrolling

Text can be scrolled horizontally and vertically. SCRIPSIT 2.0 scrolls horizontally in one character increments while SuperSCRIPSIT scrolls horizontally in eight character increments.

Automatic Word Wraparound

Both programs support automatic wraparound of text. If the word being typed exceeds the amount of space at the end of the line then the whole word is moved down to the next line, hence the term wraparound. This keeps the word intact and eliminates the worry of running out of space at the end of a line.

Forced End of Page

The end of a page of text can be forced in either program by inserting the proper symbol. When the symbol is encountered during printing the program advances the paper to top of form so that it's ready print at the top of the next page.

PRINTING FEATURES

Headers and Footers

Multiple headers and footers can be used in SCRIPSIT 2.0 while SuperSCRIPSIT allows one header and/or footer for even numbered pages and another for odd numbered pages or one header or footer for both even and odd numbered pages.

Background Printing

Background printing in SCRIPSIT 2.0 means that while one document is printing, the operator can work on a different document. This feature is not available in SuperSCRIPSIT.

Form Letters

A form letter is one that is printed repeatedly, and each time that it is printed, new values (e.g. name, address) are inserted in preset spaces.

SuperSCRIPSIT and SCRIPSIT 2.0 have the capability of creating a form letter (file) and a separate variables file (one which might contain the names and addresses) and then merging the two files together to print personalized letters.

User Print Codes

User print codes are used to embed code which contains instructions to the printer in a document. When the embedded code is encountered in the printing process, the printer executes the instructions stored in that code. This feature can be used to create special characters, issue back spaces, line feeds, etc. As an example, a user print code might be used to print a trade mark (™) symbol which cannot be entered directly from the keyboard.

Twenty-six SCRIPSIT 2.0 user print codes (A-Z) (up to six two-character hex codes) can be defined.

SuperSCRIPSIT provides for twenty user print codes, BUT each code can have up to eleven instructions stored in it. With the ability to store so many instructions, the user print keys can be quite versatile. Any of the numeric keys and shifted numeric keys (!, ", #, \$, %, &, ', (,), @, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0) can be assigned a print code.

Justification

SuperSCRIPSIT offers two methods of justification — Proportional and Monospaced. In proportional justification, the program inserts partial spaces between the words to fill out the line and even up the right margin. SuperSCRIPSIT supports true justification in the proportional print mode. Monospaced justification inserts whole spaces between words to fill out the line to even up the right margin.

SCRIPSIT 2.0 allows justification by three different methods — Space Insertion, Word Increments, and Character Increments. The space insertion method inserts full spaces between words to spread out the line. In order to spread out the line when using word increments to justify, the system adds a fraction of a space to the space between words. When using character increments a fraction of a space is added between characters to spread out the line. SCRIPSIT 2.0 does not right justify text when using proportional characters.

Drivers, Sheet Feeder, and Envelope Feeder

Both of these word processors make provision for the user to write a custom serial printer driver, and information is provided to assist in doing this. The sheet and envelope feeders for the Daisy Wheel II can be used with SuperSCRIPSIT and SCRIPSIT 2.0.

Pause Printing

A pause printing code within a SuperSCRIPSIT or SCRIPSIT 2.0 document halts printing of that document until the computer is instructed to resume printing. Pause printing makes it possible to change the ribbon, print wheel, etc., whenever desired in a document.

Screen Print

SCRIPSIT 2.0 also has a screen print function built into it. From anywhere in the program you can press (CTRL) and without releasing (CTRL) pressing (.) to print the screen contents.

More print features

Feature	SCRIPSIT 2.0	SuperSCRIPSIT
Superscripts	Yes	Yes
Subscripts	Yes	Yes
Underline	Yes	Yes
Double underline	Yes	Yes
Half line spacing	Yes	Yes
Headers/Footers	Yes	Yes
Multi-column print	Yes	Yes*

*While both programs support multiple column print, SuperSCRIPSIT handles a header differently than SCRIPSIT 2.0. On a SuperSCRIPSIT multi-column page the header is printed only once. SCRIPSIT 2.0 will print the header each time that a column of text is printed.

SPECIAL FEATURES

Double-Space Display Character

SuperSCRIPSIT displays a Δ (delta) character on the screen to indicate two spaces. Many typists type two spaces after the end of a sentence. When a sentence ends at the end of a video line, SuperSCRIPSIT uses delta to avoid beginning the next line with a space. The program

program also uses the delta to assure the best possible interline spacing for justified text. The delta can be overridden in order to type two or more spaces in a row.

Utilities

Utility	SCRIPSIT 2.0	SuperSCRIPSIT
Append (assemble) documents	Program feature	TRSDOS utility
Copy	Program feature	TRSDOS library command
Disk format/backup	Program feature	TRSDOS utility
Delete document	Program feature	TRSDOS utility
List directory to screen	Program feature	Names only †
List directory to printer	Program feature	TRSDOS library command
Clock display	Program feature	NA

†A directory containing file size and status, may be obtained through TRSDOS.

ASCII File Conversion

Conversion of a Scripsit file to an ASCII file or an ASCII file to a Scripsit file is available through either program. An ASCII formatted document can be transmitted out an RS-232 port for communication purposes. Another use of the ASCII conversion is to save a BASIC program in ASCII format, load it into Scripsit, edit it, convert it back to an ASCII file, and load and run it from BASIC.

Dictionary Programs

The capacities of the two dictionaries are different but both SuperSCRIPSIT and SCRIPSIT 2.0 can use a spelling dictionary program to proofread a document for misspelled words.

Model II Spelling and Hyphenation Dictionary contains 100,000 words and allows the user to add an additional 2,000 words to the dictionary. Hyphenation can be done automatically using the Spelling and Hyphenation Dictionary program.

The SuperSCRIPSIT dictionary for Model III contains over 73,000 commonly used words, and the Model I version contains 34,000. Both the Model I and Model III dictionaries allow the addition of up to 2,000 words to the existing dictionary. There is no automatic hyphenation feature in this program as there is in SCRIPSIT 2.0.



Screen 3— SuperSCRIPSIT System Setup Menu

Changing the System Defaults

The system defaults can be changed in either SuperSCRIPSIT or SCRIPSIT 2.0. What are system defaults? Certain values are predefined in both SuperSCRIPSIT and SCRIPSIT 2.0 which are used by the system when performing tasks. There is a default format line, page length, printer type, align tab character, etc. It is possible to change these defaults to the values that suit the individual's need. For example, in SuperSCRIPSIT the default for the type of printer is Daisy Wheel II. If the user has a Line Printer VIII, then the DW2 default would need to be changed to LP8.

HELP Menus and Comment Lines

A summary of program instructions can be viewed by the Help menus which are available in either program. There are seven Help screens in SuperSCRIPSIT and three in SCRIPSIT 2.0. SCRIPSIT 2.0 lets you enter a comment line in the text. Comment lines are not printed.

Disk Write Feature

In SuperSCRIPSIT (CONTROL) (W) writes the buffer contents to disk and empties the buffer. The buffer can hold up to 11,821 characters. This feature can be helpful where electric current is variable or unreliable and in documents that are unusually long. SCRIPSIT 2.0 writes the current page of text to disk each time another page is accessed.

Both Are Great Programs!

SuperSCRIPSIT and SCRIPSIT 2.0 are both great programs. Word processing has come a long way in the past few years and these two programs reflect the high standards and capabilities which should be present in the sophisticated word processors of today.

Comparison Chart of SCRIPSIT 2.0 and SuperSCRIPSIT

Print Features	SCRIPSIT 2.0	SuperSCRIPSIT
Background print	Yes	No
Print	Yes	Yes
User Print Codes	26 x 6	20 x 11
Merge	Yes	Yes
Justification	3 types (Character Incremental, Word Incremental, Space Insertion)	2 types (Monospaced, Proportional)
Superscript	Yes	Yes
Subscript	Yes	Yes
Boldface	Yes	Yes
Underline	Yes	Yes
Double underline	Yes	Yes
Multiple column print	Yes	Yes
Half-line spacing	Yes	Yes
Pause printing	Yes	Yes
Comment lines	Yes	No
Header/footer	Yes	Yes
Page numbers	Yes	Yes
Date/time	Yes	No
User Printer Driver	Yes	Yes
Sheet feeder	Yes	Yes
Print screen contents	Yes	No
Print single block	No	Yes
Print single page	Yes	Yes
Proportional printing with right justification	No	Yes
Double-space display char	No	Yes

Edit Features

	SCRIPSIT 2.0	SuperSCRIPSIT
Access Commands:		
Get page/block	Yes	Yes
Insert page/block	Yes	Yes
Delete page/block	Yes	Yes
Reorder page/block	Yes	Yes
Renummer page	Yes	No
Global find, delete, replace		
replace	Yes	Yes
Reverse direction	No	Yes
Text block manipulation:		
Define character	Yes	Yes
Define word	Yes	Yes
Define sentence	Yes	Yes
Define paragraph	Yes	Yes
Define above cursor	Yes	No
Define below cursor	Yes	Yes
Define across pages	No	Yes
Reformat block	Yes	Yes
Move block	Yes	Yes
Delete block	Yes	Yes
Hyphenate block	Yes	Yes
Lock/Freeze block	Yes	Yes
Character insert/delete	Yes	Yes
Repagination	Utility	Automatic
User-defined keys	28 x 256	10 x 127
User-defined keys editor	Yes	Yes
Paragraph Format	Yes	Yes
Format line display	Last line displayed	Line for current paragraph displayed
Unique file format		
ASCII file conversion	Yes‡	Yes‡
Decimal aligned tabs	Yes	Yes
Outline mode (indentation)		
Word wrap	Yes	Yes
horizontal scrolling	One character increments	Eight character increments
Vertical scrolling	Yes	Yes
Full video mode	Yes	Yes
Full cursor control	Yes	Yes
Center/uncenter paragraph	Yes	Yes
Help	Yes	Yes
Format line editing	Yes	Yes
Forced end of page	Yes	Yes
Recall text block	Yes	Yes
Local (quick) search	Yes	Yes
Cascading margins	Yes	No
Proportional spacing with right justification	No	Yes
Utilities		
Spelling Dictionary	Yes	Yes
Automatic Hyphenation*	Yes	No
Append (assemble) documents	Yes	No†
Copy	Yes	No†
Disk format/backup	Yes	No†
Delete document	Yes	No†
Change system defaults	Yes	Yes
List directory to screen	Yes	Yes
Compress document	No	Yes
Clock display	Yes	No
Return to current document at current page/cursor position	No	Yes

*This feature is available in the SCRIPSIT 2.0 Hyphenation and Spelling Dictionary.

‡The document file formats of SCRIPSIT 2.0 and SuperSCRIPSIT are not compatible.

†These are available as TRSDOS utilities and can be done from TRSDOS Ready.

Model I/III Bugs, Errors, and Fixes

Note to Users:

The following program changes and corrections are provided for your information. If you have an applications program which is working correctly, you should probably NOT make any changes to it. If you feel that the changes should be made, but you do not feel qualified to make the changes yourself, please contact your local Radio Shack Computer Center or Expanded Computer Department for assistance. If you do not have access to one of these stores, then you may want to call Computer Customer Services in Fort Worth for assistance.

Changes to BASIC programs

There are general procedures that need to be followed when any corrections are made.

1. Make a backup of the tape or disk that contains the program to be corrected. Changes should be made on the backup copy.

2. Load the program to be changed by typing CLOAD"filename" (tape) or LOAD"filename" (disk) where filename is the name of the program to be modified.

3. Make the line changes indicated in the fix. For existing line numbers, edit or retype the line to match the one in the fix. Enter new lines.

4. Save the corrected program (the one now in memory). Type CSAVE"filename" (ENTER) (tape) or SAVE"filename" (ENTER) (disk) where filename is the name of the program that has been modified.

5. Now make a backup of the corrected tape or diskette.

PATCHES

PATCHes are entered from TRSDOS READY and are used to make corrections to files stored on the disk.

1. Before making a PATCH, back up the diskette that requires modification and make the PATCHES to the backup copy of the diskette.

2. Apply PATCHES according to the information given in your TRSDOS manual.

CASSETTE PORTFOLIO (26-1506)

In Model III Version 3.0, Sell Transactions may not exactly match the original purchase if fractional units are involved. This results in the lot being left on file (i.e., units left after sell are not exactly equal to zero).

The problem is corrected by making the following corrections to the program TTRAN.

```
123 ...IFD!>S(J,0)+.005THEN116
126 ...IFS(J,0)<.005THENGOSUB132
```

The ellipses (. . .) indicate that the rest of the line up to that point remains as is.

STOCKPAK (26-1507 Version 3.0)

The Model III program demonstrates erratic performance on printed reports (syntax errors, reboot, etc.). The problem is corrected by making the following change to the program PTREPORT.

```
350 IFPEEK(16425)>50 THENPRINTCHR$(12);
:POKE16425,1:Q%=Q%+1ELSERETURN
```

Add the following line:

```
7 POKE 16412,255
```

Add the following lines to the program REPRUN/BAS.

```
5 POKE 16412,255
1503 POKE 16425,1
```

ACCOUNTS PAYABLE (26-1554)

Following are two sets of corrections for Model I Accounts Payable (26-1554). The first corrections concerns adding vendors to APS, and the second set of procedures for Model I users concerns the ability to print out a preview of checks.

1 — When vendors are deleted in 3.0 and prior versions of AP, the system does not deduct the deleted vendors from the total number of vendors so more vendors may be added.

Make changes to the appropriate version of the program APS.

MODEL I VERSION 3.0

```
161 P1=0:J=VN:IFPV=VN-1-VD THENPL=704:W1$=
"VENDOR FILE IS FULL"+STRING$(24," ")
:GOSUB121:GOTO157ELSETN=0:FORW9=1TO7
:V$(W9)="":NEXT
```

MODEL I VERSIONS PRIOR TO 3.0:

```
510 P1=0:J=VN:IFPV=VN-VD THENPL=704:W1$="VENDOR
FILE IS FULL":GOSUB430:GOTO500ELSEVV=0
:FORW9=1TO7:V$(W9)="":NEXT
```

2 — In the Model I versions, if you cannot print out a preview of checks it is because you have not selected an invoice to print a check for. (All Versions)

Symptom = If the CHECKS MENU (P) option is used and nothing happens, then execute the following steps.

1. Press the @ key — this returns program to MAIN MENU.
2. Press the (I) key — This loads the invoice selection program once MENU appears.
3. Press the (S) key — to select invoices.
4. Invoices must be posted, then one of the options: (A) option for ALL INVOICE or (V) option for VENDOR/INVOICE or (D) option for DISCOUNT/DUE DATE must be used to select invoices.

If any invoices have been put on hold and you wish to include them in the selection process, they must be released first.

If DISCOUNT DATE is chosen and the checks printed are different from that date, it is caused by a invoices being entered with a discount date of 00/00/00.

ACCOUNTS RECEIVABLE (26-1555)

Status Code 4 (over 90 days overdue) does not return to Status Code 1 (current balance due) when previous balance is paid. (Model III Version 3.0 and prior)

If you have Model III versions 3.0 or prior, please contact your local Radio Shack Computer center or Store. For version 3.1 order stock #700-2217 (proof of purchase required).

DISK PAYROLL (26-1556 Version 2.0)

In Model III Version 2.0, you cannot exempt employer from FICA on tips. The program will appear to accept the exemption, but will double employee amount when showing FICA payable on Journal.

If you are running Model III Payroll with a version number prior to 3.0, contact your local Radio Shack Computer Center or Store for Version 3.0 (stock #700-2221, proof of purchase required).

MANUFACTURING INVENTORY CONTROL (26-1559)

1 — In Model I MIC, it has been found that a problem can occur in Finish Good Maintenance which causes the machine to lock up when updating the files if more than 300 raw materials are in the finished good. (All versions prior to 3.0)

Version 3.0 of Manufacturing Inventory Control is available through your local Radio Shack Computer Center or Store (stock #700-2223, proof of purchase required).

2 — In Model I version 1.2, if Drive 4 (Data Disk) is not ready on entry, the Raw Material Use Report will be inaccurate.

Version 3.0 of Manufacturing Inventory Control is available through your local Radio Shack Computer Center or Store (stock #700-2223, proof of purchase required).

DISK SCRIPSIT (26-1563)

The following PATCH to Model III Disk Scripsit, Version 3.2 on TRSDOS 1.3 will allow documents created under Tape Scripsit at 1500 baud to load.

```
PATCH SCRIPSIT/CMD (ADD=530A,FIND=1142,CHG=0000)
```

MEDICAL OFFICE SYSTEMS (26-1568)

Below are corrections for two problems with Medical Office System.

PROBLEM 1 — Medical Office System contains a bug which may cause problems if more than three patient data files are used. (Model I/III, Version 1.0)

Changes to Medical Office Systems are somewhat critical. Make any changes carefully.

Step 1 — Make the following changes to the program ALPHASOR.

```
1240 FZ=INT((AC-1)/Z1):IFFZ<>FOTHENFO=FZ:CLOSE3
:FMS="PDATA"+CHR$(48+FO)+"":3":NU=3:GOSUB400
:CLS
1250 LR=AC-FO*Z1:PR=INT((LR-1)/2)+1:SR=LR-2*
INT((LR-1)/2)-1
```

Step 2 —

```
SAVE "ALPHASOR"
```

Step 3 — Next add the following lines to the program named MENU.

```
120 FZ=INT((AC-1)/Z1):IFFZ<>FOTHENFO=FZ:CLOSE3
:FMS="PDATA"+CHR$(48+FO)+"":3":NU=3:GOSUB400
140 LR=AC-FO*Z1:PR=INT((LR-1)/2)+1:SR=LR-2*
INT((LR-1)/2)-1
1400 DEFFNA(X)=PEEK(X)+PEEK(X+1)*256:E=
FNA(6H40A4)
```

```
1410 S=FNA(E):IFFNA(S+2)<1000THENE=S:GOTO1410
1420 PRINT"TP=TP+S-FNA(&H40A4)"IN LINE 1110":END
```

Step 4 — Next type:

```
GOTO 1400 <ENTER>
```

(The computer will pause and print TP = TP + NNNN in line 1110. The NNNN will be a number.)

Step 5 — Type:

```
LIST 1110 <ENTER>
```

(If the number after TP = TP + ... does not match the computer generated number from Step 4, line 1110 must be edited to contain the new value.)

Step 6 — Type:

```
EDIT 1170 <ENTER>
```

Line 1170 should look like this:

```
1170 DATAMORSORI,PRINT STATEMENTS,BLLPRT,SET UP
SYSTEM DISKS,INIT,16922,16923,16924,1390,
4170,3,16452,16453,16454,660,3960,6
```

If your line 1170 does not match the above line 1170, then edit your line 1170 to match.

Step 7 — Type:

```
SAVE"MENU" <ENTER>
```

Step 8 — Format a new data disk for each existing patient data disk plus any new patient data disks that you might want to add.

Step 9 — Type in the PROGRAM below:

```
10 CLEAR2000:CLS:AC=1:FO=-1:B2=-1
15 IFPEEK(293)=73THENZ1=1390ELSEZ1=660
20 D$(1)="OLD":D$(2)="NEW"
30 PRINT@320,"WORKING ON RECORD #";
40 IFFO<>INT(AC/(Z1+1))THENCLOSE1:FO=
INT(AC/(Z1+1)):D=1:GOSUB110:OPEN"R",1,"PDATA"+C
HR$(FO+48)+":1"
50 PRINT@339,AC
60 LR=AC-FO*Z1+1:PR=INT((LR-1)/2)+1:SR=LR-2
*INT((LR-1)/2)-1
70 FIELD1,SR*127ASA1$,127ASA$:GET1,PR
80 B1=INT((AC-1)/Z1):IFB1<>B2THENB2=B1:
CLOSE2:FMS="PDATA"+CHR$(48+B1)+":2":D=2:GOSUB11
0:OPEN"R",2,"PDATA"+CHR$(48+B1)+":2"
90 B=AC-B1*Z1:B3=INT((B-1)/2)+1:B4=B-2*
INT((B-1)/2)-1
100 FIELD2,B4*127ASB1$,127ASB$:LSETB$=A$:
PUT2,B3:AC=AC+1:GOTO40
110 IFAC>1THENM$="NEXT"
120 PRINT@1,"PUT "D$(D)" PATIENT DATA
DISK"FO+1"IN DRIVE"D
130 PRINT:PRINT"PRESS <ENTER> WHEN READY, OR <@>
TO END"
140 W$=INKEY$:IFW$=""THEN140
150 IFW$=CHR$(13)THENRETURNELSEIFW$="@THEN
CLOSE:ENDELSE140
```

Step 10 — Type:

```
RUN <ENTER>.
```

Follow the prompts, inserting old and new disks as requested. These new disks will be used with the modified MOS disk. DO NOT attempt to use the old patient data disks with the modified program.

PROBLEM 2 — In Model III Version 1.0 if you're using a password of fewer than eight characters, there is a possibility that files may be improperly opened on drive 0.

The problem is corrected by following the steps listed below.

Make the following corrections to the program named MENU.

```
590 GET1,3:NF=CVI(NF$):DT$=DAS:PW$=MID$(PZ$,1,
INSTR(PZ$+" ","")-1):Z1=CVI(Z1$)
:Z2=CVI(Z2$):Z3=CVI(Z3$):DP$=Z4$:BP$=Z5$
:YN$=Z6$:CLOSE1:RETURN
```

Examine lines 1350-1360. They should look like this:

```
1350 DEFFNA(X)=PEEK(X)+PEEK(X+1)*256:E=FNA
(&H40A4)
1360 S=FNA(E):IFFNA(S+2)<1000THENE=S:GOTO1360ELSE
PRINT"TP=TP+S-FNA(&H40A4)":END
```

If they do not look like this, add them as shown.

Type:

```
GOTO 1350 <ENTER>
```

The computer will respond with TP = TP + NNNN (where 'NNNN' is a number).

Edit Line 1110 so that the statement TP = TP + NNNN equals the number you just got from the computer. (Line 1110 looks like this before the change.)

```
1110 CLS:PRINT@467,"** LOADING PROGRAM
**":GOSUB570:GOSUB580:TP=PEEK(16548)+PEEK
(16549)*256:TP=TP+3836:POKE16424,
PEEK(16548):POKE16425,PEEK(16549)
:POKE16549,CINT(TP/256):POKE16548
TP-256*CINT(TP/256):RUNPG$(PR)
```

Finally,

```
SAVE"MENU"
```

with all the changes.

ADVANCED STATISTICAL ANALYSIS (26-1705)

Rounding errors appear in the standard deviation output in the Chi Square Analysis program. (Model I/III Version 3.0 & Prior)

The problem is corrected by following the steps listed below.

CLOAD the Chi Square Analysis Tape.
Make the following change:

```
280 FORI=1TONR:FORJ=1TONC:CS=CS+(ABS(O(I,J)
-E(I,J))-CC)^2/E(I,J)
```

Type:

```
CSAVE"A"
```

to save the changes in the program.

NOTE: The exponentiation operator "" is made by typing the up-arrow, but on the Model III, it will appear on the screen as a left bracket rather than an up arrow.

COBOL (26-2203)

When using the TAB option with the ACCEPT statement in an indexed file, improper program halts occur, contrary to the usage described in pages 139-144 of "RSCOBOL" in manual 26-2203. (Model I, Version 1.3b)

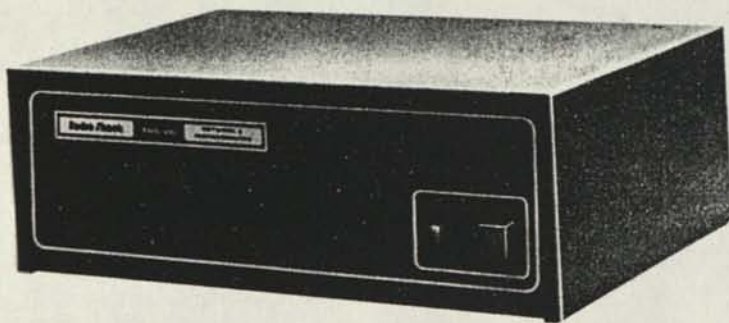
To correct the above problem, APPLY the following patch by typing the following line.

```
PATCH RUNCOBOL/CHD (ADD=AE61,FIND=C3F6A9,
CHG=AFC900)
```


● NETWORK 3—New Capabilities for the Classroom

Radio Shack's new Network 3 Controller gives non-disk student stations most of the capabilities of disk-equipped systems. Using the Controller, a teacher can create a classroom computing network by connecting as many as 16 TRS-80 Model III non-disk systems to one Model III disk system. The disk system acts as a "host" computer and the non-disk systems act as "student stations." This new Network 3 system gives the non-disk student stations the following special disk-related capabilities:

1. Stations can load and save programs at the student station keyboard, using the disk at the "host" system.
2. Stations can use a line printer attached to the "host" without interrupting other students or the teacher.
3. When running student management programs, stations can automatically send information such as student scores to the host disk. This allows class records to be kept in one location.
4. Student stations can let computer science students learn and use Disk BASIC on non-disk Model III systems.



THE DIFFERENCE BETWEEN NETWORK 3 AND NETWORK 2

The Network 3 Controller differs in important ways from Radio Shack's current Network 2 Controller and the earlier Network 1 model. Network 2 and Network 1 systems are cassette-port based. Active commands come from the "host" computer. The teacher loads a program into the central disk-based host and then sends the program through the network to the non-disk stations. One program can be loaded into all of the student stations at once, or different programs can be loaded into different stations. Once a program is loaded into a student station, the student can work independently with the program at his or her own pace.

The major advantage of the Network 2 Controller is its ability to save time for the teacher and help organize classroom use of computer resources. Network 2 centralizes

software management, keeps control of loading programs in the hands of the teacher, and gives the classroom the increased efficiency of the disk system at low cost.

The Network 3 system, in contrast, is an RS-232 based network. Most active commands come from the student stations. The teacher determines what choices (if any) will be available to the student by putting the proper program diskettes into the host's disk system. Then the student loads the program into his or her station by entering the necessary commands at that station's keyboard. Students can save and load programs directly to and from the disk, open and close data files, and print out program listings and other information. Student management data can be sent to the host disk for certain programs.

If you want your network system to be used by programming students or other students who will need to print out information, by students who are working with a student management program, or by a group of students who will be working independently with a wide variety of disk or non-disk programs, the Network 3 Controller will provide the flexibility and classroom efficiency you are looking for.

The Network 3 Controller performs more advanced networking functions than the Network 2 Controller but does not replace Network 2. The two controllers do different things. Your own school's individual needs will determine whether you'd benefit most from a Network 2 Controller, a Network 3 Controller, or a configuration that would allow you to use both as needed.

Both Network Controllers allow each participating TRS-80 student station to be used as an independent, cassette-based, stand-alone unit in addition to acting as part of the network. This flexibility helps make the network an ideal configuration for the classroom.

Now let's look a little more closely at the student station capabilities available through Network 3.

WORKING WITH DATA FILES AT NETWORK STUDENT STATIONS

With the Network 3 system and Network 3 operating software, students can open data files from student stations, put data into the files and get data from them, and close the files. Different stations can work with different data files simultaneously.

PRINTING FROM A STUDENT STATION AND "SPOOLING"

Students working at student stations can send printing requests to the host computer. From there, the material to be printed is sent to the printer attached to the host.

If more than one student station sends a printing request at the same time, a function called "spool" will store the material to be printed until the printer can get to it. Since the spool accepts information faster than the printer can print it, the student station can return to its program without delay.

If students will be printing out small amounts of material, you may want to keep the printer off during the class period and allow the "spool" storage space to fill up as students place their printing requests. Near the end of the class period, you can turn on the printer and print out all of the material at once. This technique will keep printer noise from interrupting other activities.

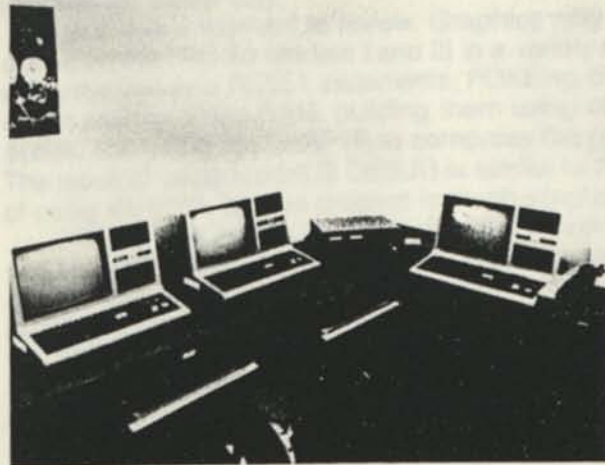
STUDENT MANAGEMENT

With Network 3, it is now possible to use Radio Shack's K-8 Math With Student Management Program in a network configuration. If you use 32K student stations (required by the K-8 With Student Management Program), you can load and run the program normally on a student station just as if you were running the program on a disk-based computer. At the end of each session, the system transfers student scores from the student station to the host computer, where they are stored on the K-8 Math Program diskette.

To view the scores stored on the diskette, the teacher types a special code at any student station which is running the K-8 Math Program. Student scores will appear on the screen at that station.

PROGRAMMING IN DISK BASIC AT THE STUDENT STATION

Numerous Disk BASIC commands are available at the student stations. The Network 3 Operating Software User's Manual contains a "quick reference" section that lists and describes the available commands.



EQUIPMENT REQUIREMENTS

The minimum "host" computer for the Network 3 system is a TRS-80 Model III with 32K of memory, one disk drive, and an RS-232C interface. A "student station" needs a minimum of 16K of memory, Model III BASIC, and an RS-232C interface. The Network 3 Controller requires one cable for each student station (10 to 100 feet in length) and one cable for the host system. The current operating software (NETWORK 3 OPERATING SOFTWARE, Cat. No. 26-2775) is designed for Model III only. Future versions will permit TRS-80 Model I, Model III, or Color Computer systems to be used as student stations and will permit a Model II (with hard disks if desired) or a Model III to be used as the host computer.

When deciding how much memory you'll need for your student stations, it is important to consider what kinds of programs you'll be running. For computer science applications, where students are writing their own programs, 16K may be sufficient. However, most packaged courseware programs will require a larger memory size. The reason for this is that the Network's operating software takes up 9.5K of memory in each student station, leaving only 6.5K of available memory in a 16K machine. Radio Shack's educational software packages will thus require 32K student stations.

LIST OF RADIO SHACK COURSEWARE FULLY COMPATIBLE WITH THE NETWORK 3

Many Radio Shack software packages are fully compatible with the Network 3 (that is, no modifications to the program are necessary). The packages which are fully compatible include:

- K-8 Math Program (Cat. No. 26-1715)
- K-8 Math with Student Management Program (26-1725)
- Essential Math Programs, Volumes One and Two (26-1716 and 26-1719)
- Euclid Geometry Tutor (26-1724)
- Advanced Graphics (26-1714)
- Vector Addition (26-1720)
- Graphical Analysis of Experimental Data (26-1722)
- Interpreting Graphs in Physics (26-1721)
- Investigations in Integral Calculus (26-2600)
- Plane Analytic Geometry (26-2602)
- AlphaKey (26-1718)

Again, please note that 32K of memory in the student stations is required to run these programs in the Network 3 configuration.

SPECIAL FEATURES OF THE NETWORK 3 OPERATING SOFTWARE

Included on the Network 3 Operating Software diskette is a BASIC program called "MENU." You can use this program to set up one or more menus that allow students to choose what programs they'll run. Options in the menus correspond to the filenames of BASIC programs that are stored on a lesson diskette at the host computer. When a student selects one of the options you've provided, the menu program automatically runs the chosen program at that student's station.

A demonstration program called "MAILBOX" is also furnished on the Network 3 diskette. It allows students to send messages to one another from the student stations.



These messages are stored on the Network 3 diskette at the host computer. The teacher (working at a student station) can post a "bulletin board" message, or a message that will be seen at all student stations. These bulletin board messages cannot be erased by the students.

HOW YOU CAN USE THE NETWORK 3

Network 3 hardware and operating system software is available through your local Radio Shack store or Computer Center. The listed price of the Network 3 Operating System software (Cat. No. 26-2775) is \$149.00. (Prices, however, may vary at individual stores and dealers.) For more information on Network 3 operation and educational applications, contact your local Radio Shack retailer, or contact the Radio Shack Regional Educational Coordinator in your area.

Compress Graphics Using Model III DEBUG

Dennis F. Tanner
Ft. Worth, Texas

For the May, 1981 Microcomputer News I wrote an article on using the VARPTR instruction to compress graphics on the Model I and Model III disk and non-disk systems. DEBUG, a utility included in Model III TRSDOS, provides an easier way.

Let's take a moment to review. Graphics may be produced on the TRS-80 Models I and III in a variety of ways: using the SET and RESET statements, POKEing character codes into the video RAM, building them using character codes, and using the VARPTR to compress the graphics. The result of using Model III DEBUG is similar to the result of using VARPTR, but the process is much simpler.

DEBUG is a powerful TRSDOS utility which lets you examine, edit, and (if you don't use it properly) really mess up disk files. For this reason, I recommend that you use a BACKUP copy of a TRSDOS disk to learn this process.

On your Model III disk system, go into BASIC and type in the following program:

```
10 A$=" " ' 8 SPACES
20 PRINT"LENGTH=" LEN(A$)
30 PRINT"A$=" A$
```

Type:

```
RUN <ENTER>
```

to run the program once to make sure you have 8 spaces inside the quotes in line 10. Then save it in ASCII format with this instruction:

```
SAVE"DEMO",A <ENTER>
```

Return to TRSDOS with this instruction:

```
CMD"S" <ENTER>.
```

At "TRSDOS Ready" go into DEBUG by typing the instruction:

```
DEBUG <ENTER>.
```

You will see the contents of some memory locations, the contents of the locations following the program counter, and the contents of the various registers. The cursor will be blinking in the lower left-hand corner. Type:

```
F
```

The "Filespec:" prompt will appear. Type:

```
DEMO <ENTER>.
```

You are now looking at the contents of the disk file. Since the file was saved in ASCII format, the program is legible on the right side of the display. The center area shows the hex (base 16) value of each byte in the program. "31" represents 1 in the line number "10," "30" represents 0, "20" represents the space, and so on.

You can now modify this disk file in DEBUG. Type:

```
M
```

A graphics block now covers the "31" in the upper left hand corner. To compress some graphics, we want to modify the contents of the quotation marks in program line 10. Using the arrow keys, position the cursor over the first "20" in the center of the top line of the screen. We will change the string of 8 spaces to a string of 8 graphic characters. Let's use graphics character 191. This is represented as "BF" in hex numbers. To replace the spaces, type:

```
BFBFBFBFBFBFBFBFB
```

Notice that the display on the right side now shows the graphics inside the quotation marks. To make this change on the disk, press:

```
<ENTER>
```

When the disk drive stops, press:

```
<BREAK> <BREAK>
```

to return to TRSDOS. Go into BASIC and type:

```
LOAD"DEMO" <ENTER>
```

When you list the program, you will see that the graphics are indeed stored in the program. Run that program to demonstrate that this process works.

Let's try some fancier graphics. Type in line 10 like this:

```
10 A$=" " ' 27 SPACES
<ENTER>
```

Run the program once to make sure the length is 27. Then type:

```
SAVE"DEMO1",A <ENTER>
CMD"S" <ENTER>
DEBUG <ENTER>
F
DEMO1 <ENTER>
```

We are going to make a graphic box with character 191 (BF) on the sides, character 131 (83) on the top row, and character 176 (B0) on the bottom row. The tricky part of this is, some special characters are needed to align the characters properly. Let's begin. Type:

M

Position the cursor over the first "20" in the center of the top line. Type:

BF838383BF

to make the top row. Character 26 (1A) moves the graphics position to the next row down in the finished graphic. Type:

1A

to position it properly. (A period appears on the right side of the screen because this hex code has no ASCII representation.)

Now the graphic position must be moved back five spaces with the backspace character 8 (08) to align it under the beginning of the previous line in the finished graphic. Type:

0808080808

The middle row of this 3-line graphic will have character 191 (BF) on each end; in the middle it will have spaces, character 32 (20). Type:

BF202020BF

To align the third row under the second row, type:

1A0808080808

Don't worry if the graphics on the right side of the screen look funny. If you have typed the codes correctly, the graphics will turn out right.

To create the bottom row, use character 191 (BF) and character 176 (B0). Type:

BF808080BF

To save these changes, press:

<ENTER>

When the drive goes off, type:

<BREAK>

<BREAK>

BASIC <ENTER>

<ENTER>

<ENTER>

LOAD "DEMO1" <ENTER>

When you list it, the result may surprise you. The graphics box appears in the listing. Run the program to see again what is stored in A\$. If you want to insert a number in the box, you can do that in BASIC. Type:

EDIT 10 <ENTER>

17 <SPACE>

C5 <ENTER>

Have fun with DEBUG and graphics!

Poor Man's Editor

Dallas L. Powell
1804 Skyland Drive
Tallahassee, FL 32303

This routine for a 16K Model III and a Line Printer VI was used to write this letter. I call it my Poor Man's Editor.

It needs refinements, but I am a novice with BASIC. (I received the system in October.)

```

100 ' *** E D I T O R *** 11/27/81 REV 22
110 CLEAR 800
: CLS
: POKE 16427,62
120 DIM A$(10)
130 A$(1)=CHR$(8)
: A$(2)=CHR$(9)
: A$(3)=CHR$(10)
: A$(4)=CHR$(13)
: A$(5)=CHR$(32)
: A$(6)=CHR$(64)
: A$(7)=STRING$(8,A$(5))
: A$(8)=CHR$(95)
140 CLS
: INPUT "NEW PRINT <Y> OR <N> ";Y$
150 IF Y$="Y" THEN B$=""
: ELSE 370
160 CLS
170 PRINT@ 900,"Type less than 241 characters: To end
type @"
180 FOR I=1 TO 23
: SET( 83,I)
: SET(127,I)
: NEXT I
190 '
200 I$=INKEY$
210 FOR I=1 TO 6
: IF A$(I)=I$ THEN 240
220 NEXT I
: GOSUB 410
: GOTO 200
230 '
240 ON I GOTO 270, 280, 290, 290, 350, 370
250 ' BK-SP TAB DN-AW ENT SP "0"
260 '
270 Z1=LEN(B$)
: Z2=Z1-1
: C$=LEFT$(B$,Z2)
: B$=C$
: GOSUB 420
: GOTO 200
280 B$=B$+A$(7)
: GOSUB 420
: GOTO 200
290 Z1=LEN(B$)
300 IF Z1 < 64 THEN Z3=64-Z1
: GOTO 340
310 IF Z1 < 128 THEN Z3=128-Z1
: GOTO 340
320 IF Z1 < 192 THEN Z3=192-Z1
: GOTO 340
330 IF Z1 < 241 THEN Z3=241-Z1
: ELSE 370
340 Z4$=STRING$(Z3,A$(5))
: B$=B$+Z4$
: GOSUB 420
: GOTO 200
350 B$=B$+A$(5)
: GOSUB 420
: GOTO 200
360 '
370 CLS
: PRINT B$
: INPUT "HARD COPY <Y> OR <N> ";Y$
: IF Y$="Y" THEN INPUT "Large or small (L) or
(S)";L$
: ELSE 140
380 IF L$="L" THEN LPRINT CHR$(31)
: ELSE LPRINT CHR$(30)
390 LPRINT B$
: GOTO 140
400 '
410 B$=B$+I$
420 Z1=LEN(B$)
: Z2=Z1+64
: IF Z1=241 THEN 370
430 IF Z1 > 240 THEN 370
440 PRINT@ 64, B$
: PRINT@ 22, A$(8)
: RETURN
450 END

```

● Scripsit Ideas

XERCH

Peter D. Beeson
250 Central Building
Seattle, WA 98104

I use the Model II SCRIPSIT 2.0 Convert Utility with the Dow Jones News Service with the intention of receiving text data over the wire and then editing the data with the SCRIPSIT program. When I first tried to do this, during the process of having the SCRIPSIT load the ASCII file created by the TERMINAL Utility of TRSDOS, I kept receiving an error message before the entire ASCII file was loaded into SCRIPSIT.

I found that the error message resulted from embedded 1E's and 1F's (Hex Notation) that were transmitted along with the text from the Dow Jones News Service. Attached is a short BASIC program that corrects the problem. This program can be executed directly from the "S" option on the TERMINAL menu by typing "BASIC XERCH - F:1". After the corrections have been made, the program immediately loads SCRIPSIT. The SCRIPSIT disk must be in drive 0 and the TRSDOS disk must be in drive 1.

XERCH

```

100 '
110 '
120 ' *** XERCH ***
130 '
140 '
150 ' THE FOLLOWING PROGRAM SCANS AN ASCII FILE FOR
160 ' 1E'S AND 1F'S (HEX NOTATION) AND REPLACES THEM
170 ' WITH A CARRIAGE RETURN SO THAT THE ASCII FILE
180 ' CAN BE LOADED INTO SCRIPSIT 2.0
190 '
200 '
210 CLS
    : PRINT
    : PRINT
    : PRINT
    : PRINT
215 PRINT@ (10,10), "TYPE IN THE NAME OF THE ASCII
    FILE THAT"
216 PRINT@ (11,10), "IS TO HAVE THE 1E'S AND THE 1F'S
    REPLACED:"
217 PRINT
    : PRINT
220 LINE INPUT WDS
230 PRINT@ (17,10), "WE ARE AMENDING "; WDS
    : PRINT
    : PRINT
    : PRINT
240 OPEN "D", 1, WDS, 1
250 FIELD 1, 1 AS A$
260 FOR X = 1 TO LOF(1)
270 GET 1, X
280 IF HEX$(ASC(A$))="1F" THEN LSET A$=""
    : PUT 1,X
    : PRINT "1F:"; X
290 IF HEX$(ASC(A$))="1E" THEN LSET A$=""
    : PUT 1, X
    : PRINT "1E:"; X
300 NEXT
310 CLOSE
320 PRINT
    : PRINT
    : PRINT
330 PRINT "THE REPLACEMENT PROCESS IS COMPLETE"
340 SYSTEM "STARTUP"
350 END

```

Underscore with SCRIPSIT 2.0

J. G. Lowder
Communication Architects
P.O. Box 300
Lynnwood, WA 98036

One contribution and one request:

SCRIPSIT 2.0 FAST UNDERSCORE

I very much appreciate the new SCRIPSIT 2.0 software, but I found all the keystrokes required to get words underscored to be cumbersome. In 1.0 Scripsit it was much easier. However, thanks to the User Sequence Editing feature I came up with a solution which others might find helpful, also. Now when I want to underscore a single word or several words, it takes only about half the keystrokes.

Here is the User Sequence, which I set up on User key 7:

F1 ESC ESC ESC _ ESC ESC ESC _ F2 LT F1^Z

The procedure for using the sequence is:

1. PRESS ESC ESC 7
2. TYPE the words to be underscored
3. PRESS F2
4. PRESS →

This sequence is much easier than having to interrupt text development, both at the opening and at the ending of the string to be underscored. It's also less likely that you might end up forgetting to end the underscore and find yourself with a whole page of underscored material.

NOW THE REQUEST: Please publish a list of all the PATCHes for MODEL II. When I call your customer service line, they tell me it is the store's responsibility to advise owners of PATCHes and when I contact the store, they try to help, but they tell me that the owner registration is sent to Fort Worth and it is Fort Worth's responsibility. In any case, I'm sure there must be many PATCHes which are optional or lost between the cracks somewhere. I recently lost an entire disk full of information because I had not been notified of a series of PATCHes which had been put out two months ago. They still haven't appeared in Microcomputer News. HELP!!

Editor's comments:

First, with Scripsit 2.0 all that is needed to begin or end an underscore is (ESC)(_).

The procedure for underscoring a word or phrase becomes:

1. PRESS ESC _
2. TYPE the words to be underscored
3. PRESS ESC _

(Continued on Page 39)

Medical Office System

Here is some information which may be of interest to users of the Model II Medical Office System.

1. If you expand the transaction file past the limits specified in the manual and receive a "Disk Space Full" error message, you should return to a Backup copy and expand the file with a smaller number of records. If you fail to do this, you will receive an error 28 — attempt to read past end of file, during End of The Month Purge processing, and all data may be lost.

2. The Daily Input program is set up to detect an overflow during math operations (posting charges, payments and adjustments). If an overflow occurs, an error message will flash on the bottom of the screen. This error will occur if:

The total due on an invoice exceeds \$9999.99 or is less than \$-999.99.

The total due for the patient exceeds \$99999.99 or is less than \$-9999.99.

The month-to-date or year-to-date fields on the specified procedure exceed \$99999.99. This can occur when using the technique of entering previous balances with a single procedure called PREVIOUS BALANCE DUE, if the A/R exceeds \$99999.99. Once this occurs, the user can delete the procedure and reenter it to clear the month- and year-to-date fields.

3. The Aged A/R report will drop the most significant digit if the balance due for insurance or patient exceeds \$99999.99. The grand total will be correct up to \$999999.99.

4. If records have been added or changed by Daily Input, it is important that the user do a Daily Recap before running either the Aged A/R Report or EOM Purge.

5. The Insurance Print and Statement Print programs are written in BASIC. Their names are GENERAL/BAS and STATEMENT/BAS. On exceedingly large invoices, either an OS or BS error can occur. You may change the CLEAR and DIM statements as required to eliminate the OS or BS error. Don't forget to resave the program under the appropriate filename. Note—these errors have only been known to occur on invoices with more than 38 line items. Caution: Do not attempt to run either BASIC program directly, even for testing. You must enter these programs through the main menu to ensure that high memory is properly protected.

6. The system will work fine on the Hard Disk system under TRSDOS-HD 4.0. If the files are allowed to grow too large, the in-memory sort of the Aged A/R and Account Summary reports may not be able to sort all records. If this happens, simply narrow the scope of the report by selecting half the patients at a time, or a single doctor at a time.

7. YOU MUST NOT CHANGE THE FILE FORMATS USING PROFILE!!! You may create additional reports using the existing file.

8. To totally reset a medical system, you should use a backup copy of the Master Disk. If this is not available, the following steps must be used:

Pay off ALL open invoices with a WO (write off) payment code.

Run Daily Recap.

Run EOM Purge using a date of 12/99 (this should purge all invoices).

Change the system date to 01/01/1982 and run EOY Reset (EOY Reset won't run except in January.) Reply 1981 to the year that just ended question.

Change the system back to the real date.

If required, Run Daily Input and request the patient deletion option (to delete all patients).

9. You should be aware that the month-to-date and year-to-date figures on the procedure file are not adjusted when an existing charge is adjusted.

More Scripsit Ideas

C. M. Mendenhall
3905 Dogwood Drive
Greensboro, NC 27410

I read the "Scripsit Ideas" article for Model I and III in the April 1982 Microcomputer News with much interest and am sure that there are many ideas TRS-80 owners have which could be of much interest to other owners. I have become so obsessed with the SCRIPSIT program that I have retired my typewriter and do all my correspondence on the TRS-80. I have developed a couple of things in using SCRIPSIT that make it so much more useful that, even though they are pretty simple, enough of your users may not be aware of them that they may be worth printing.

I am using the SCRIPSIT with a Line Printer IV and, like many other printers, it provides several different type sizes and modes. To remind me to set the printer to the mode desired, I use the TRSDOS AUTO command to load it, but I have inserted a PAUSE in the auto routine which requires that I insert either "DO COMPTYPE," "DO PROPTYPE," or "DO NORMAL." It then shifts to BASIC and sends the proper command to the printer for the type required, and then shifts back to TRSDOS and loads SCRIPSIT.

As I use several sizes of stationary, I use several formats for the various types of stationary and type sizes. I have saved on my correspondence file disks all of the formats which I use and named them FORMAT1, FORMAT2, etc. I then set up a file which I call FORMINDX which contains a listing of the various formats with their names and a description of what they do. I can then check the FORMINDX and find the format which I wish to call for a particular document.

I also use another file on each correspondence file disk called CORRINDX which has a listing of the letters contained on that file. When I need to recall a document which has been filed previously, I need only to check the CORRINDX to find the name of the document. When a new document is saved on the disk, its name is added to the CORRINDX, to keep things current.

Model II Bugs, Errors and Fixes

Note to Users:

The following program changes and corrections are provided for your information. If you have an applications program which is working correctly, you should probably NOT make any changes to it. If you feel that the changes should be made, but you do not feel qualified to make the changes yourself, please contact your local Radio Shack Computer Center or Expanded Computer Department for assistance. If you do not have access to one of these stores, then you may want to call Computer Customer Services in Fort Worth for assistance.

Changes to BASIC programs

There are general procedures that need to be followed when any corrections are made.

1. Make a backup of the disk that contains the program to be corrected. Changes should be made on the backup copy.
2. Load the program to be changed by typing LOAD"filename" where filename is the name of the program to be modified.
3. Make the line changes indicated in the fix. For existing line numbers, edit or retype the line to match the one in the fix. New lines should be entered.
4. Save the corrected program (the one now in memory). Type SAVE"filename" (ENTER) where filename is the name of the program that has been modified.
5. Now make a backup of the corrected diskette.

PATCHES

PATCHes are entered from TRSDOS READY and are used to make corrections to files stored on the disk.

1. Before making a PATCH, back up the diskette that requires modification and make the PATCHes to the backup copy of the diskette.
2. Apply PATCHes according to the information given in your TRSDOS manual.

TRSDOS-HD (26-4150 Version 4.0)

When a fixed length file with a record length of 128, 64, 32, 16, 8, 4, 2, or 1 is closed, with the last record of the file being the last record of a physical sector, an Error 50 (Invalid Space Descriptor) will be generated. This error appears in BASIC as "IE" Error; if the user does "PRINT ERRS\$," BASIC will show that the error was TRSDOS Error 50. Apply the following patches to correct the above problem.

```
PATCH SYSTEM/SYS R=85 B=131 F=DD8610 C=C3AF1F
PATCH SYSTEM/SYS R=88 B=18 F=0202001E C=010E7420
PATCH SYSTEM/SYS R=88 B=22 F=000000000000
C=DD8610CA6C1E
PATCH SYSTEM/SYS R=88 B=28 F=000000000000
C=D26C1EC36B1E
PATCH SYSTEM/SYS R=88 B=34 F=00000000 C=0202001E
```

When 4800 BAUD operation is specified using either the SETCOM command or the RS232 Initialization Supervisor Call (55), the actual BAUD rate is being set to 9600

BAUD. The following patch is for the TRSDOS-HD 4.0 only and will correct the problem described above.

```
PATCH SYSTEM/SYS R=68 B=139 F=0D C=1A
```

Inventory Control (26-4502 Version 1.1)

INV allows user to change a stock number when quantity on hand is not equal to zero. The problem is corrected by making the following corrections to IMS/BAS:

DELETE LINES 238, 240, 242

```
Line: 234 GOSUB900: PUT1,I(I,0):GOSUB184:
IFKF<>0 THENKF=0:N1=I:N2=I(I,0):
I(I,0)=-I(I,0):VN=I(I,1):ND=ND+1:
GOSUB292:N1=N1-1:RETURN
Line: 236 NI=N1-1:GOSUB318:RETURN
```

ADD THESE LINES:

```
199 C1=CVS(F1$(11)):C2=CVS(F1$(12))
900 GET2,3:LSETG1$(0)=MKD$(CVD(G1$(0))-PR1*C1):
LSETG1$(1)=MKD$(CVD(G1$(1))-PR1*C2)
910 LSETG1$(2)=MKD$(CVD(G1$(2))-CS*C1):
LSETG1$(3)=MKD$(CVD(G1$(3))-CS*C2)
920 LSETG1$(6)=MKD$(CVD(G1$(6))-QT*CS):
LSETG1$(7)=MKD$(CVD(G1$(7))-QT*PR1):
PUT2,3:RETURN
```

Type SAVE"IMS/BAS" to save the changes in the program.

Accounts Payable (26-4505 Version 1.0)

When selecting invoices, an unexpected error code 61 in line 1620 can occur. The problem is corrected by making the following corrections to "APS/BAS":

```
Line: 1615 IFP(N,AM)<1 THENN=N+1:GOTO1610
Line: 3415 J=P(N,AM):IFJ<1 THEN3410 ELSEGET2,J:
GOSUB906:GOSUB920:GOTO1800
```

Type SAVE"APS/BAS" to save the changes in the program.

Accounts Payable (26-4505 Version 1.0/2.0)

When using the Print Checks option, the user cannot disable the printing of check numbers on the checks. The problem is corrected by changing "APCHECKS/BAS":

TO ELIMINATE CHECK NUMBER FROM PRINTING ON STUB:

```
Line: 2410 LPRINTLIS:LPRINT:NAS:LPRINTLIS:
LPRINTLIS:RETURN
```

TO ELIMINATE CHECK NUMBER FROM PRINTING ON CHECK:

```
Line: 2515 LPRINTLIS:LPRINTLIS:LPRINTLIS:
LPRINTLIS:LPRINTLIS
```

Type SAVE"APCHECKS/BAS" to save the changes in the program.

Scripsit 2.0 (26-4531)

If you attempt to create a document on an almost-full SCRIPSIT diskette, no check is made to be sure that there is enough free space to create both the document and its first page. This results in "garbage" being displayed as the first page. To fix the above problem, apply the following patches:

```
PATCH SCRIPSIT A=D686 F=0000000000000000
C=2A0A811128003E96
PATCH SCRIPSIT A=D68E F=0000000000000000
C=B7ED52D8116D4A
PATCH SCRIPSIT A=D695 F=000000000000 C=DDE5E1AFC9
PATCH SCRIPSIT/SYS R=15 B=87 F=116D4ADDE5E1
C=CD86D6C2A93D
```

These patches will prevent SCRIPSIT from creating a document on any diskette that has less than 40 sectors free. After applying the above patches, apply the following patch to change the date in the initialization screen (it should show up as 01/08/1982 before you apply these patches; if not, obtain and apply all preceding patches first):

```
PATCH STARTUP A=E40E F=B0B1AFB0B8 C=B0B2AFB2B3
```

When this patch has been made, the SCRIPSIT initialization screen will show the date 02/23/1982 at the bottom.

Statistical Analysis (26-4540 Version 1.0)

Rounding errors appear in the standard deviation output in the Chi Square Analysis program. The problem is corrected by making the following changes to "CHISQ/BAS":

```
Line: 280 FORI=1 TONR:FORJ=1 TONC:
      CS=CS*(ABS(O(I,J)-E(I,J))-CC)^2/E(I,J)
```

Type SAVE"CHISQ/BAS" to save the changes in the program.

Statistical Analysis (26-4540 Version 2.0)

In Multiple Regression, the variable names are not being placed in the requested position. The problem is corrected by changing "REGAN/BAS":

```
Line: 120 PRINT:LLZ=MEM/8-1000:DIHX@(LLZ),
      BS(300),VS(12),XS(12),VNS(12),NR$(12)
Line: 200 ONERRORGOTO0:NRZ=CVI(NR$):
      NVZ=CVI(NV$):FORJ=1 TO 12:
      VNS(J)=VS(J):NR$(J)=VS(J):NEXT J
```

ADD this New Line:

```
2140 FOR DD=1 TO NV:VNS(DD)=NR$(NVZ(DD)):NEXT DD
```

TRSDOS (26-4910 Version 1.2)

The following patches (for QUME printers and TRSDOS 1.2a only) will correct LPRINT CHR\$(138) statements so that they cause a space character, carriage return to be issued to the printer. This is compatible with LP III mode.

After the patches are successfully applied, you must re-boot to use the printer. The patches are as follows:

```
PATCH IODVRS/SYS A=0056 F=FFFFFFFFFFFFFFFFFFFF
      C=213F0F3E8AB8C00620C9
PATCH IODVRS/SYS A=0F47 F=213F0F C=CD5600
```

TRSDOS (26-4910 Version 2.0)

Pressing any key during a boot-up sequence initiated by the "RESET" command causes unpredictable results if done prior to the date & time prompts.

The following patches will correct the problem

```
PATCH SYSTEM/SYS R=6 B=133 F=3E01D3F9C30000
      C=3E03D3F3C3C326
PATCH SYSTEM/SYS R=6 B=196 F=E5E5E5E5E5E5E5E5
      C=F33A5500CBADF3FF
PATCH SYSTEM/SYS R=6 B=204 F=E5E5E5E5E5E5E5E5
      C=3E01D3F9C30000
```

Scripsit (From Page 36)

It is not necessary to press (ESC) until you see the function you wish to use. As long as you know what you



want to do, simply press (ESC) once, then the appropriate key for the function you desire. If you forget what to press for a particular function you can repeatedly press (ESC) to see the various functions which are available.

Model II PATCHes: Sorry, but it would be impractical for Microcomputer News to publish a single list containing all the patches and corrections which are available for Model II and Model II software (It is a rather long list!). Also, it really isn't necessary.

In your local area you should have one or more Radio Shack Computer Centers or Expanded Computer Departments. Either of these locations will have at least one individual who is knowledgeable about program fixes and corrections, and he/she has immediate access to a manual which contains all current and past corrections to programs.

This manual is updated constantly with the most current information possible. In fact, the Bugs, Errors, and Fixes which are published in the Microcomputer News appear in this bulletin FIRST, and they are there TWO or THREE MONTHS before you get them here (we don't put them into an issue until they appear in bulletin form, and then it takes about two months before you get the information at home).

Late Note For 26-4604

The COBOL RUNTIME module of 26-4604 Accounts Receivable is version 1.3a. The current version of RUN-COBOL is 1.3b. If you have purchased 26-4604, you should go to your nearest Radio Shack Computer Center or Radio Shack Store with a Computer Department, and request up-grade 700-2023 immediately. This is to your benefit, as version 1.3a is no longer supported and 1.3b allows for error trapping capability that is otherwise unavailable.

You Want to Calculate What?

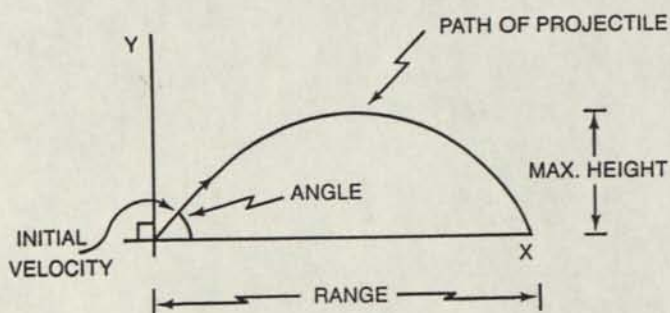
Projectile Trajectory

Larry Mermelstein
17 Bobolink Lane
Northport, NY 11768

I recently purchased a TRS-80 Pocket Computer. Being a high school senior, I felt this would be a wise investment for college. I have also found the PC great for high school work. For all those students that are having trouble with Physics, here is a program that computes the trajectory of a projectile ejected at an angle "B" with an initial velocity "A."

```

5: "A"
  : DEGREE
10: PAUSE"PROJECTILE TRAJECTORY"
15: INPUT "ENGLISH OR METRIC?";Z$
20: IF Z$="METRIC" LET L=4.9
  : GOTO 30
25: L=16
30: BEEP (1)
  : INPUT "INITIAL VELOCITY?";A
35: BEEP (1)
  : INPUT "ANGLE?";B
40: X=(COS B)*A
  : V=(SIN B)*A
45: C=V/L
  : D=X*(V/L)
  : H=V^2/(L*4)
50: PRINT "AIR TIME=";C
55: PRINT "RANGE=";D
65: PRINT "MAX. HEIGHT=";H
70: END
  
```



Above is a diagrammatic representation of what the program calculates. Note: The "Air Time" is the amount of time the projectile stays in the air. The velocity is to be entered in feet or meters per second. This will yield the time in seconds and the range and height in feet or meters.

Measurement Conversion

Frederick E. Bishop, R.A.
P.O. Box 648
Sharon, PA 16146

I am an architect and have a couple of programs of interest to those who work with dimensions.

Program 1 — Addition and Subtraction of Feet/Inches

The first one allows you to add and subtract feet and inches. Note: When subtracting enter the number of feet to be subtracted as a negative number. To complete a calculation when adding or subtracting press (ENTER) twice.

```

99: "A"
  : CLEAR
100: INPUT "FT ";A,"IN ";B,"NUM";C,"/";D
  : GOTO 100
150: PRINT USING "###";G;"FT ",F;" ";E;"/16 IN"
155: GOTO 99
160: END
510: IF D=2 LET C=C*8
  : GOTO 517
513: IF D=4 LET C=C*4
  : GOTO 517
516: IF D=8 LET C=C*2
517: IF A<0 GOTO 550
518: REM ADD
520: E=E+C
  : IF E>16 LET F=F+1
  : E=E-16
530: F=F+B
  : IF F>12 LET G=G+1
  : F=F-12
531: G=G+A
540: RETURN
545: REM SUBTRACT
550: E=E-C
  : IF E<0 LET F=F-1
  : E=E+16
555: F=F-B
  : IF F<0 LET G=G-1
  : F=F+12
560: G=G+A
570: RETURN
  
```

Program 2 — Shift Contents of Display to Memory

The second program is a very handy one for me. It allows storage of a displayed value for use as the variable "X." It can either be used for storage or, by pressing "SHIFT =" can be used for the initial calculation.

```

10: "="
  : AREAD X
  : PRINT X
  : END
  
```

Program 3 — Convert Decimal to Feet-Inches

The third program, which converts feet to feet and inches, is useful when working with surveyor's materials which are always in decimal.

```

20: "B"
   : CLEAR
22: INPUT "DECIMAL=";A
24: B=INT(A)

26: C=A-B
   : D=C*12
   : E=INT(D)
28: F=D-E
   : G=INT(F*16)
30: PRINT B;" FT ";E;" ";G;"/16 IN"
32: END

```

If 21.673 is entered as the decimal value in this program then the result would be 21' 8 $\frac{1}{16}$."

Note that in programs 1 and 3 the results are displayed in some fractional sixteenth inches. In order to convert back to halves, quarters, etc. would take another twenty or so steps, and was just not worth it to me.

Factorials

Thomas Cox
Drawer 2327
Greenville, SC 29602

Factorials are not a built-in function of the Pocket Computer, but they can be easily calculated as illustrated in Program 1.

Program 1. Calculation of Factorials

```

10: REM CALCULATE N FACTORIAL
20: INPUT "N?";N
30: IF N > 69 PAUSE "N MUST BE LESS THAN 70"
   : GOTO 20
40: Q=1
   : FOR W=1 TO N
   : Q=Q*W
   : NEXT W
50: PRINT N;"!=";Q
   : END

```

Unfortunately, Program 1 is very slow when a program must calculate a multitude of large factorials. Speed can be greatly enhanced if a look-up table approach is used. A disadvantage is that 70 memories are required to store the factorials of 0 through 69. An example of using a look-up table for factorials is given in Program 2.

Program 2. Look-up Table Factorials

```

10: REM LOOK-UP TABLE FACTORIALS
20: A(27)=1
   : FOR W=1 TO 69
   : A(27+W)=A(26+W)*W
   : NEXT W
30: INPUT "N?";N
40: IF N>69 PAUSE "N MUST BE < 70"
   : GOTO 30
50: M=27+N
   : PRINT N;"!=";A(M)
60: GOTO 30

```

Cost Comparison

Vernon T. Scally
P.O. Box 447
Edwards, CA 93523

With the economy in the state it is, more people are becoming cost conscious. Unit cost comparison is one

method to help save money; however, when shopping in a busy supermarket, trying to figure what the unit cost is for items not marked can get very confusing. The pocket computer can be carried to the supermarket with you and this program can help you by doing the comparing for you with little effort on your part.

Enter the cost of one item and its weight in ounces or grams, then enter the cost and weight of a second item and the computer will display the unit cost for both items at the same time.

```

10: "A"
   : CLEAR
20: INPUT "OUNCES(O) OR GRAMS(G)?";Z$
30: INPUT "A PRICE =.";A
40: INPUT "A UNIT =.";B
50: INPUT "B PRICE =.";C
60: INPUT "B UNIT =.";D
70: IF Z$="G" GOTO 110
80: G=INT B
   : G=16G
   : H=(B-INT B)*100
   : I=C+H
90: J=INT D
   : J=16J
   : K=(D-INT D)*100
   : L=J+K
100: GOTO 120
110: I=B
   : L=D
120: E=A/I
   : F=C/L
130: E=INT(100E+.5)/100
   : F=INT(100F+.5)/100
140: IF Z$="G" GOTO 170
150: PRINT "A= $";E;"/OZ B= $";F;"/OZ"
160: GOTO 10
170: PRINT "A= $";E;"/GM B= $";F;"/GM"
180: GOTO 10

```

Note 1 lb. 5 oz. will be entered as 1.05
12 oz. will be entered as .12
120 grams will be entered as 120

Resistor Values

Robert F. Edwards
Steed Rd. Box 581-A
Chenango Forks, NY 13746

To calculate resistor values first enter the resistor colors in the RESERVE mode as follows:

A:	BLA	'Black
S:	BRO	'Brown
D:	RED	'Red
F:	ORA	'Orange
G:	YEL	'Yellow
H:	GRE	'Green
J:	BLU	'Blue
K:	VIO	'Violet
L:	GRA	'Gray
=:	WHI	'White

Run the following program in the RUN mode and enter the resistor color by using (SHIFT) plus the color key.

```

4: BEEP 1
   : PAUSE "RESISTOR CALCULATOR"
10: INPUT "1ST BAND";A$
   : GOSUB 200
20: E=D*10
30: INPUT "2ND BAND";A$
   : GOSUB 200

```

```

40: F=D
   : T=E+F
50: INPUT "3RD BAND";A$
   : GOSUB 2000
60: IF D=0 LET G=1
70: IF D=1 LET G=10
80: IF D=2 LET G=100
90: IF D=3 LET G=1E3
100: IF D=4 LET G=1E4
110: IF D=5 LET G=1E5
120: IF D=6 LET G=1E6
130: IF D=7 LET G=1E7
140: IF D=8 LET G=1E8
150: IF D=9 LET G=1E9
160: H=T*G
170: IF H>1000 LET H=H/1000
   : GOTO 190
180: PRINT H;" OHMS "
   : GOTO 10
190: PRINT H;" K OHMS "
   : GOTO 10
200: IF A$="BLA" LET D=0
210: IF A$="BRO" LET D=1
220: IF A$="RED" LET D=2
230: IF A$="ORA" LET D=3
240: IF A$="YEL" LET D=4
250: IF A$="GRE" LET D=5
260: IF A$="BLU" LET D=6
270: IF A$="VIO" LET D=7
280: IF A$="GRA" LET D=8
290: IF A$="WHI" LET D=9
300: RETURN

```

Mileage Calculations

Alan J. Leiwant
301 Livingston Ave.
Livingston, NJ 07039

On a recent vacation trip, since I had some time on my hands and my pocket computer with me, I worked out a small program to calculate total miles traveled, mileage per leg, trip miles traveled and average trip mileage.

The way the program is written, it could provide for an unlimited number of legs within a trip or within the program usage. However, we've set 21 as the number of storage registers required to store each trip leg. The remaining five registers are used to control the loop and to maintain the mileage calculations.

The program is entered by selecting the DEFINE mode and pressing (SHIFT) (M). It then looks to see whether storage register A(1) is equal to 0. If it is, it clears all data registers and requests starting mileage. The next time through, the program automatically by-passes the clear position and requests current mileage and fuel used. It will then print out on the printer the miles traveled that leg, mileage, total trip miles, and average trip miles.

```

10: "M"
   : REM GAS MILEAGE ROUTINE
20: FOR V=1 TO 21
   : IF A(V)=0 GOTO 30
25: NEXT V
26: A(2)=A(V)
27: FOR V=3 TO 21
   : A(V)=0
28: NEXT V
29: V=3
30: IF V=1 GOTO 100
35: INPUT"ENTER CURRENT MILES ";A(V)
40: INPUT"ENTER FUEL USED ";Y
   : Z=Z+Y
50: W=(A(V)-A(V-1))/Y
   : X=A(V)-A(V-1)
   : PRINT USING"###.##";"MILES TRAVELED = ";X

```

```

60: PRINT USING"###.##";"MILEAGE =";W
70: W=(A(V)-A(1))/Z
   : X=A(V)-A(1)
   : PRINT USING"#####.##";"TOT. TRIP MILES =";X
80: PRINT USING"###.##";"AV. TRIP MILEAGE =";W
90: END
100: CLEAR
   : INPUT"ENTER STARTING MILES ";A
110: END

```

MPG on the PC

Bruce K. Taub
75 Captains Road
North Woodmere, NY 11581

After using the PC for sometime in my home and wood-working shop, I decided to take the PC onto the open road to test the MPG that my car gets.

I do a lot of traveling, and I prefer to know the actual MPG of my cars rather than the factory estimated MPG. By using this program I've found the conditions under which my car operated the best. I've also found the actual difference between city and country driving.

The program is very simple and easy to use. The PC takes the miles traveled, the amount of gas used, and finds the MPG. It also takes the past few MPG readings and averages them together for an overall average.

Program Description:

Press (SHIFT) (A) and the PC asks the TIME#:

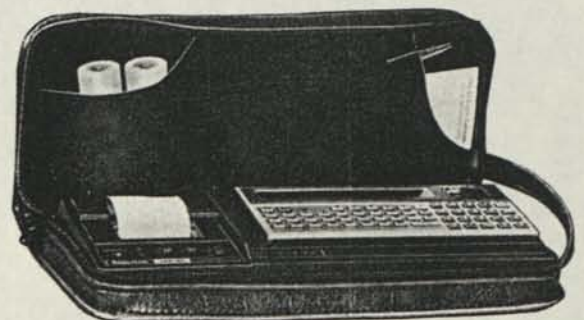
TIME#: — means how many times have you used the program. If it is your first time enter 1, not 0.

The first time, the computer sets the odometer. Be sure that you enter the odometer reading just after you filled the gas tank.

```

5: REM BY BRUCE K. TAUB APRIL 82
10: "A"
   : INPUT "TIME #:";T
20: INPUT "ODOMETER READING:";A(T)
25: IF T=1 THEN "MSG"
30: INPUT "AMT. LITERS:";X
   : Z=.26417X
60: LET V=(A(T)-A(T-1))/Z:PRINT USING "###.###";V;"
   MILES/GALLON"
70: LET S=(A(T)-A(T-1))/X
   : PRINT S;" MILES/LITER"
   : USING
80: LET R=T+5
90: A(R)=V
100: IF T>2 THEN 120
110: END
120: Y=0
130: FOR W=2 TO T
140: Y=Y+A(W+5)
150: NEXT W
160: U=Y/(T-1)
   : PRINT USING "###.###";U;" AVERAGE M/G"
170: END
180: "MSG"PAUSE"RESULTS NEXT TIME."
   : END

```



Color Editors

Text Editor 2.2

Brian Lieallen
9714 NW 20 Avenue
Vancouver, WA 98665

This is a text editing program for a Color Computer with Extended Color BASIC. In this program there are three basic modes: compose, edit, and list to printer.

In compose, you enter text for the first time. In the edit mode you can move the cursor up, down, left or right. You can also insert, delete or change characters, or you can extend your text.

Here is a list of the control keys in the different modes:

I. COMPOSE

- <Left Arrow> = Backspace and erase
- <Up Arrow> = Exit to menu
- <ENTER> = Carriage return when sent to printer

II. EDIT

- <Up Arrow> = Move cursor up one line
- <Down Arrow> = Move cursor down one line
- <Left Arrow> = Backspace
- <Space Bar> = Advance one space
- (1) = Delete one character
- (2) = Change one character
- (3) = Insert until <Up Arrow>
- (4) = Extend until <Up Arrow>
- <ENTER> = Return to Menu

```

1 '*** TEXT EDITOR 2.2 ***
2 '*** BY BRIAN LIEVALLEN ***
3 '*** NO RIGHTS RESERVED ***
10 CLS
   : CLEAR 200, 16311
   : GOTO 5000
20 INPUT "LINE WIDTH"; LW
30 DEFUSR0=16370
   : DEFUSR1=16342
   : DEFUSR2=16313
40 CLS
50 PRINT "'1' COMPOSE"
60 PRINT "'2' EDIT"
70 PRINT "'3' LIST TO PRINTER"
80 INPUT A
   : ON A GOTO 100, 210, 550
90 GOTO 40
95 ' *** COMPOSE ***
100 POKE 65479, 0
   : POKE 65481, 0
110 X=USR0(0)
120 A=1536
130 GOSUB 3000
140 AS=INKEY$
150 POKE A, 207
   : POKE A, 143
160 IF AS="" THEN 140
170 IF AS="--" THEN POKE A, 255
   : GOTO 40
180 IF AS=CHR$(8) THEN A=A-1
   : GOTO 140
190 GOSUB 2000

```

(Continued on Page 44)

SUPERTEXT

Matthew L. McGovern
RFD#1 Box 1330
Jay, Maine 04239

This letter was written using my program which I have dubbed SUPERTEXT.

I developed this program so that the Line Printer VII and Extended Color BASIC could be used to create paragraph and/or manuscript type papers. To the best of my knowledge, the program can perform each of the following: Title Positioning and underlining (7BS or 8BS, the 8BS requires TRS-80 PTFX16 for Extended Color BASIC computer), form new paragraphs, automatically indent paragraphs, and use the computer's upper/lower case mode (<SHIFT> and (0) key).

The program can only be appreciated by giving some examples of its use. A most notable example is the fact that I used it to write a term paper for school (I'm a 17 year old Junior at Jay High School) and it was gladly accepted by my teacher. And, while it does not produce a quality high enough to send to publishers, I find it quite useful in writing rough drafts of my material.

The program works by using several Line Printer commands. These commands are outlined in the booklet supplied with each LINE PRINTER VII, so I feel it is not that necessary to go in depth about their technical workings. However, CHR\$(26) was quite helpful in intercepting the computer's automatic message to the printer to advance a line, and CHR\$(18) and CHR\$(30) in switching from graphics to character printing modes (The graphic mode is used with 8BS underlining).

The program responds to very simple commands — (Y)=YES, (N)=NO. This is the major understanding necessary for immediate operation and execution of the program. Yet, a few aspects of it must be noted. First of all, an input must be entered to the prompt of Y=8BS/N=7BS or an ERROR message (one written into the program) will appear in the extreme upper left hand corner of the screen. If a lengthy paragraph is being entered, and you come to the end of the input line allowance (225 characters), simply enter the existing material and start typing where you left off on your next TEXT...? prompt. And, as in typing, remember to space twice after each sentence and at the end of each TEXT...? input to assure correct sentence spacing. Good luck with the program!

P.S. I have thoroughly enjoyed the new TRS-80 Screen Print Program for the Extended BASIC Computer. I hope more material of this sort is in the planning process.

```

10 ' *** SUPER TEXT ***
20 ' ***BY MATT MCGOVERN***

```

(Continued on Page 45)

Text Editor 2.2 (From Page 43)

```

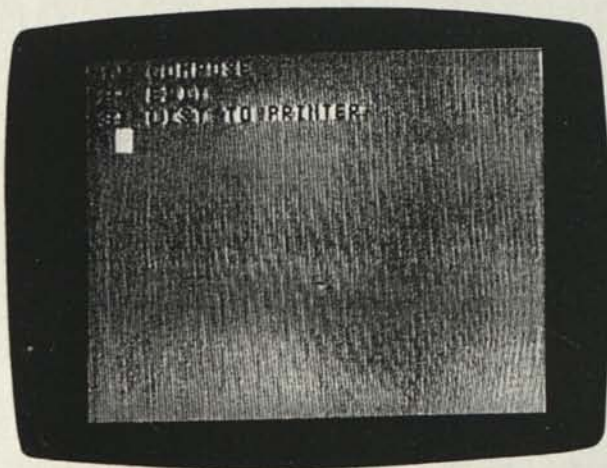
200 POKE A, ASC(AS)
   : A=A+1
   : GOTO 130
205 ' *** EDII ***
210 POKE 65479, 0
   : POKE 65481, 0
220 A=1536
230 P=PEEK(A)
   : POKE A, 175
240 AS=INKEY$
   : IF AS="" THEN 240
250 POKE A, P
260 IF AS=CHR$(32) THEN A=A+1
270 IF AS=CHR$(8) THEN A=A-1
   : IF A=1535 THEN A=1536
280 IF AS=CHR$(10) THEN A=A+32
290 IF AS="" THEN A=A-32
   : IF A<1536 THEN A=1536
300 GOSUB 3000
310 IF AS=CHR$(13) THEN 40
320 ON VAL(AS) GOTO 340, 380, 410, 490
330 GOTO 230
335 ' ***DELETE***
340 F1=(A+1)/256
   : F2=(F1-INT(F1))*256
350 POKE 16343, F1
   : POKE 16344, F2
360 X=USR1(0)
   : GOTO 230
375 ' *** CHANGE ***
380 AS=INKEY$
   : IF AS="" THEN 380
390 GOSUB 2000
400 POKE A, ASC(AS)
   : A=A+1
   : GOTO 230
405 ' *** INSERT ***
410 AS=INKEY$
   : IF AS="" THEN 410
420 GOSUB 2000
430 IF AS="" THEN 230
440 F1=(A/256)
   : F2=(F1-INT(F1))*256
450 POKE 16314, F1
   : POKE 16315, F2
460 X=USR2(0)
470 POKE A, ASC(AS)
   : A=A+1
480 GOTO 410
485 ' *** INSERT ***
490 AS=INKEY$
   : POKE A, 207
   : POKE A, 143
   : IF AS="" THEN 490
500 A1=ASC(AS)
510 IF A1=8 THEN A=A-1
   : GOTO 490
520 GOSUB 2000

```

```

530 IF AS="" THEN POKE A, 255
   : GOTO 230
540 POKE A, ASC(AS)
   : A=A+1
   : GOSUB 3000
   : GOTO 490
545 ' *** SEND TO PRINTER ***
550 FOR A=1536 TO 3071
560 A1=PEEK(A)
570 IF A1>0 AND A1<27 THEN A1=A1+96
580 IF A1=143 THEN A1=32
590 IF A1=255 THEN 640
600 IF A1=191 THEN 640
610 B$=B$+CHR$(A1)
620 IF LEN(B$)=LW THEN 640
630 NEXT A
640 PRINT #2, B$
   : B$=""
650 IF A1=255 THEN 40
660 IF A=3071 THEN 40
670 NEXT A
675 ' *** CHR CODE CONVERT ***
2000 A1=ASC(AS)
2010 IF A1>96 AND A1<127 THEN AS=CHR$(A1-96)
   : RETURN
2020 IF A1=32 THEN AS=CHR$(143)
   : RETURN
2030 IF A1=13 THEN AS=CHR$(191)
2040 RETURN
2045 ' *** SCREEN SELECT ***
3000 IF A>2559 THEN POKE 65479, 0
   : POKE 65480, 0
   : POKE 65483, 0
   : RETURN
3010 IF A>2047 THEN POKE 65483, 0
   : POKE 65478, 0
   : POKE 65480, 0
   : RETURN
3020 POKE 65479, 0
   : POKE 65481, 0
   : POKE 65482, 0
   : RETURN
3025 ' *** MACHINE LANGUAGE ***
5000 FOR A=16313 TO 16382
5010 READ B
   : POKE A, B
   : NEXT
5020 DATA 142, 6, 0, 166, 128, 230, 132, 167,
128, 30, 137, 129, 255, 39, 7, 140, 11, 255,
39, 2, 32, 239, 231, 132, 134, 255, 167,
132, 57
5030 DATA 142, 0, 0, 230, 132, 231, 130, 166,
129, 230, 132, 193, 255, 39, 10, 140, 11,
255, 39, 5, 231, 130, 126, 63, 221, 231,
130, 57
5040 DATA 198, 143, 142, 6, 0, 231, 128, 140, 11,
255, 38, 249, 57
5050 GOTO 20

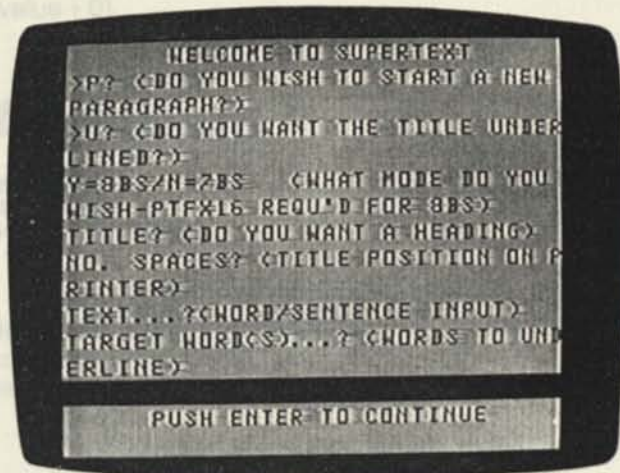
```



SUPERTEXT (From Page 43)

700 PRINT #2, CHR\$(30);
710 GOTO 300

```
30 ' *** JANUARY 4, 1981 ***
40 ' *** RFD#1 BOX 1330 ***
50 ' *** JAY, MAINE 04239 ***
60 ' *** WORD PROCESSOR ***
70 ' *** UPPER/LOWER CASE ***
80 ' *** CHANGE PARAGRAPH ***
90 ' *** UNDERLINE TITLES ***
100 ' *** 7BS/8BS CAPACITY ***
110 CLS
120 PRINT@ 416, STRING$(32, 128);
130 PRINT@ 6, "WELCOME TO SUPERTEXT"
140 PRINT ">P? (DO YOU WISH TO START A NEW
PARAGRAPH?)"
150 PRINT ">U? (DO YOU WANT THE TITLE
UNDERLINED?)"
160 PRINT "Y=8BS/N=7BS (WHAT MODE DO YOU
WISH-PTFX16 REQU'D FOR 8BS)"
170 PRINT "TITLE? (DO YOU WANT A HEADING)"
180 PRINT "NO. SPACES? (TITLE POSITION ON
PRINTER)"
190 PRINT "TEXT...?(WORD/SENTENCE INPUT)"
200 PRINT "TARGET WORD(S)...? (WORDS TO
UNDERLINE)"
210 PRINT@ 453, "PUSH ENTER TO CONTINUE"
220 QS=INKEY$
: IF QS=CHR$(13) THEN 230 ELSE 220
230 CLEAR 1000
240 CLS
250 INPUT "Y=8BS / N=7BS"; GS
260 IF GS="N" OR GS="Y" THEN 270 ELSE 610
270 CLS
: INPUT "TITLE"; FS
: IF FS="Y" THEN 280 ELSE 300
280 INPUT "NUMBER OF SPACES OVER"; A
290 L=L+A
300 CLS
310 IF FS="Y" THEN 330
320 LINE INPUT ">P?"; AS
: IF AS="Y" THEN A=3
: PRINT#-2, CHR$(10);
: L=0
: RS=""
: GOTO 270
330 LINE INPUT "TEXT...?"; SS
340 IF SS="" THEN 610
350 IF RS="STOP" THEN PS=""
: GOTO 380
360 IF FS="" OR FS="N" THEN 380
370 INPUT ">U"; PS
380 IF A>80 THEN A=ABS(A-80)
390 IF A>80 THEN A=ABS(A-80)
400 IF A>80 THEN A=ABS(A-80)
410 FOR G=0 TO A
420 PRINT #2, CHR$(32);
430 NEXT G
440 B=LEN(SS)
450 A=A+B
460 PRINT #2, SS; CHR$(26);
470 IF PS="Y" THEN 500
480 L=L+A
490 GOTO 320
500 LINE INPUT "TARGET WORD(S)...?"; Y$
510 FOR G=0 TO INSTR(1, SS, Y$)+(L-1)
520 PRINT #2, CHR$(32);
530 NEXT G
540 IF GS="Y" THEN 640
550 FOR T=0 TO (LEN(Y$))-1
560 PRINT #2, CHR$(95);
570 NEXT T
580 PRINT #2, CHR$(26);
590 RS="STOP"
600 GOTO 320
610 PRINT@ 0, "***ERROR**"
620 FOR P=1 TO 300
: NEXT P
630 GOTO 240
640 PRINT #2, CHR$(18);
650 FOR T=0 TO (LEN(Y$)*6)
660 PRINT #2, CHR$(192);
670 NEXT T
680 RS="STOP"
690 PRINT #2, " "
```



Color Scripsit and the Line Printer VIII

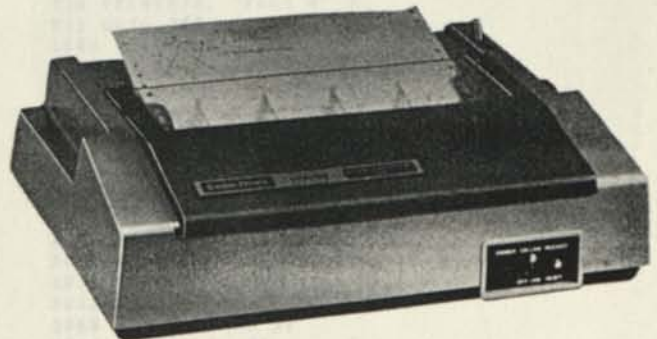
Garrett E. McGowan
3207 Evergreen Ave.
Hopewell, Virginia 23860

This correspondence was prepared on a Line Printer VIII, which will do 3 sizes of characters. It was also done by Color Scripsit on a 32K Color Computer. While the Color Scripsit was basically prepared to be used with a Line Printer VII, which I also have, it will do the job on a Line Printer VIII as well.

To do so the computer must be off, with the Program Pak out. Now follow these steps:

1. Turn on the printer
2. Turn on the computer (comes up in BASIC)
3. Initialize the printer to the appropriate type style 'PRINT#-2, CHR\$(27);CHR\$(n)'
4. Turn off the computer (leave the printer on)
5. Insert the Scripsit Program Pak
6. Turn the computer on
7. Choose option #6 (Change Standards)
8. Set Line Width and Margins as appropriate
9. Proceed as usual

Maybe this will assist those who have a Line Printer VIII. It certainly helps me.



Color Computer Bugs, Errors and Fixes

Notes to Users:

The following corrections and clarifications are provided for your information. If you feel that you do not fully understand this information, please contact your local Radio Shack Computer Center or Expanded Computer Department for assistance. If you do not have access to one of these stores, then you may want to call Computer Customer Services in Fort Worth for assistance.

Color Scripsit (26-3105)

Several problems have appeared in the Color Computer Scripsit Program Pak.™

First, at the main menu, do not press (7) for an option. Pressing (7) will cause the program to lock-up. To recover you will have to reset the machine, which will result in the loss of any material which is currently in Scripsit.

Second, if you are INSERTing material into a text line, do NOT press (SHIFT) (RIGHT ARROW) (a tab) if you are inserting over a carriage return ((ENTER) key). If you are in the regular text mode, tabbing when no tab position has been defined will simply move the cursor to the end of the current line. Tabbing over a carriage return while in the INSERT mode, with no tab defined, will result in the computer apparently locking up, because the program is inserting spaces to the end of memory. No data will be lost but it will be a long wait before you regain control of the program.

Third, in a Global Replace, if you are trying to change all occurrences of a character (example: change all occurrences of X in the string XXX to Y), the program will only change every other occurrence on the first pass. Simply make a second pass to change the remaining characters. Example: The text had the string 'XXXXXX' next to the string 'YYYYYY.' In this phrase, we want (for what ever reason) to change every occurrence of the letter X to a W. (BREAK)(7) gets us into the Global Replace mode. We indicate X and W as the character to be changed and the character to change to respectively. The computer displays the text, and we indicate that all occurrences should be changed. The resulting sentence will be: The text had the string 'WXWXWX' next to the string 'YYYYYY.' To change the remaining Xs to Ws, simply repeat the process.

Color Spectaculator (26-3104)

An error can occur when a number less than one is subtracted from a larger number which is also less than one. For example, if you tried to subtract .08 from .1. This type of subtraction will result in a large negative (and incorrect) result.

If position C1 contains 0.1, and C2 contains 0.08, the result of C1-C2 will be -9999.98, instead of the correct answer of 0.02. Changing the formula to read: C1-(C2 + 0) will result in the correct answer.

To apply this fix generally, all columns or rows that might contain a value less than one (1) should be entered

as: (Column Designation+0) or (Row Designation+0). Also, constants less than 1 should be entered as (value+0).

Asteroids

Richard Zepp
1972 Battle Row
Augusta, GA 30904

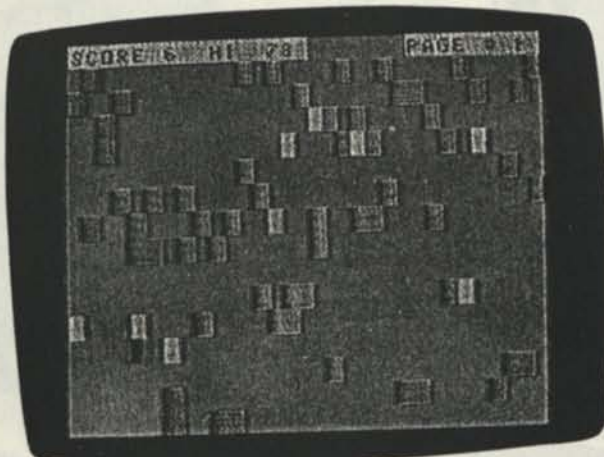
The program is for 4K Color Computers, high score to date 3349. I did not include the instructions because it is easy to figure out. HINT: use ← → for control, aim at the yellow ones.

```
1 CLS
   : PRINT "PRESS ANY KEY TO START"
2 A$=INKEY$
   : A=RND(0)
   : IF A$="" THEN 2
3 HS=0
94 C=1000
95 B=RND(64)
96 P=0
   : S=0
99 CLS(3)
100 FOR X=1 TO 63
110 Y=RND(31)
120 SET(X, Y, 8)
130 NEXT X
135 P=P+1
140 FOR X=26 TO 31
150 SET(B, X, 3)
160 NEXT X
170 SET(B, 31, 4)
171 FOR X=1 TO 10
172 A=RND(63)
   : W=RND(25)
   : SET(A, W, 2)
   : NEXT X
180 FOR X=1 TO 460
190 NEXT X
195 V=31
200 A$=INKEY$
205 IF A$="" THEN 210 ELSE SOUND 1, 1
210 SET(B, V, 3)
220 IF A$=CHR$(8) THEN B=B-1
   : IF B<1 THEN B=1
230 IF A$=CHR$(9) THEN B=B+1
   : IF B>63 THEN B=63
235 V=V-1
240 IF POINT(B, V)=8 THEN 1000
245 IF POINT(B, V)=2 THEN S=S+25
250 SOUND 200, 1
270 IF V=1 THEN 99
280 SET(B, V, 4)
290 IF B<>BC THEN S=S+1
300 BC=B
310 S=S+1
315 IF S>C THEN 2000
320 PRINT@0, "SCORE" S " HI " HS;
330 PRINT@23, "PAGE #" P;
335 GOTO 200
1000 PRINT "YOU LOSE"
1010 PRINT
   : PRINT
   : PRINT
   : PRINT
   : PRINT "TO PLAY AGAIN PRESS <ENTER>"
1020 INPUT Q
1030 IF S>HS THEN HS=S
1040 GOTO 94
2000 C=C+1000
2010 CLS(3)
2050 Z=30
2055 Y=33
2060 FOR X=10 TO 31
2080 SET(Y, X, 2)
```

```

2090 SET(Z, X, 2)
2095 Z=Z-1
      : Y=Y+1
2100 NEXT X
2105 Z=33
      : Y=31
2106 Y=30
2110 FOR X=10 TO 1 STEP-1
2120 Z=Z+1
      : Y=Y-1
2130 FOR A = Y TO Z
2135 SET(A, X, 8)
2140 NEXT A
2150 SET(31, X, 1)
2160 SET(32, X, 1)
2170 NEXT X
2171 RL=RND(10)+25
2172 V=31
2200 A$=INKEY$
2201 SET(RL, V, 3)
2210 U=RND(3)
2211 IF U=1 THEN RL=RL+1
2212 IF U=3 THEN RL=RL-1
2220 IF A$=CHR$(8) THEN RL=RL-1
2230 IF A$=CHR$(9) THEN RL=RL+1
2240 IF POINT(RL, V)=2 THEN 99
2245 V=V-1
2259 IF POINT(RL, V)=1 THEN 2500
2270 SET(RL, V, 4)
2275 FOR X=1 TO 25
2276 NEXT X
2280 P=P+1
2300 GOTO 2200
2500 S=S+250
2501 GOTO 99

```



Fireworks

Franklin G. Pigounias
2226 Texas Avenue
Savannah, GA 31404

As of yet I have not approached the advanced stage in programming, but here is a program for Extended Color BASIC. I receive your Microcomputer News on a regular basis and enjoy the helpful hints and information made available to me through it and hope that my programs may help others through your newsletter as it has helped me in the past.

P.S. Hope to get a 32K added soon.

```

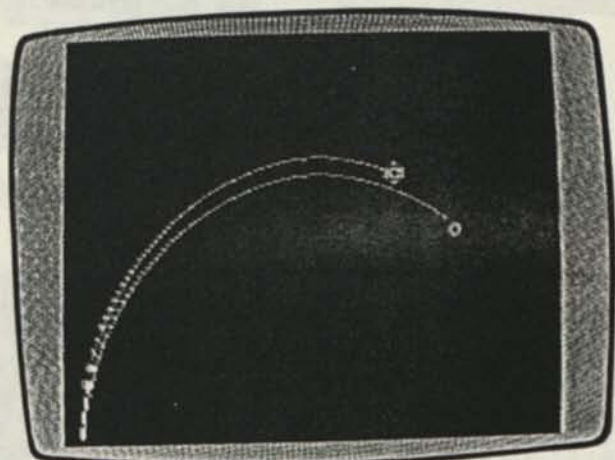
1 '-----
2 'FIREWORK'S DISPLAY
3 '-----
10 PHODE 4, 1

```

```

20 PCLS
30 SCREEN 1, 1
40 CIRCLE(128, 188), 110, , 1, .50, .75
50 CIRCLE(128, 79), 3
60 CIRCLE(128, 79), 5
70 CIRCLE(128, 79), 7
80 CIRCLE(128, 168), 110, 0, 1, .50, .75
90 CIRCLE(128, 79), 10
100 PCLS
110 LINE(256, 196)-(160, 100), PSET
120 CIRCLE(160, 100), 2
130 CIRCLE(160, 100), 3
140 LINE(256, 196)-(160, 100), PRESET
150 CIRCLE(160, 100), 5
160 CIRCLE(160, 100), 8
170 CIRCLE(160, 100), 10
180 CIRCLE(160, 100), 2, 0
190 CIRCLE(160, 100), 3, 0
200 CIRCLE(160, 100), 5, 0
210 CIRCLE(160, 100), 8, 0
220 CIRCLE(160, 100), 10, 0
230 CIRCLE(128, 188), 120, , 1, .50, .85
240 CIRCLE(128, 180), 120, , 1, .50, .80
250 CIRCLE(198, 94), 3
260 CIRCLE(168, 68), 3
270 CIRCLE(168, 68), 5
280 CIRCLE(128, 188), 120, 0, 1, .50, .85
290 CIRCLE(128, 180), 120, 0, 1, .50, .80
300 CIRCLE(168, 68), 8
310 CIRCLE(198, 94), 6
320 CIRCLE(198, 94), 10
330 PCLS
340 CIRCLE(52, 176), 3
350 CIRCLE(204, 176), 3
360 CIRCLE(128, 176), 3
370 CIRCLE(52, 176), 6
380 CIRCLE(204, 176), 6
390 CIRCLE(128, 176), 6
400 CIRCLE(52, 176), 3, 0
410 CIRCLE(204, 176), 3, 0
420 CIRCLE(128, 176), 3, 0
430 CIRCLE(52, 176), 8
440 CIRCLE(204, 176), 8
450 CIRCLE(128, 176), 8
460 CIRCLE(52, 176), 6, 0
470 CIRCLE(204, 176), 6, 0
480 CIRCLE(128, 176), 6, 0
490 PCLS
500 LINE(128, 192)-(128, 50), PSET
510 FOR D=1 TO 70
      : NEXT D
520 LINE(128, 192)-(128, 50), PRESET
530 CIRCLE(128, 50), 4
540 CIRCLE(128, 50), 4, 0
550 CIRCLE(128, 50), 6
560 CIRCLE(128, 50), 8
570 CIRCLE(128, 50), 6, 0
580 CIRCLE(128, 50), 10
590 CIRCLE(128, 50), 8, 0
600 CIRCLE(128, 50), 10, 0
610 GOTO 20

```



TRS-80[®]

Volume 4, Issue 8

September, 1982

Price \$1.50

• Microcomputer News

Information Published for TRS-80 Users.

Education Issue

Color Planetarium



Fort Worth Scene



TANDY NEWSLETTER

We are currently using the Tandy Newsletter on CompuServe to make program listings from TRS-80 Microcomputer News available to you without your having to enter them. We don't have the staff to create a tape or disk version of the News, but we can provide the listings on CompuServe.

What you will find in the Tandy Newsletter (GO TRS-1 in CompuServe) are compact program listings (no extra spaces, etc.) which can be downloaded and run in the appropriate machine, assuming you have appropriate hardware and software. You will not find instructions, explanations or supporting material for the programs. This information is available in the appropriate issue of the News. We know that this is not a solution that every TRS-80 user will be able to take advantage of, but we want to make it available for those who can use it.

Disk Model I and III CompuServe users can download the material using the new Disk-based VIDTEX which is available from CompuServe. Model II users can download using the TRSDOS TERMINAL utility. Radio Shack does not currently offer downloading capabilities for either the Color or Pocket Computers.

We will be including all program listings which are more than ten lines in length in this service. We know that some of the material (such as for PC-1) can not be downloaded into the machine the program was designed to run in. We are making all of the programs available, in the hope that the people who can download the material will be more willing to take a program designed for a TRS-80 other than their own and modify it for their machine. If you do modify one of these programs, send us a copy of the changes (through CompuServe or on electronic media) and we may be able to run them in a future issue of the News.

We have also begun a feature on the CompuServe Tandy Newsletter which allows us to get corrections to published issues of the News to you very quickly. While we make every effort to produce error free issues of TRS-80 Microcomputer News, there always seem to be a few errors that find their way into print. The News correction section will give you a quick way of finding corrections. If you happen to locate an error which is not listed on CompuServe, please let us know so we can pass the information along to others.

A NEW WRITER

With this issue we are happy to welcome a new writer to the News staff. Annette Zamberlin-Main has joined us and we have probably already overloaded her with work. Annette will be working with both the News and with the Tandy Newsletter on CompuServe. Annette is largely responsible for keeping the Tandy Product Information section on CompuServe up to date. With her help we are expanding that section to include all Radio Shack products, not just Computer items.

New Computer Clubs

THE BASICS

Frederick Town High School
c/o Gary Mayton
117 Columbus Rd.
Fredericktown, OH 43019

JOHNSON AMATEUR COMPUTER SOCIETY

34 South Fir Street
Medford, OR 97501
Computer Bulletin Board System phone number
503-535-6883

SILICON VALLEY COMPUTER CLUB

GTE Systems Bldg. 3
Central Expressway and Whisman Rd.
Mountain View, CA

SOUTH BAY TRS-80 USERS GROUP (SBUG)

DYSAN Bldg.
5401 Patrick Henry Dr.
Santa Clara, CA

TRS-80 USERS GROUP

Harriette Gross
c/o John C. Hart Memorial Library
1130 Main St.
Shrub Oak, NY 10588

See Page 38 for More Computer Clubs.



Sam Rosen Elementary School
Fort Worth Independent School District

Reproduction or use, without express written permission from Tandy Corporation of any portion of the Microcomputer News is prohibited. Permission is specifically granted to individuals to use or reproduce material for their personal, non-commercial use. Reprint permission for all material (other than William Barden's article), with notice of source, is also specifically granted to non-profit clubs, organizations, educational institutions, and newsletters.

TRS-80 Microcomputer News is published monthly by Radio Shack, a division of Tandy Corporation. A single six month subscription is available free to purchasers of new TRS-80 Microcomputer systems with addresses in the United States, Puerto Rico, Canada and APO or FPO addresses. Subscriptions to other addresses are not available.

The subscription rate for renewals and other interested persons with U.S., APO or FPO addresses is twelve dollars (\$12.00) per year, check or money order. Single copies of the Microcomputer News may be purchased from Radio Shack Computer Centers or Computer Departments for \$1.50 suggested retail each. The subscription rate for renewals and other interested persons with Canadian addresses is Fifteen dollars (\$15.00) per year, check or money order in U.S. funds. All correspondence related to subscriptions should be sent to: Microcomputer News, P.O. Box 2910, Fort Worth, Texas 76113-2910.

Retail Prices in this newsletter may vary at individual stores and dealers. The company cannot be liable for pictorial and typographical inaccuracies.

Back issues of Microcomputer News prior to January, 1981 are available through your local Radio Shack store as stock number 26-2115 (Suggested Retail Price \$4.95 for the set). Back issues of 1981 copies are not available.

The TRS-80 Newsletter welcomes the receipt of computer programs, or other material which you would like to make available to users of TRS-80 Microcomputer systems. In order for us to reprint your submission, you must specifically request that your material be considered for reprinting in the newsletter and provide no notice that you retain copyrights or other exclusive rights in the material. This assures that our readers may be permitted to recopy and use your material without creating any legal hassles.

Material for publication should be submitted on magnetic media (tape, disk, or CompuServe). If you submit material on tape or disk, and it is accepted for publication, we will send you two cassettes or diskettes for each one you sent us. Cassettes will come from our box of mixed blank cassettes. If you submit material on CompuServe, and we think we may use the material, we will extend your Microcomputer News subscription by six months for each article accepted. If you are submitting material over CompuServe, please include your name and address or your subscription number so we can find you. If the material is very short, send it to us in E-Mail. If you have more than a few lines, you need to place the material in the ACCESS area of CompuServe and then let us know it is there by leaving a message on E-Mail.

Material may be submitted by mail to P.O. Box 2910, Fort Worth, Texas 76113-2910, or through CompuServe. The Microcomputer News' CompuServe user ID number is 70007.535.

Notes to Program Users:

Programs published in the Microcomputer News are provided as is, for your information. While we make reasonable efforts to ensure that the programs we publish here work as specified, Radio Shack can not assume any liability for the accuracy either of the programs themselves, or of the results provided by the programs.

Further, while Microcomputer News is a product of Radio Shack, the programs and much of the information published here are not Radio Shack products, and as such can not be supported by our Computer Customer Service group. If you have questions about a program in the Microcomputer News, your first option is to write directly to the author of the program. When possible, we are now including author's addresses to facilitate communications. If the address is not published, or if you are not happy with the response you get, please write us here at Microcomputer News. We will try (given the limited size of our staff) to find an answer to your question and, in many cases, will publish the answer in an up-coming issue of Microcomputer News.

Trademark Credits

- CompuServe[™] CompuServe, Inc.
- Dow Jones[™] Dow Jones & Co., Inc.
- NEWS/RETRIEVAL Service[®] Dow Jones & Co., Inc.
- ReformatTer[™] Microtech Exports
- Time Manager[™] Image Producers
- VisiCalc[®] VisiCorp, Inc.
- AlphaKey[™] Tandy Corporation
- MicroPILOT[™] Tandy Corporation
- Program Pak[™] Tandy Corporation
- SCRIPSIT[™] Tandy Corporation
- TRSDOS[™] Tandy Corporation
- TRS-80[®] Tandy Corporation

Contents:

Color Computer

Program

Planetarium by J.W. Hurley 45

Computer Clubs

Computer Customer Service

Computers in Education 8

Data Bases

CompuServe

The Multiple Choice Puts You to the Test 13

Dow Jones

Window on Wall Street 14

Profile

Transferring Data From Records Into Shorter Records 43

VisiCalc

Personal Time Savers 17

Education

Figure It Out With Alice and Mac 12

micro Research Problems 15

News From the Education Division 30

Straight Talk About Radio Shack Network Systems 29

Feature Story

Radio Shack Network in the Classroom 5

Fort Worth Scene

General Interest

Agriculture Source Book 44

IRA Account by E. C. Hoover 7

New Journal Calls For Papers 44

Search and Rescue and the TRS-80 by J. Wellman 7

Model I/III

Bugs, Errors, and Fixes

Accounts Payable (26-1554) 25

Accounts Receivable (26-1555) 26

Cassette Portfolio (26-1506) 25

COBOL Compiler (26-2203) 29

Disk Payroll (26-1556) 27

Programs

Arithmetic Practice by P. Fried 24

Crush - A Mathematical Game by T. Myerson 22

Customary to Metric by T. Pruzan 19

Directory Algorithm by T. Kelanic 20

Physical Science Program by A. Freeman 19

Prime Generator #1 by G. Greer 23

Prime Generator #2 by R. Euliano 23

String Formatting and the Window Game by D. Fuger 21

Model II

Bugs, Errors, and Fixes

Accounts Payable (26-4505) 37

Accounts Receivable (26-4554) 38

BISYNC (26-4715) 38

General Ledger (26-4501) 36

Programs

Auto Field Routine by M. Lupul 33

Calculator by A. Banick, Jr 34

General Electric ASCII Sequential Files by R. Halloran 35

Read Screen by A. Pacheco 34

String Machine Language Programs by C. Hamkins 33

Peripherals

Modem II Questions and Answers 16

Pocket Computer

Start-Up Instructions 39

Programs

Homework by Brian Cutler 39

Use of the TRS-80 Pocket Computer in Celestial Navigation by D. Parson 40

View from the 7th Floor

by Jon Shirley 4

Prices shown in TRS-80 Microcomputer News are in U.S. Funds.

View From the Seventh Floor

by Jon Shirley
Vice President
Radio Shack Computer Merchandising

This month my column is devoted to the Color Computer. Those of you who do not own one may still enjoy reading this as there is a little fun for everyone here. I get several letters a week from Color Computer owners who complain that we do not understand that it is a very powerful machine, that we only make and advertise game software, that we do not give out any information on the product (where have I heard that before?), and that in general we totally neglect it. Rather than blunt my pen continuing to answer all these letters personally I would like to use my forum here to respond.

The first thing that I learned from these letters back in April is that the Color Computer is not a TRS-80 or a Color Computer but a CoCo. Up until then I thought a CoCo was the Cockatiel we had at home that escaped one Sunday to enjoy life in the wilds of Texas. All in all I must say that CoCo is a more flattering nickname than that other one we got a few years ago.

Look guys, we know the 6809 is the best 8 bit processor and that the CoCo is a very powerful computer. After all we did design the thing. In fact we worked very closely with our semi-conductor supplier to specify the SAM chip inside which connects the 6809 to the VDG and eliminates a whole mess of chips that would have cost more and made the board larger. The use of the 6809 and the SAM is what lets our graphics work so fast. In fact there are many bench mark programs that run as fast on a CoCo as on a 16 bit IBM P.C.

On the other hand we also designed the CoCo to be cheap, which is why I assume a lot of you bought it. Any computer involves a lot of design trade off's. If you design for low price you trade off features, yet we managed to include quite a few like on-board extended BASIC, RS-232C, etc. The primary market we saw, and still see, for the CoCo is in the home. And in addition to being cheap that means game software in ROM, easy to use, and as child proof as possible. And the more we sell because of this huge market, the more software and hardware we can bring out for the CoCo and the more we can lower the price as we did in July.

The major design trade off we made was in the use of the VDG (for those of you who are lost, the VDG is the Video Display Generator, an integrated circuit that generates everything you see on the screen) and it's 32 character by 16 line display and associated color set and graphics resolution. There are other ways to generate the screen, label them expensive. No upper and lower case is one of the missing items but there are some benefits. Unlike some of our competitors this is not an "exclusive" chip and all the information needed to do machine language programming is available. And frankly I think 32 by 16 beats 23 by 22 (or is it 22 by 23?)

The "you only advertise games" is one that surprises me. We have not only advertised the extended BASIC ver-


sion a lot, but we also advertised our disk drives and ran an ad in BYTE and others with quotes from a variety of magazines and experts who all were saying what a powerful product the CoCo really is. And as far as software in general is concerned, it is NOT all on ROM paks (example the three cassette adventure games), and it is NOT all games (examples Scripsit, Spectaculator, Learning Lab, etc.). I will be the first to admit the disk software is late, but it is in the works and Spectaculator and Scripsit (with software upper and lower case) will be out soon to be followed by many more.

The annual catalog will feature two new peripherals designed for the CoCo and there are two more in the works. Also in the works are not one but two books by Mr. Bill Barden, the super author of Machine Language Programming on the TRS-80, who will tell you lots about its insides.

Another source for information is a magazine for CoCo owners, the RAINBOW. Now almost one year old it has grown to some 60 pages in the latest issue, has several software ideas in each issue and ads from those outside folks who are writing some pretty neat software and making some rather clever, but at your own risk, hardware items. For information write to the RAINBOW, 5803 Timber Ridge Drive, Prospect, KY 40059.

As for us, we WILL continue to bring out more games in ROM (and some on cassette), and we WILL continue to advertise the CoCo as a home computer, BUT we will also bring out "serious" software and peripherals and books to support those of you who want to unlock some of the power within. We will not bring out a General Ledger; I think that's a little too much.

And for all of you who have sent me letters, thanks. I read them all, the CoCo product manager read them all, and all should have received an answer. You have given us many things to think about, and some of your requests will lead directly to additional products.

Until next month. 



Radio Shack Network In The Classroom

Craig Moore
Computer Coordinator
Harriet Tubman Middle School
2508 N.E. Everett
Portland, Oregon 97232

The Radio Shack Model I and Model III microcomputers have a cassette port through which programs enter and exit the computer. Normally a cassette tape recorder is connected to the cassette port. When you enter "CLOAD", a program loads into the computer from the cassette tape recorder. When you enter "CSAVE", the program in memory exits the computer and is recorded on the cassette tape. If, however, one has several computers and a network controller, the cassette ports can be used to connect the computers so that programs can be transferred from computer to computer, rather than computer to cassette tape recorder.

The middle school computer-use class that I taught last year was very dependent upon Network I Controller and sixteen Model I 16K microcomputers. In this article I would like to share what I learned about the use of a network in the classroom.

(Editor's Note: These remarks also apply to the Network 2 System, which has the capabilities of Network I, but is faster.)

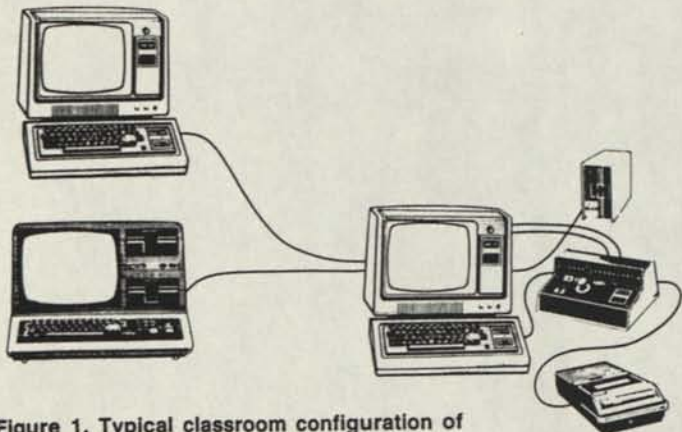


Figure 1. Typical classroom configuration of central disk system, Controller, and two student systems.

The Network I Controller is a controller (control box) and seventeen single strand cables (see fig. 1). One cable connects the controller to a computer with disk system that is designated as the "host". The remaining sixteen cables connect the controller to as many as sixteen microcomputers, referred to as "student stations". This equipment was placed on a rectangular arrangement of tables that occupied $\frac{3}{4}$ of the floor space in my classroom (see fig. 2).

One use of the network is to send a program currently in the host computer, through the controller to the student sta-

tions. When the host operator enters the command "CSAVE"; any student station with the command "CLOAD" will receive the program issued by the host computer.

This is a common procedure in my class for "handing back" student work. I keep student written programs from each class on a separate diskette. When students come to class they write the name of their program on the board. The host operator (often a student) inserts the class diskette and loads a student program into the host computer. The student enters "CLOAD" at the student station. A light comes on the controller indicating the student station is ready to receive the program. The host operator sees this, enters "CSAVE" and the program is transferred to the student station. If all sixteen controller lights were on, all sixteen student stations would receive the program. Students often will enter "CLOAD", before it is their turn, to view other student's programming efforts.

Because the program is transferred at the same 500 BAUD rate as in a cassette loading operation, the handing out of work can take as long as ten minutes (the time varies depending on the length of the programs that are sent). (Editor: Using a Network 2 Controller and TRS-80 Model III host and student stations, you can do this at 1500 BAUD—three times faster.)

The network can be used to send students instructional programs. For example, to illustrate the flow of activity in nested FOR-NEXT loops, I prepared a demonstration program. At the proper moment in my lecture, I had the students CLOAD the program from the host. While they ran the demonstration program, I asked them questions to focus their understanding of the nested FOR-NEXT loops.

Our schools scheduled one period each day to use the network for math computer assisted instruction. The programs we used (Radio Shack K-8 Math) had a wide range of skill levels and accommodated our ability grouped student's need for individualized lessons. It was possible to serve two groups, of sixteen students each, in the 45 minute class period.

Without a network controller, each student station would require a cassette tape recorder for saving programs. Although each TRS-80 Model I microcomputer system comes complete with a cassette recorder, I was relieved to not have the responsibility of accounting for sixteen recorders each period. With the network, a student would tell the host operator he/she wanted to save a program. The host operator

would turn a dial on the controller to the number of the student station and enter "CLOAD" on the host computer. When "CSAVE" was entered at the student station the program would transfer to the host. The host operator would then save the program on the class diskette or send it to the line printer for a hardcopy of the program.

Programs can also be sent from student station to student station with the same process as student station to host. Any computer with a "CLOAD" will pick up the transfer.

The handing in of programs at the end of a class session could take up to ten minutes.

The drawback to the use of a network controller is the slow transfer of programs from computer to computer. (Editor: This problem is reduced if you are using the faster Network 2.) If, in CAI class, the teacher has to load several long programs the time on task is shortened for the students whose program is loaded last. The teacher of a programming class may spend 20 minutes of class time transferring programs (although by saving programs in the same order as they were loaded, students should not have to wait more than 1/2 the total transfer time).

If a network were not used to transfer student programs to a central disk system for storage, but rather each computer had its own means of storage, i.e. cassette recorder, there would be no transfer time but the review of student work would be much more difficult. The teacher would have to pull programs off many different cassettes.

With the network, when student programs were to be graded, I would put the class diskette into the drive of the host

system and load a student program (loading off the diskette is much faster than loading off a cassette). Then I would make a listing on the printer and add notes about how the program ran. I was usually able to view and make listings of an entire class in a 45 minute period. Corrections and suggestions were usually added later when the listing was reviewed in detail.

We have begun the 1981-1982 school year with an additional seventeen-computer network system. Unlike our first system, the computers are TRS-80 Model III and the network controller is the improved Network 2. The advantages to this system include simplified operation and, most importantly, a 1500 Baud rate. This means that the program transfer time is significantly reduced.

Clustering many computers in one classroom has definite advantages. The high access to the computer that is gained by clustering allows programming students to learn a great deal by demonstrations and trial and error. The network facilitates the clustering of many computers in one room by making possible rapid, organized and inexpensive storage and retrieval of programs.

A computer network can provide computer assisted instruction to a whole class. Given a good collection of educational software, all teachers in a building should be able to schedule their class for CAI. The quicker transfer rate of the Network 2 suggests the possibility of a teacher loading several different programs to different groups within the CAI class, without a significant loss of time at task.

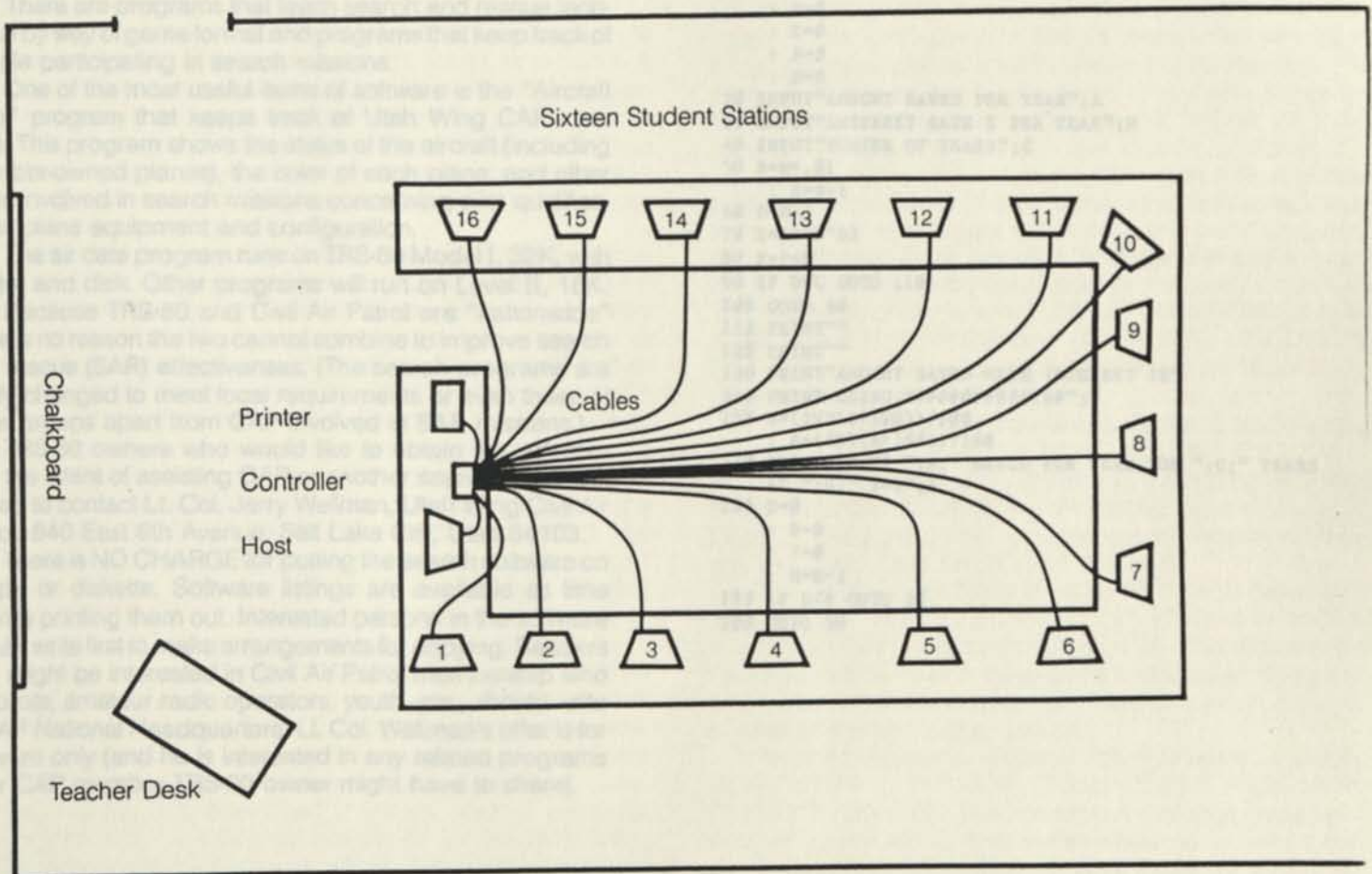


Figure 2. A Typical Classroom Setup

Search and Rescue and TRS-80

By Jerry Wellman,
840 E. 6th Ave.,
SLC, UT 84103

Can a TRS-80 have an important role in a search and rescue mission? Definitely, say several Utah TRS-80 users as they have demonstrated the use of the machine as members of Utah Civil Air Patrol.

(Civil Air Patrol is a nationwide, volunteer organization dedicated to search and rescue, aerospace education and a youth (cadet) program. National Headquarters of the CAP can be contacted at: Civil Air Patrol, Maxwell AFB, Alabama 36112.)

Some of the functional programs include compiling search effectiveness reports, listing a day's aircraft search effectiveness and — get this — offer a computer recommendation of best allocation of air search planes for the next day's effort.

Other programs keep track of up to 50 planes participating on a search (listing pilot information, time launched, time landed and total hours flown), and programs that convert latitude and longitude to the Air Force standard search grid (based on aircraft sectional maps) or converting the search grid back to latitude and longitude.

There are programs that teach search and rescue technique by way of game format and programs that keep track of people participating in search missions.

One of the most useful items of software is the "Aircraft Data" program that keeps track of Utah Wing CAP's aircraft. This program shows the status of the aircraft (including member-owned planes), the color of each plane, and other items involved in search missions concerning pilot qualifications, plane equipment and configuration.

The air data program runs on TRS-80 Model I, 32K, with printer and disk. Other programs will run on Level II, 16K.

Because TRS-80 and Civil Air Patrol are "nationwide" there is no reason the two cannot combine to improve search and rescue (SAR) effectiveness. (The search programs are easily changed to meet local requirements or even those of other groups apart from CAP involved in SAR missions.)

TRS-80 owners who would like to obtain the software with the intent of assisting CAP or another search group are invited to contact Lt. Col. Jerry Wellman, Utah Wing Civil Air Patrol, 840 East 6th Avenue, Salt Lake City, Utah 84103.

There is NO CHARGE for putting the search software on a tape or diskette. Software listings are available as time permits printing them out. Interested persons in the software should write first to make arrangements for copying. Readers who might be interested in Civil Air Patrol membership who are pilots, amateur radio operators, youth, etc., should write to CAP National Headquarters. Lt. Col. Wellman's offer is for software only (and he is interested in any related programs other CAP member-TRS-80 owner might have to share).

Figuring an I.R.A. Account

Emanuel C. Hoover
Rural Route 2 Box 172
Star City, IN 46985

This program figures I.R.A. type accounts at any amount of years and interest. The program is yearly compounded and puts results to a printer. It prints a set for the year span entered and the interest rate from the one entered down to six percent. After doing this, it asks for a new set of inputs. That is the reason for setting values to zero in line ten. Without that the values for all but the three inputs carry over into the beginning of the next set.

I am using this program with a 16K Color Computer and Line Printer VIII. We farm and have a small contracting business. We find this combination works fine for keeping farm records, figuring and printing grade survey figures, and pricing and billing materials. Our most recent purchase was the Spectaculator program, and we recommend it to anyone using a Color Computer for record keeping, inventory, or anything dealing with numbers.

```
10 CLS
   : A=0
   : B=0
   : C=0
   : D=0
   : E=0
   : F=0
   : H=0
20 INPUT"AMOUNT SAVED PER YEAR";A
30 INPUT"INTEREST RATE % PER YEAR";H
40 INPUT"NUMBER OF YEARS";C
50 B=H*.01
   : B=B+1
60 D=D+1
70 E=A*(B^D)
80 F=F+E
90 IF D=C GOTO 110
100 GOTO 60
110 PRINT""
120 PRINT""
130 PRINT"AMOUNT SAVED WITH INTEREST IS"
140 PRINT USING "#####.##";F
150 F=(INT(F*100))/100
   : A=(INT(A*100))/100
170 PRINT#-2,"$ ";A;" SAVED PER YEAR FOR ";C;" YEARS
   AT ";H;" %=$";F
180 D=0
   : B=0
   : F=0
   : H=H-1
190 IF H<6 GOTO 10
200 GOTO 50
```

Computers in Education

The Computer Age is here! Teachers and parents are outnumbered by children who use computers, learn via computers, and accept computers as a part of their world. Computers have become a part of our society and everyday life patterns. Teachers and parents who know nothing about computers often feel PANIC as their children advance into this new age of computer literacy. We are confronted with a computer generation which is forcing traditional education down new paths of learning. This article is intended to reassure inexperienced adults that they can cope and even excel in computer literacy along with the younger generation.

Teachers and parents don't need a lot of knowledge to keep up with this new age. The key philosophy to be remembered is not to be afraid of these electronic devices. Teachers and parents are generally apprehensive about computers, but computers don't break easily. Change causes us to think, to adjust, and to grow. It is time for educators and parents to become aggressive and use this computer age to their benefit. Some of the following suggestions will help you become more familiar with computers and allow you to incorporate them into your life.

Building a good software library will allow a computer to become a good learning center in the school or at home. A software library may be developed in many ways: buying pre-existing software which is ready to run on your computer, keying in programs from numerous books that you will find for sale in computer stores (some computer literacy helps in debugging and correcting book programs), or writing your own programs for specific uses.

Many of the Radio Shack education programs are excellent in school settings and very helpful for home reinforcement in learning necessary skills. Before attempting to make use of these or other pre-existing programs, a teacher or parent should review the manuals which come with their computer. By the time you have familiarized yourself with your machine and how to hook it up with all of the peripherals, you are ready to learn the fundamentals of BASIC and what capabilities your computer has. TRS-80 Model III's and Color Computers come equipped with an Owner's Manual and a book called GETTING STARTED WITH TRS-80 BASIC. You will talk to your computer by typing on the keyboard and it will talk to you on the video screen. If the computer doesn't understand you, it will say so. You really can't break or damage your computer by typing on the keyboard.

After completing your GETTING STARTED WITH TRS-80 BASIC, you will feel more confident in working with programs to be typed in and in using pre-existing programs.

If you choose to try pre-existing programs, you will find a wide variety of programs with many different skills. Radio Shack has developed a special kind of computer program which is designed specifically for teaching called CAI (Computer Assisted Instruction). This method is based on the use of the computer as a medium of instruction. Our

CAI Courseware development has added an additional instructional strategy. Our courseware products are developed and field tested, taking into consideration the feedback from teachers and curriculum developers. The CAI techniques which have proven most effective for supplementing the regular classroom instruction have been centered on individualized sessions where the computer generates and presents exercises for certain subject areas.

Many studies are being conducted on the impact of computers on education. Not only are they tools for learning, but they are a means for helping prepare students to live in a world increasingly dependent on computers. Radio Shack is meeting these education needs with their COMPUTER EDUCATION SERIES (CES). The Series is designed for use in the classroom at secondary and college levels. A teacher's manual, 25 workbooks, and a set of transparencies are included. PART 1: INTRODUCTION TO BASIC (26-2150) is designed as a first introduction to computer programming. PART 2: BASIC PROGRAMMING (26-2152) continues the concepts introduced in PART 1. PART 3: ADVANCED BASIC (26-2154) deals with more advanced programming concepts. These instructional materials require little programming or computer knowledge on the part of the instructor. All programming exercises are designed to be entered and run by students with hands-on access to the TRS-80. All three parts are designed for use with any Model I or Model III.

COMPUTER DISCOVERY is a new addition to our Computer Education Series. It introduces students to computers, their evolution and impact on society, and basic programming concepts. Each package includes interactive exercises, 25 student workbooks, and an instructor's manual. The two levels available are COMPUTER DISCOVERY FOR JUNIOR HIGH (26-2630) and COMPUTER DISCOVERY FOR SENIOR HIGH (26-2632). Additional workbooks are available separately.

Other pre-packaged courses designed to teach Basic Instruction include Model III BASIC COURSE (26-2010)—eight self-paced lessons on diskette to teach programming Model III BASIC COURSE—TAPE (26-2015), and Model II DISK COURSE (26-2014)—four self-paced lessons on diskette that cover DISK BASIC and TRSDOS.

The TRS-80 MICROCOMPUTER INFORMATION HANDBOOK FOR EDUCATORS (26-2757) is a booklet which describes microcomputers in detail. What they are and how they can be used in classrooms is discussed along with costs, courseware, and other considerations which could be of special concern to the educator.

Another Resource Material for educators is BASIC COMPUTER LITERACY: COMPUTERS PAST AND PRESENT (26-2755). This classroom package includes a teacher's guide and six spirit masters illustrating the history of computers, what a computer is, what it can do, and how it works. This is an introductory computer education package.

for grades four through eight. It explores the history of computers from the abacus and the first mechanical calculator to the programmable computers of today. A sample computer program is shown and analyzed. A colorful wall chart is also included. Additional activities and answers are printed on the non-duplicating side of the masters. This package was designed for effective use whether or not the class has access to a computer. It is not necessary for the teacher to have any previous knowledge of computers.

The PROPOSAL WRITING GUIDE (26-2754) is a step-by-step guide for educators seeking funding from an external source (such as the Federal government or a private foundation) for computer-related education projects. The guide leads you through procedures necessary to the writing of a successful proposal; detailing the most important elements.

MY TRS-80 LIKES ME (26-2751) is a new 32-page teachers' guide to helping elementary students understand and enjoy the BASIC computer language. It includes lots of programs and activity ideas.

NUMBER PATTERNS (26-2752) is a 36-page booklet showing how to introduce learners to sequences and series in an entertaining way using computers.

The TRS-80 EDUCATIONAL SOFTWARE SOURCEBOOK, VOLUME I (26-2756) has been introduced as a special edition of the TRS-80 SOFTWARE SOURCEBOOK for educational courseware. Vendor-furnished listings include a description of the program and important characteristics. It is indexed by subject, user level, and instructional technique.

A further aid for administrative personnel, available in the future, is a REKORD Planner which is a record-keeping system that can be personalized to fit your own needs. It will file and store student data, print out student and school records, and generate reports. It will be available in three versions: Administrator's version (26-2725), Counselor's version (26-2726), and Special Programs version (26-2727). It will be available for use on the Model III.

Our Educational Software includes a variety of Math programs ranging from K-8 MATH for the elementary grades through ESSENTIAL MATH for grades 7 through 12. K-8 MATH has a Student Management version which assists the teacher by keeping a complete record of all lessons and tests taken by each student. This feature allows the teacher to review and print out student reports. It automatically keeps track of lesson promotions and demotions sending each student to the correct lesson at the beginning of a new student session. The K-8 MATH, VOLUME ONE (26-1715) is for use with TRS-80 Model I or Model III. It is furnished on cassette and diskette and requires a minimum system of 16K tape or 32K 1-drive. The K-8 MATH WITH STUDENT MANAGEMENT PROGRAM, VOLUME ONE (26-1725) is for use with Model I or Model III. It is furnished on diskette only and requires a minimum system of 32K 1-drive with a line printer being optional. Unless otherwise indicated, all of the education programs are for use with TRS-80 Model I (with Level II BASIC) or Model III (with Model III BASIC).

The K-8 MATH WORKSHEET GENERATOR program provides an added feature in allowing a teacher to design and print worksheets and answer sheets for the addition, subtraction, multiplication and division programs of the K-8 MATH program. A teacher can design headings for the worksheet, choose the lesson numbers and the number of problems from each lesson, print identical or different worksheets

on the same level of difficulty and then save the worksheet design on diskette for future use. K-8 MATH WORKSHEET GENERATOR (26-2162) is for use with Model I or Model III. It is furnished on diskette and requires a minimum system of 32K 1-drive with a line printer.

Included with each of these programs is the Radio Shack K-8 MATH CROSS-REFERENCE (26-2750) which coordinates individual K-8 MATH lessons with specific chapters and pages in six widely used math basals.

The ESSENTIAL MATH PROGRAM, for grades 7 through 12, reinforces math concepts which have been introduced by the teacher. VOLUME ONE provides skill building exercises in addition, subtraction, multiplication, division, and number concepts. VOLUME TWO covers fractions, decimals, percents, and pre-algebra concepts. This program also includes a placement mode, as well as an option that allows automatic promotion and demotion of the student through the lessons. Constant feedback is provided to the student with reinforcement messages keyed for correct and incorrect responses. At the end of each lesson, a report giving the student's score appears on the screen. Scores can then be recorded on copies of a sample student record sheet included in the teacher's manual. This will provide a complete record of each student's progress. ESSENTIAL MATH PROGRAM, VOLUME ONE (26-1716) and ESSENTIAL MATH PROGRAM, VOLUME TWO (26-1719) are for use on Model I or Model III. It is furnished on cassette and diskette and requires a minimum system of 16K tape or 32K 1-drive.

The EUCLID GEOMETRY TUTOR program is designed to reinforce basic geometry concepts introduced by the teacher. It allows students to develop proofs using nine basic postulates of Euclidean Geometry. A teacher's manual includes an exercise section with a wide variety of problems. A quick-reference card is also included for use with the program. The EUCLID GEOMETRY TUTOR (26-1724) is for use with Model I or Model III. It is furnished on cassette and diskette requiring a minimum system of 16K tape or 32K 1-drive.

A series of Math and Physics Programs for the Secondary Grades are available for simulation and problem solving. All of these programs are for use with the Model I or Model III. They are furnished on cassette and diskette and require a minimum system of 16K tape or 32K 1-drive. The following programs are included:

ADVANCED GRAPHICS (26-1714)

VECTOR ADDITION (26-1720)

INTERPRETING GRAPHS IN PHYSICS (26-1721)

GRAPHICAL ANALYSIS OF EXPERIMENTAL DATA (26-1722)

INVESTIGATIONS IN INTEGRAL CALCULUS (26-2600)

PLANE ANALYTIC GEOMETRY (26-2602)

NUMERIC DATA ENTRY PRACTICE is a 25-lesson practice course designed to help students develop speed and accuracy in the 10-key entry of numeric data. Students use the TRS-80 standard numeric keypad to "key in" numbers that appear on the video screen. Students work through exercises of progressive difficulty. The computer records the student's performance on diskette for review by the teacher at a later date. Student reports may also be printed out using a line printer. NUMERIC DATA ENTRY PRACTICE (26-2601) is for use on the TRS-80 Model I or Model III. It is furnished on diskette and requires a minimum system of 32K 1-drive with a

line printer being optional.

More and more teachers are becoming aware of the instructional potential of microcomputers. An area of increasing interest to many teachers is that of reading instruction using the computer.

One of our first programs in that area is AlphaKey™, which is an alphabet and keyboard familiarization for children four to six years old. Parts 1 and 2 can teach children to discriminate capital letters, to associate capital and lower case letters, and to recognize the order of the alphabet. Part 3 gives the child practice in typing the letters of the alphabet in order with no hints. Immediate reinforcement is given for every response. A Happy Face appears for correct answers. If an incorrect answer is typed, the Happy Face will disappear until the correct answer is given. A report is given at the end of each part showing letters identified correctly on each try along with average response time. AlphaKey™ (26-1718) is for use with Model I or Model III. It is furnished on cassette and diskette requiring a minimum system of 16K tape or 32K 1-drive.

The HIGH MOTIVATION READING SERIES is designed to interest and challenge students at reading levels 4-6. Each package in the series contains four copies of an illustrated-format reader, and a read-along audio cassette tape. Educational activities related to the story in the reader are furnished on diskette for students to work on at the computer. Additional student activities are included in the teacher's manual. The HMRS STUDENT RECORDS SYSTEM (26-2521) is available separately and can be used with any of the Model III HIGH MOTIVATION READING SERIES programs to maintain a complete record of student activity scores. Titles in the series include:

CHARLES LINDBERGH/AMELIA EARHART (26-2513)
THE HOUND OF THE BASKERVILLES (26-2514)
DRACULA (26-2515)
MOBY DICK (26-2516)
THE BEATLES (26-2517)
20,000 LEAGUES UNDER THE SEA (26-2518)

The HMRS™ is for use with the TRS-80 Model I with lower-case modification and the Model III. It is on diskette and the minimum system requires the use of the TRS-80 MicroPilot™ program in conjunction with all seven titles. The HMRS STUDENT RECORDS SYSTEM does require a 32K 2-drive system with a line printer being optional. All of the other HMRS programs require 32K 1-drive systems. All of the catalog numbers listed above for the HMRS are for the Model III versions. Please check with your local Radio Shack dealer for availability of these programs for the Model I.

Radio Shack also has four READING IS FUN packages available for the Color Computer. They are designed for grades 4-6. Each package includes an illustrated reader, a read-along audio cassette tape, and a computer tape with spelling and vocabulary exercises. DRACULA (26-2550), THE HOUND OF THE BASKERVILLES (26-2551), MOBY DICK (26-2552), and 20,000 LEAGUES UNDER THE SEA (26-2553) require a 4K Color Computer and a CTR-80A cassette recorder.

The Radio Shack COMPUTER ASSISTED READING DEVELOPMENT program is adapted from the Philadelphia Reading Program and was designed to help develop the reading skills of students at the upper elementary level. C.A.R.D. 1: SENTENCES provides various exercises

and tutorials in recognizing sentences and sentence parts. Other modules will include PARAGRAPHS, DIRECTIONS and COMPREHENSION. These programs are for use with TRS-80 AUTHOR I or the TRS-80 AUTHOR I LESSON PRESENTATION PACKAGE. The C.A.R.D. modules include both the lesson program and a student record keeping feature. C.A.R.D. 1: SENTENCES (26-2603) is for use with the TRS-80 Model III and is furnished on diskette. It requires a minimum system of 32K 1-drive and also requires TRS-80 AUTHOR I or the TRS-80 AUTHOR I LESSON PRESENTATION PACKAGE.

Another group of programs designed for Educational use include the authoring systems. These programs allow teachers and users to create their own computer assisted instruction material.

The QUICK QUIZ program allows teachers to create, store, and give multiple-choice tests using a TRS-80 computer. It handles up to 40 questions with 4 answer choices per question. Up to 50 students can be tested and their scores stored on a disk file for later review. QUICK QUIZ (26-1728) requires a 32K Model I or Model III disk system.

TRS-80 MicroPilot allows the user to create or adapt courseware from curriculum suitable for computer assisted instruction to the TRS-80. It is a command-oriented authoring language and is based on the PILOT computer language. It has extended capabilities for graphics generation and file handling. A basic knowledge of computer programming is not required, but would be helpful. The MicroPilot package includes a sample lesson on the program diskette, a user's reference card, a screen reference card to aid in producing graphics, and a teacher's manual with complete instructions. MicroPilot (26-2205) is for use with the TRS-80 Model I and MicroPilot (26-2718) is for use with the TRS-80 Model III. It is furnished on diskette and requires a minimum system of 32K 1-drive.

TRS-80 AUTHOR I is a lesson development system designed to allow creation of courseware. Knowledge of computer programming is not necessary. A step-by-step instruction manual guides the user through all phases of lesson creation. The system allows text and graphics presentation, extensive branching, control over the number of tries allowed the student on a question, student access to glossary words provided by the author, hints to help students, and includes optional student record keeping. TRS-80 AUTHOR I (26-1727) is for use with the TRS-80 Model I or Model III and is furnished on diskette. It requires a minimum system of 32K 1-drive.

The TRS-80 AUTHOR I LESSON PRESENTATION PACKAGE is the portion of AUTHOR I which presents the lessons to the student using the TRS-80 computer. This module is available separately to allow teachers who do not want to develop their own lessons to use AUTHOR I lessons created and sold by others. The TEACH lesson presentation program and the STUDENT record keeping program are provided with this module. The TRS-80 AUTHOR I LESSON PRESENTATION PACKAGE (26-2707) is for use on Model I and Model III and is furnished on diskette. It requires a minimum system of 32K 1-drive.

Recent additions to Radio Shack's line of Educational Software includes HISTORY OF TECHNOLOGY for the Color Computer. PIONEERS IN TECHNOLOGY (26-2624) includes The Age of Flight, Space Exploration, The Electric Car

and History of Computers. INVENTIONS THAT CHANGED OUR LIVES (26-2625) includes Edison's Electric Inventions, Bell and the Telephone, The Story of Railroads, and The Age of Television. These packages describe technological achievements through graphics, text, and recorded speech.

A 16K Color Computer and a CTR-80A cassette recorder are required.

VOCABULARY TUTOR I (26-2568) and VOCABULARY TUTOR 2 (26-2569) are future programs which provide a fun way for your child to learn new words. They use the new Color Tutor system which combines pictures and text on the screen with spoken information and feedback.

An additional program for the Color Computer is TRS-80 Chemistry Lab, Volume I. The program includes a set of chemistry laboratory experiment simulations which allow students to control and see the results of various experiments many times without the expense or danger of using chemicals and labware. Titles in Volume 1 include Kinetic Theory, Charles' Law, Boyle's Law, Titration, Conductivity, and Solubility. The package includes an instruction manual and student experiment books. Volume 2 of Chemistry and two volumes of physics simulation are planned. TRS-80 Chemistry Lab, Volume I (26-2609) requires a 16K Color Computer with Extended BASIC and a CTR-80A cassette recorder.

The Radio Shack Education Division has also made a commitment to the hardware portion of student instruction. The TRS-80 NETWORK 2 CONTROLLER and the NETWORK 3 CONTROLLER allow an educator to design a network of up to 16 non-disk student stations around a single disk "host" system. With the NETWORK 2 CONTROLLER, a teacher can download programs into all 16 student stations at once, or into any combination of the student stations. Different programs can be loaded into different stations. Student programs can also be uploaded to the host. The NETWORK 2 (26-1211) is for use with a host computer that can be either a Model I or III (with at least one disk drive recommended) or a Color Computer (with a cassette recorder or disk drives). From one to sixteen student stations can be used. The TRS-80 Systems used as student stations do not require disk drives or cassette recorders. Model I and Model III computers can be mixed in a NETWORK 2 configuration. If Color Computers are used with a NETWORK 2, all computers in that NETWORK (including the host) must be Color Computers.

The NETWORK 3 (26-1212) is designed to allow nondisk student stations to access the disk drives and printer of the teacher's host system. This provides the student stations with most of the same disk features that make a disk-equipped computer so powerful.

Some of the features of the NETWORK 3 CONTROLLER are:

- * All 16 student stations can load and save programs using the disk at the host system, as well as use a line printer attached to the host system - without interrupting other students or the teacher.

- * Student stations can store information on the host disk, allowing class records to be kept in one location.

- * Each student station can be a part of the Network, or operate independently without disturbing the network cable connections.

- * Teachers can use the included MENU program to

create a "menu" (or option list) of programs for students, allowing them to choose from a selection of programs on the disk at the host computer.

- * NETWORK 3 makes it possible for computer science students to learn and use many of the Disk BASIC functions on non-disk TRS-80s.

The NETWORK 3 is designed to operate best in a student-centered classroom, a lab, or a resource room situation. It requires no teacher intervention under normal operating conditions. These features complement the NETWORK 2, which operates best in a teacher-oriented environment and requires teacher coordination of students and computers.

An operating system, "STUDENT", must be loaded into each student station via the SYSTEM command from cassette tape or NETWORK 1 or NETWORK 2 before NETWORK 3 operation begins. A planned ROM initializer will eliminate this cassette load and allow the student station operating software to be loaded directly from the Host disk. When the ROM is available, it can be installed at any Radio Shack Repair Center. (Contact your local Radio Shack Store or Computer Center for details.)

Student stations will need to have adequate memory for NETWORK 3 operating software and student application programs. In a 16K student station, STUDENT and NBASIC programs occupy about 9K of memory, leaving 7K. If more than 7K of student application program space is needed, additional RAM may be purchased in 16K increments up to a maximum of 48K.

The NETWORK 3 equipment required for this system set-up includes the NETWORK 3 CONTROLLER (26-1212), the NETWORK 3 Operating System (26-2775), the appropriate length RS-232 Cable for each computer, and a 6" RS-232 Card Extender (26-1497) for each computer. Also included in this set-up would be the host and student station computers and installation of RS-232 interface boards in all computers.

Many Radio Shack Software programs are fully compatible with the Network 3. Radio Shack educational software packages compatible with the initial version of the Network 3 Operating System include:

- K-8 MATH PROGRAM

- K-8 MATH WITH STUDENT MANAGEMENT
ESSENTIAL MATH, VOLUMES ONE AND TWO
EUCLID GEOMETRY TUTOR

- ADVANCED GRAPHICS

- VECTOR ADDITION

- GRAPHICAL ANALYSIS OF EXPERIMENTAL DATA

- INTERPRETING GRAPHS IN PHYSICS

- INVESTIGATIONS IN INTEGRAL CALCULUS

- PLANE ANALYTIC GEOMETRY

- AlphaKey

Since the NETWORK 3 is for use in an educational setting, it does not support the Radio Shack Business Programs. The major reason for this is that only one student station may access and open a particular file at one time, which prevents all other student stations from accessing that file until it is closed. Different student stations may access different files at the same time. A special version of SCRIPSIT for Education and a Business Education Series is being planned which will be compatible with the NETWORK 3.

Radio Shack offers educators a variety of different services. Our National Bid Department handles the school bids and offers the best possible price quotations on quantity

purchases of TRS-80 equipment and courseware.

Our National Lease Department is able to put the TRS-80 into classrooms with terms to meet almost any budget. Because you can lease everything at once, instead of buying equipment gradually, you can provide more hands-on experience for the students immediately.

The operation of a TRS-80 requires more training than just learning to operate a movie projector or a tape recorder. Since some instruction and training is desirable, Radio Shack offers free training sessions to teachers and school administrators. Our instructors are able to conduct these classes to fit your schedule. Contact your nearest Radio Shack Computer Center for details.

Most of our Radio Shack Computer Centers have a Customer Service Representative to assist customers with any problems which may develop. If you desire more help than your store is able to provide, Computer Customer Service has an Educational Software group (817-390-3302) for helping educators and users of the educational packages. These people are able to help you implement a computer learning program, answer questions on new products, and also help to apply packages to your individual classroom. The Educational Software group will help talk you through any problems with existing programs. However, they are not able to help you modify any of the existing software programs. They realize the importance of the educational use of computers and are dedicated to helping you get the most out of your TRS-80 classroom.

Accept the challenge of the Computer Age—become informed and learn with the new generation—use your Radio Shack Resources to help educate and equip you to deal with this age of advancement in computer technology.

Computer Customer Services Address and Phone Number

8AM to 5PM Central Time

Computer Customer Services
400 Atrium, One Tandy Center
Fort Worth, Texas 76102

These are New Numbers and Were Effective Aug. 27

Model I/III Business Group	817-870-2041
Model II/16 Business Group	817-870-2042
Languages and Compilers	817-870-2044
Color and Pocket Computer Group	817-870-2150
Hardware and Communications Group	817-870-2571
Educational Software	817-390-3302
Games, Books, and New Products	817-870-2271

Figure It Out With Alice and Mac


The TRS-80 Color Computer will be seen on some Educational Television stations this fall in a new mathematics series called "Figure Out". Tandy Corporation donated the Color Computer to the station that produced the show—the

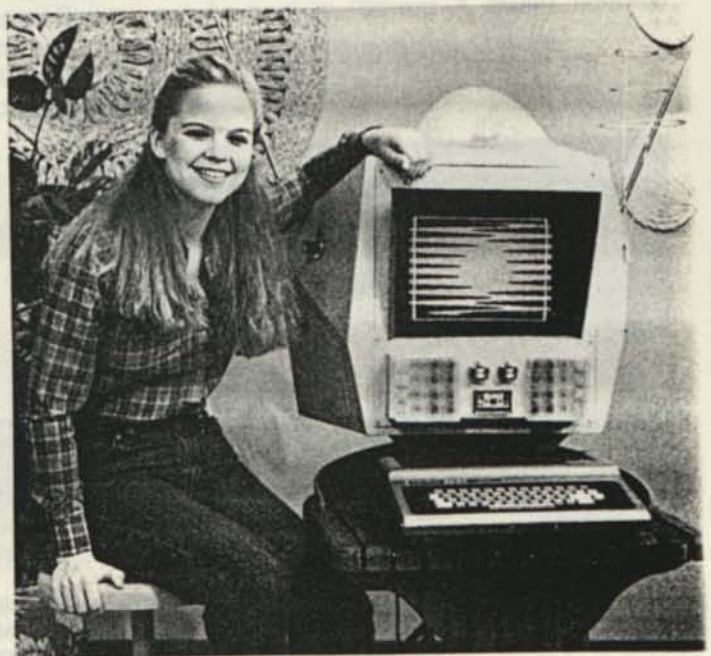
Mississippi Authority for Educational Television.

"Figure Out" presents the adventures of Alice, a young 'computer whiz', and Alice's business partner Mac, a computer with a mind of his own and a lot of personality. The TRS-80 Color Computer is part of Mac's body, along with a large monitor decoratively enhanced by the art department at Mississippi ETV.

Alice and Mac work together in Alice's computer programming company (called "Alicesystems"), located in Alice's apartment in an old observatory. Some of their adventures involve trying to get the business going in spite of an villainous competitor who is trying to destroy Mac. Comic features of the show include Mac's attempts to flirt with Alice's refrigerator.

The math concepts presented are part of the story, emphasizing the practical applications of math. Alice and Mac figure out problems such as how much they owe the air conditioner repair man, how much money Mac will make if he goes to Hollywood to become a singing star, and what the shortest route is to a destination. Usually the characters figure out the problem, a narrator explains the concepts involved and the viewer gets to solve practice problems. Problems are displayed and solved on the television screen using large graphic numerals, pencil and paper, and at times a calculator. Concepts from numeration and rounding to multiplication and division are covered.

At the same time as they entertain, the lessons cover a lot of math in a cumulative and systematic way. Four lessons are on numeration and rounding, one is on addition, three lessons each are on subtraction and multiplication, and division is covered in four lessons. Because the series emphasize use of traditional tools to solve math problems, Mac's role as character is at least as important in "Figure Out" as are his actual capabilities as a computer. But "Figure Out" makes the important point that computers don't need to be cold, frigid, or enigmatic machines. With the right programming, a computer can be a warm and friendly partner or helper. Alice and Mac prove it because, as the narrator says, "together they can figure out anything"! 



The Multiple Choice Puts You to the Test

Editor's Note: The CompuServe Information Service is one of the largest information and entertainment services available to owners of personal computers and computer terminals. With each issue of TRS-80 Microcomputer NEWS, various features of CompuServe will be discussed.

The CompuServe Information Service is sold at Radio Shack stores nationwide and in Canada.

You want to attend graduate school, and wonder how you'd fare on the entrance exam. Your child has mastered reading, and he's anxious to show off his news skills. Or, what if the world's most accomplished trivia buff has nothing on you, and you want to prove it?

The answers can be found in one of CompuServe's home services offerings — The Multiple Choice (TMC).

TMC, number eight on the Home Services menu, offers tests and games the entire family can enjoy together. It was designed that way by its creator, Howard Millman, a professor of educational psychology at the University of Maryland.

Millman had heard about CompuServe, and the idea of learning by computer intrigued him. He thought about the value of a program that would make learning fun and educational, and that families could enjoy together. He bounced some ideas off of friends and family, contacted CompuServe with a proposal, and TMC was born about a year ago.

Millman, with the help of family and friends, composes all TMC questions himself. A friend provides programming advice and helps him program TMC from his home.

A glance at the TMC menu shows plenty of choices. If it's pure fun you're after, play "Trivia Unlimited", which lets players compete against themselves and others by answering a wide variety of trivia questions. If your recall skills can stand a little brushing up, the 10-question warm up quiz is probably your best choice. But if you're the type who pores over little-known facts, the "sudden death" option is for you. Questions are much more difficult, and the scoring method truly tests your mettle: you keep answering questions until you've missed three; then the test ends and your score is compared with the top three scores by all other test takers.

The children can choose a TMC designed especially for them. "TMC for Kids!" has five tests and games that are fun and help perfect already-learned skills. "Silly Fill-Ins" asks kids seven and older to fill in missing words in a story. A sports quiz tests knowledge of basketball, baseball, soccer, football and tennis rules and terms. "J*U*M*B*L*E*D Words" asks players to recognize disguised words, while another game tests knowledge about the 50 states. And if that grade-schooler of yours knows what color Kermit's bicycle was in "The Muppet Movie", they'll like "Trivia for Kids".

In a more serious vein, Millman also created an intelligence quotient exam and one that provides some insights

into who you are. "The TMC Intelligence Test", while designed to be similar to an actual IQ exam, is not meant to be a substitute for one. But the results "should give you a fair estimate of how smart you are", Millman maintained.

"The TMC Personality Profile" was created based on general personality patterns, Millman said. Questions cause the test taker to look at his psychological, physical and spiritual self, and measure how he views relationships and decision making. At the test's end, the computer provides a chart that shows how well the test taker is caring for himself in each of eight areas examined.

Perhaps the most serious TMC category is "TMC Analogies", which simulates a national exam taken by graduate school applicants. Questions are close to the test in length, type of analogies and trick questions. Test takers can choose a practice quiz or pick the more difficult timed exam.

After investigating any of the TMC categories, Millman invites users to tell him what they thought via "TMC Feedback". Millman personally reads all the "letters" he receives and answers many of them.

"I get mostly good comments", he said. "People give me lots of pep talks. They said 'thanks, I needed that', or 'keep up the good work.'"

Keeping up the good work is something Millman is always striving for. He's now in the process of programming 10 more TMC categories into the system, and a sneak preview reveals that there will be something for everyone to enjoy.

Questions and comments about the CompuServe Information Service can be sent to Richard A. Baker, editorial director, CompuServe Information Service, P.O. Box 20212, Columbus, Ohio 43220 or through Feedback on the User Information menu.



University of Maryland Professor Howard Millman reviews questions for the Multiple Choice on CompuServe with his daughter.

Window On Wall Street

The Corporate Earnings Estimator

Editor's Note: With over 45,000 subscribers, Dow Jones News/Retrieval is the leading provider of online business and financial information. We will keep you up-to-date on new software, new data bases and new prices in upcoming issues of TRS-80 Microcomputer News. Dow Jones News/Retrieval is sold at Radio Shack stores nationwide and in Canada.

Wouldn't you like to see the same earnings per share estimator that is available to professional investors before you decide to buy or sell a stock?

Try the Corporate Earnings Estimator. It is for those of you who do not want to rely on a single broker for investment advice, and who are aggressive enough to decide, on your own, how your investment portfolios should be structured.

The Corporate Earnings Estimator makes it possible for the first time in the history of the stock exchange, for the average investor to see the full range of Wall Street brokerage estimates.

Before the service was offered by News/Retrieval, only major institutions were able to make decisions based on earnings per share (eps) estimates that were as complete, up-to-date and accurate as possible. But now, you can see the same information yourself.

The Corporate Earnings Estimator provides a statistical summary of brokerage analyst eps from the top firms on Wall Street, which include 1,200 analysts at 50 major brokerage firms.

"We receive virtually all of the brokerage research published in the United States and we have a staff that goes through it every day", said Leonard Zacks, founder of Zacks Investment Research, Inc., which provides the service to News/Retrieval.

"You get all of the estimates in a statistical, summarized form . . . the mean, the high . . . the low," Zacks said. "Most individual investors' buy/sell decisions are made on incomplete information. By looking at the eps estimates, you can make better decisions to buy stock".

"We pull out the estimates from the brokers' reports. Then we send Dow Jones News/Retrieval tapes each week", he said, who in turn make the information available to News/Retrieval subscribers.

Zacks initially founded the company in Chicago in 1977 to assist institutional investors in using quantitative technology in their investment processes.

Before founding the company, which now provides brokerage analyst eps estimates, rankings and software systems, primarily to institutional investors, Zacks was Vice-President of A.G. Becker and Company, Inc., which was

responsible for the design of quantitative research service performance measurement services and econometric analyses.

Zacks holds a Ph.D. from M.I.T. and is the author of several articles in the quantitative and financial areas.

To retrieve Corporate Earnings Estimates on the Dow Jones News/Retrieval service, just type: //EARN and press the <ENTER> key.

Once in the data base, access a particular company by typing the company's stock symbol (located in the Dow Jones News/Retrieval Operating Guide and Directory Symbols), followed by <ENTER>. To view a second page of estimates for the following fiscal year, just press the <ENTER> key again.

```

TADY CORP
--FISCAL YEAR ENDS 6/82

EARNINGS PER SHARE ESTIMATES
--MEAN 2.22
--HIGH 2.48
--LOW 2.12
NUMBER OF ANALYSTS 16
P/E RATIO (ESTIMATED EPS) 14.49

PAST EARN PER SH ESTIMATES (MEAN)
--WEEK AGO 2.21
--13 WEEKS AGO 2.21
--26 WEEKS AGO 2.17

PRESS RETURN FOR NEXT PAGE
  
```

Screen 1

```

TADY CORP
--FISCAL YEAR ENDS 6/83

EARNINGS PER SHARE ESTIMATES
--MEAN 2.81
--HIGH 3.88
--LOW 2.67
NUMBER OF ANALYSTS 13
P/E RATIO (ESTIMATED EPS) 11.41

PAST EARN PR SH ESTIMATES (MEAN)
--WEEK AGO 2.81
--13 WEEKS AGO 2.88
--26 WEEKS AGO 2.72
  
```

Screen 2

You can get about two stocks per minute when you access the service with a personal computer or terminal.

Peter Sohn, who works with the New York based money firm, Tweedy, Browne, Inc., said he uses the data base daily. "It comes in as a second opinion", Sohn said of the service. "The information on the estimator gives you kind of a tip-off sometimes to something that someone sees and the other ones do not see . . . You get diversity and you get more information and it is accessible for the little guy".

Sohn said he checks three to five companies "at pop". "Both News/Retrieval and the estimator have achieved that ultimate distinction of becoming indispensable", he said.

Barney Ziv uses the estimator to retrieve earnings estimates on portfolios managed by his Chicago based investment firm.

"We are fairly new subscribers, but we like the service. It provides a service that we will probably use increasingly", he said.

Zacks said brokers do not see the service as a threat. In fact, he said they "love it," because they can get all of the information also.


With other services, Zacks said you do not get information that is particularly up-to-date. Sometimes, it can be as many as three months old, he said.

But with the Corporate Earnings Estimator, the information is updated daily and reports for News/Retrieval are generated weekly.

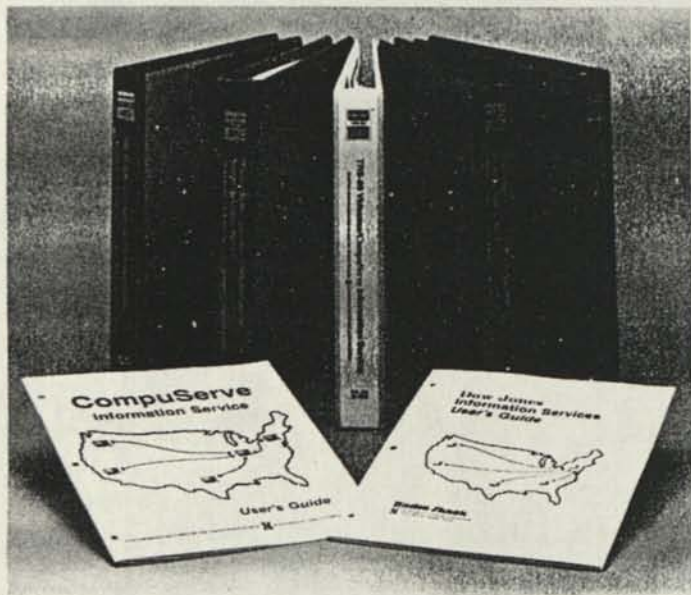
If you would like more information on Dow Jones News/Retrieval, or have any questions on the service, please call the Dow Jones Customer Service Hotline at:

1-800-257-5114

in New Jersey and Canada Call:

609-452-1511. 

DOW JONES NEWS/RETRIEVAL[®]



microRESEARCH Problems

Richard V. Andree
microRESEARCH
P.O. Box 2910
Fort Worth, Texas 76113-2910

Before we go on with microRESEARCH problem #2, here is a brute force program to solve problem #1:

```

10 CLS
20 FIRST=SQR(1023456789)
30 FIRST=INT(FIRST)
40 LAST=SQR(9876543210)
50 LAST=INT(LAST)
60 FOR I=FIRST TO LAST
65 FLAG$="YES"
70 TEMP#=I*I
75 'PRINT@0,TEMP#
80 TEMP$=STR$(TEMP#)
90 TEMP$=RIGHT$(TEMP$,LEN(TEMP$)-1)
100 IF LEN(TEMP$)<10 THEN 999
110 IF LEN(TEMP$)>10 THEN I=LAST
    : GOT0999
120 FOR J=0 TO 9
122 F(J)=0
123 NEXT J
125 FOR J=1 TO 10
130 J$=MID$(TEMP$,J,1)
140 K=VAL(J$)
150 F(K)=F(K)+1
160 IF F(K)>1 THEN FLAG$="NO": J=10
170 NEXT J
180 IF FLAG$="NO" THEN 999
190 PRINT C,I,TEMP#
200 C=C+1
999 NEXT I

```


This program was written on a TRS-80 Model II by Microcomputer News Editor Bruce Elliott, and will list the integers whose squares contain each of the ten digits (0-9) exactly once. I call it a "brute force" program, because there is nothing elegant about it. The program simply starts with appropriate values for FIRST and LAST, that is the smallest and the largest possible integers that might yield a proper result, and then uses a FOR-NEXT loop to examine every value in-between. How does this compare with your solution? If I get a more elegant solution in the next month or two, I will publish it.

microRESEARCH problem #2.

A **palindrome** is a phrase or sentence which reads the same forward as backward if you disregard spaces and punctuation. Example: "Ma is as selfless as I am."

There are also number palindromes. It is fairly easy to find perfect squares which are palindromes.

$(202)^2 = 40804$; $(264)^2 = 69696$. There are infinitely many palindrome squares. However, palindrome squares having an **even** number of digits, such as: $(836)^2 = 698896$ are fairly rare. Only three palindrome squares having an **even number of digits** are known today. All three are discoverable using your TRS-80. Do so.

Possible extensions: Although palindromic cubes and palindromic fourth powers also exist $(111)^3 = 1367631$ and $(11)^4 = 14641$, they are harder to find. With the exception of $1^k = 1$, no palindromic k-th powers for $k > 4$ are yet (1982) known to your author. Some reader may find one. 

Modem II Questions and Answers

Paul Schreiber
Radio Shack Research and Development

One of the handiest accessories a computer user can have is a modem. The Modem II, a Radio Shack top of the line modem, has many features that are not to be found in other modems. The most useful features are the auto answer/auto dial modes and two types of test modes. The Modem II has a custom microprocessor which makes things easy from a user's standpoint since the modem follows the micro's code, and the modem's operation is predictable. When programming the modem, some users have difficulty in getting it to "behave" properly. Perhaps the following questions and answers will give programmers a better understanding of the Modem II.

Q. WHAT IS THE PURPOSE OF THE FORCE DTR SWITCH?

A. The force DTR switch was installed to allow the Color Computer and other 3-wire RS-232 schemes to program the modem by use of the asterisk (*) command.

Q. WHAT ELSE DOES THE FORCE DTR SWITCH DO?

A. The DTR line is used as a master enable line and to place the modem in the programming mode. When the Force DTR switch is on

1. It forces CD (carrier detect). This is a Color Computer specification.
2. The modem's software driver cannot use CD to determine if another modem has answered.

See page 19 of the TRS-80 Modem II Operation Manual for further information on carrier detect.

3. It allows the Modem II to go into the programming mode only when it receives an * at the proper time.

The decision of when to use DTR has to be determined by the individual programmer based on his/her application.

Q. WHY WON'T THE MODEM II WORK WITH "XXX" BULLETIN BOARD SOFTWARE?

A. Software written to program a Brand X modem will not work with a Brand Y modem. Due to the manufacturing design differences in modems, Modem II software will not be compatible with another manufacturer's modem. In fact, no two manufacturers modems program alike.

Q. THE MODEM II CAN'T BE PROGRAMMED THE WAY THAT I EXPECTED IT TO. WHAT SHOULD I DO?

A. Read the manual carefully, study the flowchart in the back of the manual, and write your program to conform to these instructions. If you do, the modem can be programmed without any difficulty.

The majority of the difficulties experienced with the Modem II stem from pre-conceived notions on how it should respond. Be sure that you are writing software to fit the modem and not trying to make the modem fit the software.

Q. HOW CAN I MAKE THE MODEM IGNORE INCOMING RINGS WHILE MY SOFTWARE IS DOING OTHER THINGS?

A. Following are two procedures to make the Modem II ignore incoming rings.

1. When the Modem is in LOCAL Test, the hardware resets the ring counter to zero so all incoming rings are ignored. To accomplish this under software control, send a QX (local test command) to start the test and then using software, drop DTR (the force DTR switch must already be off) to get out of local test.
2. If you must have force DTR on, then go into the programming mode. While in the programming mode, the modem will not process incoming rings. By sending an X, you will exit the programming mode, but remember that whatever is stored in the modem at the time the X is sent will be executed.

Q. I CAN'T PROGRAM THE MODEM WITH MY "XXX" TERMINAL SOFTWARE.

A. This problem usually occurs because of a lack of knowledge of the data format. The Modem II recognizes only upper case, 8 bit, no parity characters.


Q. WHEN USING FORCE DTR OFF, THE MODEM SOMETIMES SEEMS TO "HANG UP". WHY?

A. Usually when the modem appears dead in that it no longer responds to commands, it is because the computer is asserting or unasserting DTR for either the wrong amount of time or at the wrong time.

A good example of this may occur when the programmer uses DTR to disconnect the Modem II from the phone line after completing data transfer. When the DTR is dropped, the modem will promptly disconnect. Now, if DTR is kept low longer than 3 seconds (refer to the flowchart in the manual), the modem drops into the programming mode. If the unwary programmer fails to realize this and asserts DTR without first sending an X to exit the programming mode, the modem may appear to hang up. Be careful when using DTR to turn the modem on and off or you may accidentally go into the programming mode.

If you wish to hang up the phone, it is better to send a space disconnect than it is to drop DTR. This eliminates the chance of accidentally going into the programming mode.

Q. HOW DO I GET THE SPACE DISCONNECT TO WORK?

A. Check to see what you are actually sending. The space disconnect does not refer to an ASCII space (20H) but a constant LOW output from the UART to the RS-232 drivers without start or stop bits. The internal microprocessor of the Modem II checks the status of the data every 75 ms during the 1.5 second and 3 second disconnect intervals. If during one of these times the data goes HIGH, then the timers are reset. Each computer/terminal's UART sends the space by different means, so check the technical specification of your system. 

Personal Time Savers

VisiCalc command files can make your model building sessions quicker and more convenient.

Most people who turn to financial model building with the VisiCalc program say it saves them vast amounts of time when compared to the alternative methods of financial analysis with pencil, paper, and calculator. Yet you can save even more effort by spending a little time now thinking about frequently used formulas and often-repeated commands and converting them into VisiCalc command files.

How many times have you found during a model building session that you needed to insert ten rows — then had to key each insert in separately? How many times have you had to key in the same complex formula to a number of your models, because you thought there was no way of transferring a single formula from one worksheet to another? (Transferring values, by the way, from one sheet to another for consolidation is simple.)

You can go far towards eliminating this type of computer "grunt work" by using VisiCalc command files. When the VisiCalc program loads in a command file, it acts as though it were receiving commands typed at the keyboard. While command files can not do everything (for example, they cannot sequentially load a number of worksheets into one, or save worksheets for you), they can:

- perform multiple row and column inserts (at your cursor position)
- perform multiple row and column deletes
- format specified areas of your worksheet
- blank specified areas of your worksheet
- build a complex formula in a specified area of your worksheet

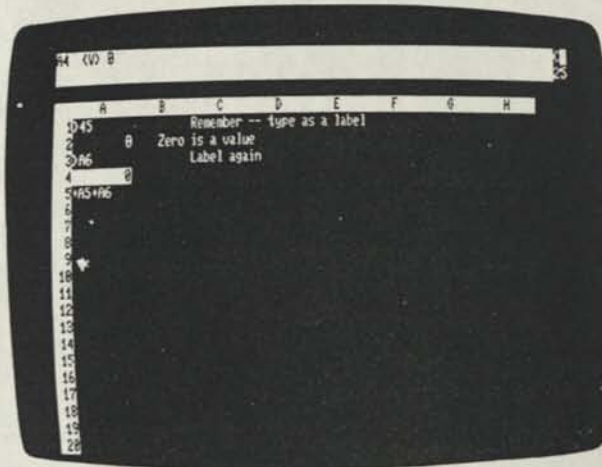
VISICALC[®]

BUILDING A COMMAND FILE

Creating a set of VisiCalc command files which suit your needs starts with the VisiCalc program and a blank, initialized data diskette. The data disk will hold your final library of VisiCalc command files, each one identified by a descriptive filename.

Let us start with a simple experiment which will show how well the VisiCalc program listens to your command files. We will type onto the blank VisiCalc sheet labels which resemble keyboard commands such as /F, >A5, etc. All these commands must be typed in as labels. If you wish to put a value into a formula (such as 0, for starters), it may be typed in as a value. Here is a simple addition formula which will show what we mean.

On a blank VisiCalc worksheet, bring your cursor to A1. Type the following:



Screen 1.

What we have done here is use the VisiCalc program to create text which looks like the kind of commands that would be typed into the computer keyboard. The row and column positions we are concerned with here are those appearing in the text. They will control where the command file appears when it is finally loaded in. We began writing at position A1 for convenience; the command file itself will begin the addition at A5, just as we commanded it to.

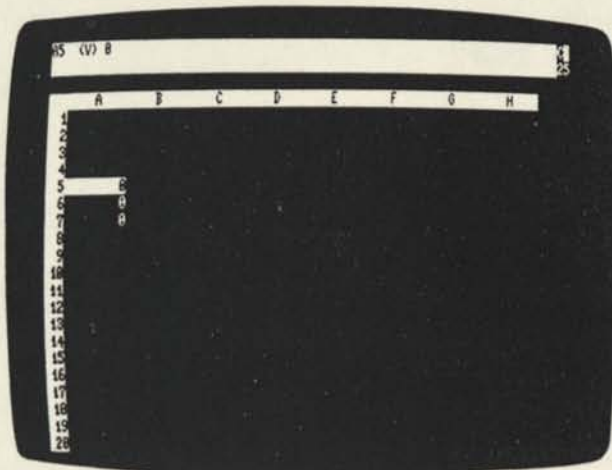
One cautionary note: While you may write your command file instructions from left to right on more than one row (one command to a cell position), you cannot write more than one column of commands vertically. This is because the VisiCalc program reads the commands from left to right until there are no more in a given row. Then it moves down to the next row and reads that from left to right. We have chosen to represent the command file as written down a column for clarity, but only one such column may be written.

Once our command file is written, we need to save it to diskette with the /PF command (Print to Filename), rather than the usual /SS (Storage Save). This alters the way the commands are stored on the diskette; the VisiCalc program will interpret them as commands (instead of labels).

Cautionary note: If you are planning to place a complex formula in a single cell, remember that /PF stores ONLY what appears on the VisiCalc screen. Make sure your column width reveals the entire formula, or the command file will not work when reloaded into the program.

In response to the program's request for a filename, name the file ADD, and give the LOWER RIGHT coordinate as A7.

Next, clear your worksheet with the /CY command. With /SL select the file ADD to be loaded. You should see a screen similar to the chart below.



Screen 2.

Your cursor should be positioned over A7, revealing the formula +A5+A6. Try different values in locations A5 and A6. The formula in A7 responds as though you had typed it in yourself . . . and in a sense, you have. That formula could have read @SUM(A5 . . . A6), or something even more complex.

Another note: the very nature of a formula such as +A5+A6 ties the answer to whatever is appearing in those two areas. However, if you have built your command file with no cursor direction commands (no >A5, for example), the formula loads by default wherever your cursor happens to be. You can always /E (edit) the formula to refer to the proper cell positions; but be careful where your cursor is located when you do load a positionless formula onto your worksheet.

The same kind of command file creation can be used to build a library of often used calculations for incorporation into your models. If your often used formulas are of great complexity and do require a number of cells because of values referred to or sheer size, leave "formula area" at the top (or bottom, right, left, etc.) of all your worksheets. Then you will be able to write specific cursor positions (>A10) into your command file for all your formulas.

When loaded in, your library of formulas will always appear at the same open area and may then be moved or replicated to another area of your current worksheet.

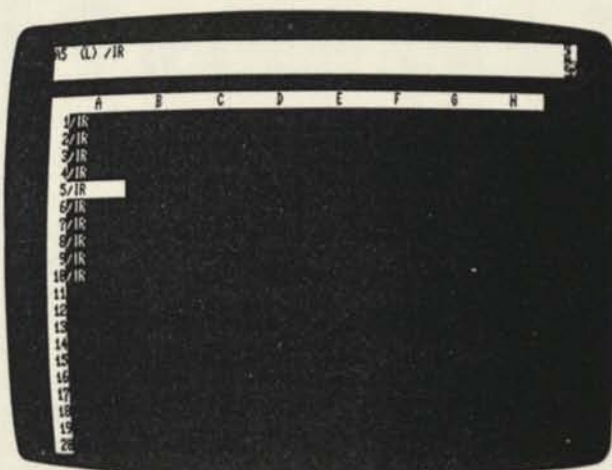
MORE BACKFLIPS AND SOMERSAULTS

Multiple row and column inserts can also be accomplished by VisiCalc command files. To free up ten rows, you would normally have to type the keystrokes / I R (in that sequence) ten times — 30 keystrokes — plus the few moments wait while your model moves down.

With a VisiCalc command file, you can reduce those keystrokes to / S L — until the correct filename shows, then ENTER. A command file for 10 row insertions is simple:

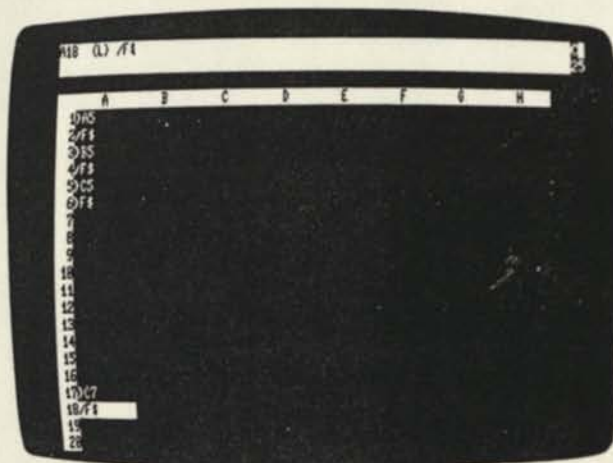
Remember that all these slash command entries must be typed as labels (preface each entry with a quotation mark " to trigger the label format).

Once again, save this INS10ROW file with the /PF command. As written, it will insert ten rows at whatever your current cursor position happens to be. With the addition of the label >C27 at the beginning, for example, it can be made to insert ten rows only at location C27.



Screen 3.

If your models require that the two-decimal place design format (/F\$) appear in an area bounded by A5, C5, and C7, then you can write the following command file:



Screen 4.

At first this may appear laborious. Think though, what you are typing here — only once — is the same sequence of keystrokes you would type in every model in which you needed them. Being able to format an area, rather than having to format globally and delete formats (/FD) or having to format cells individually over a large area is a time saver.

And that is what your TRS-80 and the VisiCalc programs are all about, isn't it? Saving time.

If you are using VisiCalc command files to help setting up and using VisiCalc models, we would like to hear about it, and may share your experience with other VisiCalc users here in the TRS-80 Microcomputer News.

Send a note to:

VisiCalc Page Editor
c/o Microcomputer News
P.O. Box 2910
Fort Worth, TX 76113-2910

Educational Programs

Customary-to-Metric

Todd Pruzan, Age 11
7920 Cindy Lane
Bethesda, MD 20034

Three months ago our family purchased a Model III. We have gotten great use out of it, and it is fun to work with. Now I have written a customary-to-metrics program. You can get a head start on the metric system with it. First it asks you several familiar words (miles, quarts, pounds, etc.). Type in the first letter—area is an exception to this—and it will ask you "MEASUREMENT?" Here you type the desired number, preferably a whole number from 1 to 100,000. Here you get an answer converted into metric. Lines 20 - 50 are skipped. You can use them up with REMs or PRINTs.

The sample run:

```
CUSTOMARY SYSTEM CONVERTED TO METRIC
BY TODD PRUZAN
LENGTH, AREA, MASS OR VOLUME? M ___
OUNCES, POUNDS OR SHORT TONS? P ___
MEASUREMENT? 17 ___
7.65 KILOGRAMS.
ANOTHER CONVERSION? Y ___
LENGTH, AREA, MASS OR VOLUME? A ___
SQUARE INCHES, SQUARE FEET, SQUARE YARDS,
SQUARE MILES OR ACRES? SQUARE F ___
MEASUREMENT? 9 ___
.81 SQUARE METERS.
ANOTHER CONVERSION? N ___
READY ___
```

The program

```
10 CLS
20 PRINT "CUSTOMARY SYSTEM CONVERTED TO METRIC"
   : PRINT "BY TODD PRUZAN"
60 PRINT "LENGTH, AREA, MASS OR VOLUME";
70 INPUT M$
80 IF M$="L" THEN 120
90 IF M$="A" THEN 200
100 IF M$="M" THEN 260
110 IF M$="Y" THEN 300
120 PRINT "WHAT DO YOU WANT CONVERTED?"
130 PRINT "INCHES, FEET, YARDS, OR MILES";
140 INPUT I$
150 PRINT "MEASUREMENT";
   : INPUT MT
160 IF I$="I" THEN PRINT MT*25;" MILLIMETERS."
   : GOTO 350
170 IF I$="F" THEN PRINT MT*30;" CENTIMETERS."
   : GOTO 350
180 IF I$="Y" THEN PRINT MT*.9;" METERS."
   : GOTO 350
190 IF I$="M" THEN PRINT MT*1.6;" KILOMETERS."
   : GOTO 350
200 PRINT "SQUARE INCHES, SQUARE FEET, SQUARE YARDS,
SQUARE MILES OR ACRES"
   : INPUT SQ$ : PRINT "MEASUREMENT";
   : INPUT A
210 IF SQ$="SQUARE I" THEN PRINT A*6.5;" SQUARE
CENTIMETERS."
   : GOTO 350
```

```
220 IF SQ$="SQUARE F" THEN PRINT A*.09;" SQUARE
METERS."
   : GOTO 350
230 IF SQ$="SQUARE Y" THEN PRINT A*.8;" SQUARE
METERS."
   : GOTO 350
240 IF SQ$="SQUARE M" THEN PRINT A*2.6;" SQUARE
KILOMETERS."
   : GOTO 350
250 IF SQ$="A" THEN PRINT A*.4;" SQUARE HECTOMETERS."
   : GOTO 350
260 PRINT "OUNCES, POUNDS, OR SHORT TONS ";
   : INPUT O$
265 PRINT "MEASUREMENT";
   : INPUT B
270 IF O$="O" THEN PRINT B*28;" GRAMS."
   : GOTO 350
280 IF O$="P" THEN PRINT B*.45;" KILOGRAMS."
   : GOTO 350
290 IF O$="S" THEN PRINT B*.9;" MEGAGRAMS (OR METRIC
TONS)."
   : GOTO 350
300 PRINT "OUNCES, PINTS, QUARTS, OR GALLONS";
   : INPUT P$
   : PRINT "MEASUREMENT";
   : INPUT C
310 IF P$="O" THEN PRINT C*30;" MILLILITERS."
   : GOTO 350
320 IF P$="P" THEN PRINT C*.47;" LITERS."
   : GOTO 350
330 IF P$="Q" THEN PRINT C*.45;" LITERS."
   : GOTO 350
340 IF P$="C" THEN PRINT C*3.8;" LITERS."
350 PRINT "ANOTHER CONVERSION";
360 INPUT X$
370 IF X$="Y" THEN 60
380 END
```

Physical Science Program

Allan Freeman, Age 14
100 5th Place
Pleasant Grove, Alabama 35127

I own a Model III and a Line Printer VII. I wrote this program to show how helpful a computer can be when your studying for a test.

```
2 CLS
5 REM
   : This program was written for the Model III
10 REM December 5, 1981 (No Rights Reserved)
15 REM Written by Allan Freeman
20 POKE 16916,4
22 POKE 16412,1
25 PRINT TAB(20);"Physical Science (True or False)"
26 PRINT TAB(20);"Written by: Allan Freeman"
27 CLEAR 100
28 PRINT STRINGS(64,191)
30 INPUT "1. A metric ruler, graduated cylinder, and
microscope are tools used in science. (T/F)";A$
32 IF A$="T" THEN GOSUB 200
34 IF A$="F" THEN GOSUB 400
40 INPUT "2. A hypothesis is always true. (T/F)";A$
42 IF A$="T" THEN GOSUB 400
44 IF A$="F" THEN GOSUB 200
50 INPUT "3. The metric system of measurement is a
decimal system. (T/F)";A$
52 IF A$="T" THEN GOSUB 200
```

```

54 IF A$="T" THEN GOSUB 400
60 INPUT "4. A scientific law can never be disproved.
(T/F)"; A$

62 IF A$="T" THEN GOSUB 400
64 IF A$="F" THEN GOSUB 200
70 INPUT "5. Metric units have international use in
science. (T/F)"; A$
72 IF A$="T" THEN GOTO 218
74 IF A$="F" THEN GOTO 418
200 CLS:PRINT TAB(20);"C O R R E C T ! ! ! "
210 FOR X=1 TO 1000
215 NEXT X
: CLS
: RETURN
218 CLS
219 PRINT "C O R R E C T !
C O R R E C T !
C O R R E C T !
C O R R E C T !
220 PRINT TAB(15);"YOU'RE ALMOST AS SMART AS I AM."
221 GOTO 600
400 CLS
: PRINT TAB(20);"W R O N G ! ! ! "
410 FOR X=1 TO 1000
415 NEXT X
: CLS
: RETURN
418 CLS
419 PRINT TAB(5);"YOU MISSED THAT ONE."
: INPUT "DO YOU WISH TO TRY AGAIN?"; D$
420 IF D$="Y" GOTO 30
421 IF D$="N" GOTO 600
600 POKE 16916,0
: POKE 16412,0
: PRINT TAB(20); "ADIOS"
: PRINT TAB(20); "END OF PROGRAM"
: STOP

```

Directory Algorithm

Thomas J. Kelanic
935 Lilly Lane
Pittsburgh, PA 15221

I became interested in multiple-page directories while writing a computer program to generate daily absence bulletins for my school. Under the old system, a secretary collected attendance rosters from all homerooms, then typed a list of absent students. The list was divided into separate sublists, one for each homeroom. A teacher wishing to check-up on a student first had to know, or find elsewhere, the student's homeroom number. After finding the sublist for the given homeroom, the teacher then looked for the student's last name, alphabetically.

I wanted to improve on the old system by generating a daily directory that looked like a telephone directory. All the inquiring teacher would have to know would be the last name of the student, since there would be only one alphabetized list of names. Following each name would be whatever other information was required, such as homeroom number and cumulative absences. To avoid the possible future necessity of having to go back and re-write my program to take into account changed requirements, I decided that the number of pages, columns, lines per page, and page dimensions should all be variable.

The following program does not produce daily absence bulletins. It will produce any multiple-page directory listing required from a string array in RAM. I believe my program is a general algorithm for producing directories.

In lines 100 - 130, the user's string array is simulated in RAM. The 676 strings "AA—" through "ZZ—" are produced in approximately one minute and forty seconds. These lines must be omitted if the user has their own A\$ array in RAM.

In lines 170 - 230, not counting 220, the given environmental constants are set. These lines could easily be replaced by LETs. Lines 230 - 290 are the program variable used to specify the directory required, and are the parameters used in the examples which follow. In a real application, N, in line 230, would be constant. However, in this demonstration program N is variable, since the user may not wish to see all 676 strings.

No error-checking is provided in my program. If the user asks for something physically impossible, the program will "bomb". In this case, the user may issue a "GOTO 230" command, and should reconsider his inputs.

To get an idea of the flexibility of the program, the user may try the following two examples. Assuming the constant input were "actual max." = 5, "highest (fixed)" = 63, "lowest (fixed)" = 0, the user is ready.

Example #1: Use 100 strings, 63 highest, 34 lowest, width, 6 columns, and 15 lines per page.

Example #2: Use of 61 strings, 63 highest, 0 lowest, 3 width, 2 columns, and 10 lines per page.

To guarantee at least one space between columns, use field-width one more than the actual maximum string length. For the example strings of length 5 a field-width of 6 will do.

If you have a line-printer, you must use the correct value for your printer in lines 180 - 210, and change the two PRINTs in lines 600 and 610 to LPRINTs. Also, you may wish to insert some LPRINTs between lines 630 and 640.

```

10 '
20 '          DIRECTORY ALGORITHM
30 '
40 '          THOMAS J. KELANIC
50 '          935 LILLY LANE
60 '          PITTSBURGH, PA. 15221
70 '
80 '          SIMULATE USER ARRAY A$
90 '
100 CLEAR 6000
: T6=26
: DIM A$(T6*T6)
110 FOR I=1 TO T6
: FOR J=1 TO T6
120 A$((I-1)*T6+J)=CHR$(I+64)+CHR$(J+64)+"--"
130 NEXT J
: NEXT I
140 '
150 '          GET PARAMETERS
160 '
170 INPUT "ACTUAL MAX. STRING LENGTH"; W2
180 PRINT "HIGHEST PRINT POSITION ";
190 INPUT "(FIXED) FOR OUTPUT DEVICE"; W9
200 PRINT "LOWEST PRINT POSITION ";
210 INPUT "(FIXED) FOR OUTPUT DEVICE"; W0
220 DIM B(INT((W9-W0+1)/W2)+1)
230 INPUT "NUMBER OF STRINGS TO BE USED"; N
240 INPUT "HIGHEST PRINT POSITION TO BE USED"; W8
250 INPUT "LOWEST PRINT POSITION TO BE USED"; W1
260 INPUT "COLUMN FIELD-WIDTH TO BE USED"; W
270 PRINT "SELECT I TO"; INT((W8-W1+1)/W);
280 INPUT "COLUMNS"; C
290 INPUT "NUMBER OF LINES PER PAGE"; L
300 '
310 '          FOR EACH DIRECTORY
320 '
330 K=L
: R=0
: P=INT((N-1)/(C*L))+1
340 CLS
: PRINT P;"PAGES WILL BE PRODUCED"
350 PRINT "PRESS (ENTER) FOR NEXT PAGE"
360 FOR I=1 TO 600
: NEXT I
370 '
380 '          FOR EACH PAGE
390 '
400 FOR H=1 TO N STEP C*L

```

```

410 CLS
   : B(1)=H
   : P2=INT((B(1)-1)/(C*L))+1
420 PRINT "PAGE NUMBER"; P2
430 FOR I=1 TO 400
   : NEXT I
   : CLS
440 IF P2=P THEN D=N+1-B(1)
   : K=INT(D/C)
   : R=D-K*C
450 '
460 '          SET BREAKPOINTS
470 '
480 FOR I=1 TO C
   : B(I+1)=B(I)+K
490 IF I <= R THEN B(I+1)=B(I)+1
500 NEXT I
510 '
520 '          OUTPUT PAGE
530 '
540 FOR I=0 TO K
   : FOR J=1 TO C
550 IF I+B(J)=B(J+1) THEN 630
560 T=INT(((W8-W1+1-C*W)/(C+1))*J+(J-1)*W+W1)
570 '
580 '          OUTPUT STRING
590 '
600 PRINT TAB(T);A$(I+B(J));
610 NEXT J
   : IF T+W-1 < W9 THEN PRINT
620 NEXT I
630 X$=INKEY$
   : IF X$="" THEN 630
640 NEXT H
   : PRINT
   : GOTO 230

```

```

70 INPUT "AREA CODE";AC
80 INPUT "PHONE NUMBER (FORMAT#####)";PN
90 T1=INT(PN/10000)
100 T2=PN-(T1*10000)
110 PRINT
120 PRINT USING$;N$,MB,DB,YB,AC,T1,T2
125 LPRT USING$;N$,MB,DB,YB,AC,T1,T2
130 PRINT
140 GOTO 30

```

When Line 20 is entered, use the down arrow to move to the second line after entering the second percent sign. Use the down arrow again after the sixth # sign following "BIRTHDATE : " to move to the next line.

Although I have not tried the concept of the Instant Recall program by Dwight Dager in the February issue, it brought to mind a game my current five-year-old has enjoyed for almost two years. It is like Concentration and she calls it her "Window Game" because of the screen format. It did not take many times for her to memorize the words and I feel the word recognition has assisted her in learning to read. I wrote the program on my Model I Level II TRS-80.

Bracken's Window Game

```

90 CLS
   : PRINT "WINDOW GAME FOR BRACKEN"
100 DIM N$(16), W$(16), W(16)
110 FOR N=1 TO 16
   : READ N$(N)
   : W$(N)=""
   : NEXT N
120 FOR N=1 TO 8
121 READ W$(N)
130 N1=RND(16)
   : IF W$(N1) <> "" THEN 130 ELSE W$(N1)=W$(0)
140 N2=RND(16)
   : IF W$(N2) <> "" THEN 140 ELSE W$(N2)=W$(0)
160 NEXT N
170 GOSUB 1000
180 GOSUB 2000
200 T=1
   : C=0
210 GOSUB 3000
400 IF W$(N1)=W$(N2) THEN 500
410 FOR Z=0 TO 1000
   : NEXT Z
420 GOSUB 2000
430 T=T + 1
   : PRINT@ 970, CHR$(30);
450 GOTO 210
500 REM
520 PRINT@ 970, CHR$(30);
   : PRINT@ 970, "MATCH";
525 FOR Z=0 TO 600
   : NEXT Z
530 W$(N1)=""
   : W$(N2)=W$(N1)
   : N$(N1)=""
   : N$(N2)=N$(N1)
535 FOR Z =1 TO 16
   : IF W$(Z) <> "" THEN 540 ELSE NEXT Z
537 PRINT@ 970, "GOOD JOB";
538 E$=INKEY$
   : IF E$=<> "" THEN RUN ELSE 538
540 C=C+1
   : T=T+1
   : GOSUB 2000
550 GOTO 210
1000 CLS
   : PRINT CHR$(23)
1010 FOR Y1=0 TO 11
   : FOR X1=0 TO 46 STEP 2
1020 X=15434 + 64 * Y1 + X1
   : POKE X,191
1030 NEXT X1, Y1
1040 RETURN
2000 N=1
   : FOR Y1=0 TO 3
   : FOR X1=0 TO 3
2010 X=142 + 192 * Y1 + 10 * X1
2020 PRINT@X, N$(N);
   : W(N)=X
   : N=N+1

```

String Formatting and the Window Game

Doug Fuge
2427 Hooohoi Street
Pearl City, HI 97682

Having just received my February issue of the newsletter and enjoying some of the programs, I thought I would comment on one article and send another of the programs I wrote for my youngest daughter. Both she and I got a big kick out of seeing our names in print in the December newsletter. Furthermore, it was rewarding to receive two inquiries from readers who had tried the Character Recognition program and were having minor difficulties getting it to run properly (Note: the rendition in the newsletter was accurate - cockpit errors were the problems).

The article More PRINT USING was most interesting to me as I am working on several programs where formatting is the key. While working with the concept advanced by Johnny Bond, and the various examples of the use of the TAB() function, I found I could utilize the STRING\$(,) function to insert spacing into the format string. Attached is Program #2 of Johnny Bond's article with line 20 changed to use the STRING\$(10,32) to indent the output on both the Video and my Line Printer IV. I am sure other readers really appreciate the power of the concept advanced in that article.

```

10 CLS
20 S$=STRING$(10,32)+"NAME :      "
   "+STRING$(10,32)+"BIRTHDATE : 00/00/00"
   "+STRING$(10,32)+"TELEPHONE : (000) 000 0000"
30 INPUT "NAME";N$
40 INPUT "YEAR OF BIRTH (2 DIGITS)";YB
50 INPUT "MONTH OF BIRTH (NUMBER)";MB
60 INPUT "DAY OF BIRTH";DB

```

```

2030 NEXT X1, Y1
2040 RETURN
3000 PRINT@ 906,T;" TRIES ";C" CORRECT";
3010 PRINT@ 970,"WINDOW?";
: O=1
: GOTO 3020
3015 PRINT@ 990," AND
: ";
3020 GS=INKEY$
: IF GS="" THEN 3020 ELSE PRINT GS;
3030 HS=INKEY$
: IF HS="" THEN 3030 ELSE PRINT HS;
3035 IS="" "+GS+HS
3040 FOR N=1 TO 16
: IF NS(N)=IS THEN 3060 ELSE NEXT N
3050 IF O=1 THEN PRINT@ 970, CHR$(30);
: GOTO 3010
3055 PRINT@ 990, CHR$(30);
: GOTO 3015
3060 IF O <> 1 THEN 3080
3070 O=O+1
: N1=VAL(GS+HS)
: Q=N1
: GOSUB 4000
: N1=Q
: GOTO 3015
3080 N2=VAL(GS+HS)
: Q=N2
: GOSUB 4000
: N2=Q
: IF N2=N1 THEN 3055
3090 RETURN
4000 IF Q<20 THEN Q=Q-10
: GOTO 4070
4010 IF Q<30 THEN Q=Q-20
: GOTO 4060
4020 IF Q<40 THEN Q=Q-30
: GOTO 4050
4030 Q=Q-40
4040 Q=Q+12
: GOTO 4070
4050 Q=Q+8
: GOTO 4070
4060 Q=Q+4
4070 PRINT@ W(Q), WS(Q);
4080 RETURN
9000 DATA "11","12","13","14","21","22","23",
"24","31","32","33","34","41","42","43",
"44"
9010 DATA "CAT", "DOG", "HAT", "DAD", "MAT", "RAT",
"MAD", "LOG"

```

```

: SET(39,Y)
: NEXT Y
40 FOR Y=18 TO 47
: SET(80,Y)
: SET(79,Y)
: NEXT Y
50 FOR X=1 TO 127
: SET(X,47)
: NEXT X
55 IF HH=0 THEN GOTO 70
60 FOR X=1 TO 127
: SET(X,15)
: NEXT X
65 IF HH=1 THEN GOTO 100
70 PRINT @994-14,STRING$(20,191)
75 HH=1
: GOTO 60
100 PRINT @128+64," (36 spaces)
":GOSUB 1000
102 PRINT @24,"*** CRUSH ***"
110 P=P+1
: PRINT @128+64,"PROBLEM #";P;" - ";
A;"";UI$;"";B;"="";
120 INPUT G
125 IF UI$="" THEN IF G<>A+B THEN GOSUB 2000
126 IF UI$="-" THEN IF G<>A-B THEN GOSUB 2000
127 IF UI$="X" THEN IF G<>A*B THEN GOSUB 2000
130 GOTO 100
1000 A=RND(TI)
: B=RND(TI)
1010 IF UI$="-" AND A<B THEN GOTO 1000
1020 RETURN
1520 RETURN
2000 IF C=67 THEN GOTO 50000
2005 FOR Y=16 TO 44
2010 SET(C,Y)
: SET(D,Y)
2020 NEXT Y
2025 C=C-3
: D=D+3
2030 RETURN
4000 CLS
: PRINT @64*4,"WHAT TYPE OF PROBLEMS WOULD YOU
LIKE"
4010 PRINT @64*5,"- SUBTRACTION, X MULTIPLICATION, +
ADDITION"
4020 INPUT"PLEASE TYPE X,+,-";UI$
4025 CLS
4030 RETURN
8000 PRINT @64*6,"UP TO WHAT # TABLES WOULD YOU LIKE
";
8010 INPUT TI
8020 IF TI >= 30000 PRINT "TOO HIGH"
: IF TI >= 30000 THEN GOTO 8010
8030 CLS
8040 RETURN
30000 PRINT@473 + 64, STRING$(4,128) + CHR$(159) +
CHR$(155) + CHR$(149)
30010 PRINT @473 + 128, CHR$(144) + STRING$(3,128) +
CHR$(131) + CHR$(151) + CHR$(129) +
STRING$(2,128) + CHR$(160)
30020 PRINT@64 * 3 + 473, STRING$(2,143) + CHR$(189)
+ CHR$(190) + STRING$(2,191) + CHR$(189) +
CHR$(190) + STRING$(2,143)
30030 PRINT@64 * 4 + 473, STRING$(3,128) + CHR$(171)
+ STRING$(2,191) + CHR$(151)
30040 PRINT@64 * 5 + 473, STRING$(2,128) + CHR$(160)
+ CHR$(186) + STRING$(2,191) + CHR$(181) +
CHR$(144)
30050 PRINT@64 * 6 + 473, STRING$(2,128) + CHR$(170)
+ CHR$(191) + CHR$(128) +
CHR$(128)CHR$(191)CHR$(149)
30070 PRINT@64 * 7 + 473, STRING$(2,128) + CHR$(138)
+ CHR$(159)CHR$(128)CHR$(128)CHR$(175)CHR$(133)
30080 PRINT@64 * 8 + 473, STRING$(2,128) + CHR$(130)
+ CHR$(131) + CHR$(128) + CHR$(128) + CHR$(131)
+ CHR$(129)
40000 RETURN
50000 CLS
50010 PRINT"YOUR LIFE WAS ";P;" PROBLEMS LONG"
50020 IF P > 20 THEN PRINT "I CAN TELL YOU KNOW YOUR
FACTS YOU ONLY MISSED 3 OUT OF ";P
50030 IF P < 20 THEN PRINT "YOU NEED TO LEARN YOUR
FACTS"
50040 PRINT
: PRINT
50050 INPUT "DO YOU WANT TO PLAY AGAIN";GAS
50060 IF LEFT$(GAS,1)="Y" THEN RUN
50070 CLS
: CMD"S"

```

Keep the good news coming. I look forward to each issue.

Crush—A Mathematical Game

Terry Myerson, Age 9
233 Pine Cone Trail
Ormond Beach, FL 32074

```

0 *****
1 *** TERRY MYERSON ***
2 *** GRADE 4,AGE 9 ***
3 *** ORMOND BEACH ***
4 *** FLORIDA 32074 ***
5 *****
8 CLS
10 C=76
: D=43
: P=0
15 GOSUB 8000
16 GOSUB 4000
17 GOSUB 30000
20 FOR X=40 TO 80
: SET(X,18)
: NEXT X
30 FOR Y=18 TO 47
: SET(40,Y)

```

Prime Number Generator #1

Gary M. Greer
31923 Angela Drive
Goshen, IN 46526

I wrote the following program on my TRS-80 Model I computer as a prime and factor number generator.

You are asking why would anyone be interested in primes? Well, I felt this would be an interesting task for a beginner such as myself.

Enough said. Well, okay... go to it Model I, spit out those primes!

```
10 'PRIME NUMBER GENERATOR
20 'BY GARY GREER...GOSHEN, IN.
30 CLS
40 INPUT "ENTER STARTING NUMBER"; N
50 FOR A=2 TO N
60 IF N<2 THEN N=N+1
   : GOTO 50
70 Q=N/A
80 IF Q=1 THEN 110
90 IF INT(Q)*A=N THEN 120
100 NEXT
110 PRINT TAB(10) "PRIME" N
   : N=N+1
   : GOTO 50
120 PRINT TAB(25) "FACTOR" N
   : N=N+1
   : GOTO 50
```

```
ENTER STARTING NUMBER? 3
PRIME 3          FACTOR 4
PRIME 5          FACTOR 6
PRIME 7          FACTOR 8
                FACTOR 9
                FACTOR 10
PRIME 11         FACTOR 12
PRIME 13         FACTOR 14
                FACTOR 15
                FACTOR 16
```

Prime Number Generator #2

Russ Euliano
201 Sweetwater Cove Blvd. N.
Longwood, FL 32750

This program is something I have been playing with for a while and I seem to have found a very fast way to find or generate prime numbers. No, this will not help you with your payroll but I find it fascinating that the TRS-80 can find all the primes between 1 and 1,000 in only 23 seconds. This task could take days or weeks using a pencil and paper.

This program is short but fairly involved so I will do my best to explain the logic of the program. This is the program. I call it PRIME/BAS.

```
10 DEFINT A-Z
20 INPUT "ENTER THE NUMBER TO STOP SEARCHING FOR
   PRIMES"; Q
30 IF MEH<Q*2+200 THEN PRINT "TOO HIGH. TRY AGAIN"
   : GOTO 20
40 DIM A(Q)
50 FOR X=2 TO Q/2
60 IF A(X)=1 THEN GOTO 110
70 PRINT USING "##,### "; X;
80 FOR Y=2*X TO Q STEP X
90 A(Y)=1
100 NEXT Y
110 NEXT X
120 FOR X=Q/2+1 TO Q
130 IF A(X)=0 THEN PRINT USING "##,### "; X;
140 NEXT X
```

The basic principal in this program is to start with 2 and find all the numbers with 2 as a factor and set that element of the array equal to 1 which means it is not a prime. It then moves on to three and does the same thing. Since 4 is a factor of 2 it would be a waste of time to find all the numbers with 4 as a factor since all of them would have already been ruled out by the number 2.

The FOR X-NEXT X loop contains the number in which the program searches for all numbers which have it as a factor and only goes half way to the ending number since no number more than half way could be a factor of any number between 1 and the ending number.

Line 60 checks to see if the number X has any other number as a factor, and if it does, it skips to LINE 110 which gets a new number.

If the number doesn't have any factors, then it is a prime and is printed out using an 8 space format so the numbers will be printed in nice, neat columns.

The Y loop is next and goes from two times the number X (the next number with X as a factor) to the ending number and counts by the number X to find all the numbers with X as a factor. When the X loop is done all the the primes between 1 and the ending numbers have been found and all the primes between 1 and half way to the ending number have been printed so the only thing left is to print out the rest.

I then simply set up a FOR NEXT loop between 1 more than half way and the end which prints all the elements that are equal to zero (have no factors).

Actually this program finds all the numbers that AREN'T primes and prints all the others.

```
How Many Files?
Memory Size?
TRS-80 Model III Disk BASIC Rev 1.3
(c) (p) 1988 by Tandy Corp. All Rights Reserved
Created 5-Jul-88
38,282 Free Bytes 3 Files
READY
)RUN "PRIMEEU
ENTER THE NUMBER TO STOP SEARCHING FOR PRIMES? 100
 2   3   5   7   11  13  17  19
23  29  31  37  41  43  47  53
59  61  67  71  73  79  83  89
57
READY
)
```

Arithmetic Practice

Peter Fried
17 Hearthstone Circle
Scarsdale, NY 10583

I designed the following little Model III Disk BASIC program for my son Kenny. It allows him to work Addition, Subtraction, Multiplication and Division problems at random using integers between -10 and +10.

The program also demonstrates the use of the Video scroll control, cursor character control, as well as the <ENTER> key control (PEEK 14400). The score is constantly displayed on top of the screen.

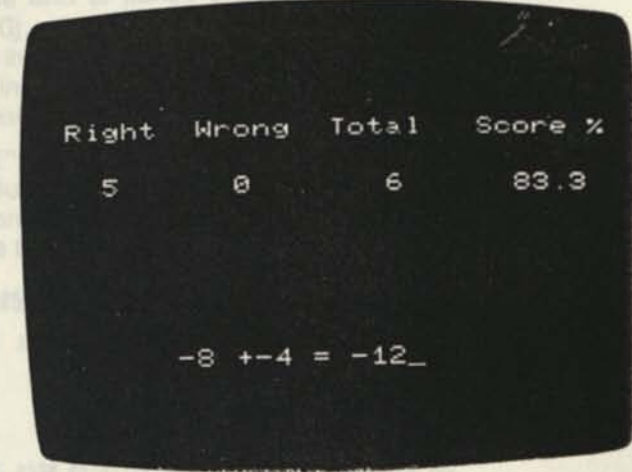
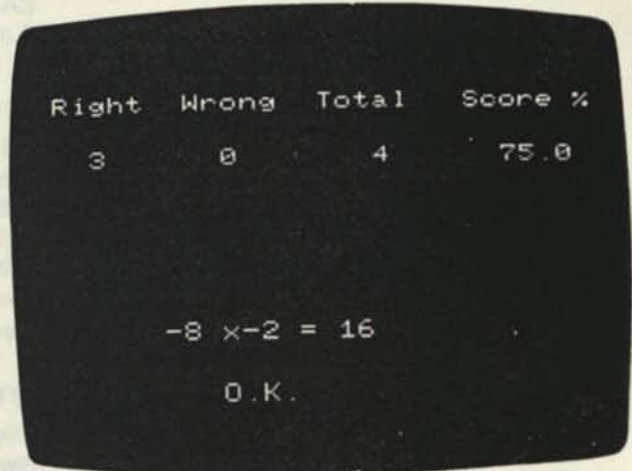
Note: the program will run in a non-disk Model III by changing the LINE INPUT statement in line 1020 to INPUT.

```

2 REM PROGRAM NAME : PRACTICE
5 CLS
  : PRINT @ 261, "This program generates Addition,
  Subtraction,"
  : PRINT " Multiplication and Division
  problems at Random."
  : PRINT
  : PRINT TAB(6) "Hit "; CHR$(34); "ENTER";
  CHR$(34); " to Start. ";
6 PRINT TAB(16) "Enter "; CHR$(34); "9999"; CHR$(34);
  " to quit."
  : PRINT @ 963, "Program by Peter Fried .....
  February 10, 1982";
10 IF PEEK(14400)-1 THEN 10 ELSE CLS
  : PRINT CHR$(23)
  : RANDOM
15 POKE 16419, 95
  : POKE 16916, 3
  : PRINT @ 6, "Right Wrong Total Score %"
20 D=0
  : C=RND(4)
22 A=RND(20)-10
  : B=RND(20)-10
  : PRINT @ 512,;
25 IF A=B THEN 20 ELSE IF B=0 THEN 20 ELSE IF
  A/B<>INT(A/B) THEN 22
30 T=T+1
  : ON C GOTO 100, 200, 300, 400
100 REM ..... ADDITION .....
110 GOSUB 1010
  : PRINT @ 530, A; "+"; B; "= ";
  : GOSUB 1020
  : IF X=A+B THEN 1000
125 D=D+1
  : IF D=1 THEN W=W+1
130 GOTO 110
200 REM ..... SUBTRACTION .....
210 GOSUB 1010
  : PRINT @ 530, A; "-"; B; "= ";
  : GOSUB 1020
  : IF X=A-B THEN 1000
225 D=D+1
  : IF D=1 THEN W=W+1
230 GOTO 210
300 REM ..... MULTIPLICATION .....
310 GOSUB 1010
  : PRINT @ 530, A; "x"; B; "= ";
  : GOSUB 1020
  : IF X=A*B THEN 1000
325 D=D+1
  : IF D=1 THEN W=W+1
330 GOTO 310
400 REM ..... DIVISION .....
410 GOSUB 1010
  : PRINT @ 530, A; "/"; B; "= ";
  : GOSUB 1020
  : IF X=A/B THEN 1000
425 D=D+1
  : IF D=1 THEN W=W+1
430 GOTO 410
1000 PRINT
  : PRINT TAB(12) "O.K."
  : IF D=0 THEN R=R+1
1005 FOR I=1 TO 200
  : NEXT
  
```

```

: PRINT @ 136, USING "## ## ##"
###.# "; R. W. T. R*100/T;
: PRINT @ 576,;
: PRINT
: GOTO 20
1010 PRINT @ 530, CHR$(220);
: RETURN
1020 LINE INPUT X$
  : IF X$="9999" THEN POKE 16916, 0
  : CLS
  : END
1025 X=VAL(X$)
  : RETURN
  
```



Head Injury

Southern New Jersey Head Injury Support Group
1316 Beaverbrook Drive
Cherry Hill, NJ 08034

The above group is seeking information on computer programs relating to head injuries or listings thereof. If individual or association has such information please contact the Southern New Jersey Head Injury Support Group at address shown.

Model I/III Bugs, Errors, and Fixes

NOTE TO USERS:

The following program changes and corrections are provided for your information. If you have an applications program which is working correctly, you should probably NOT make any changes to it. If you feel that the changes should be made, but you do not feel qualified to make the changes yourself, please contact your local Radio Shack Computer Center or Expanded Computer Department for assistance. If you do not have access to one of these stores, then you may want to call Computer Customer Services in Fort Worth for assistance.

CHANGES TO BASIC PROGRAMS

There are general procedures that need to be followed when any corrections are made.

1. Make a backup of the tape or disk that contains the program to be corrected. Changes should be made on the backup copy.
2. Load the program to be changed by typing CLOAD "filename" (tape) or LOAD "filename" (disk) where filename is the name of the program to be modified.
3. Make the line changes indicated in the fix. For existing line numbers, edit or retype the line to match the one in the fix. Enter new lines.
4. Save the corrected program (the one now in memory). Type CSAVE "filename" <ENTER> (tape) or SAVE "filename" <ENTER> (disk) where filename is the name of the program that has been modified.
5. Now make a backup of the corrected tape or diskette.

PATCHES

PATCHes are entered from TRSDOS READY and are used to make corrections to files stored on the disk.

1. Before making a PATCH, back up the diskette that requires modification and make the PATCHES to the backup copy of the diskette.
2. Apply PATCHES according to the information given in your TRSDOS manual.

CASSETTE PORTFOLIO (26-1506)

The sell calculations in the option program are reversed. To correct the problem CLOAD the Option tape and make the following changes. (Model III Version 1.0)

```
Line: 340 FOR I=1 TO NO:GOSUB 800: NEXT
```

Add these lines:

```
800 CV#=05*(I)*O(I,2)*100:REM CURRENT VALUE
810 OV#=O(I,0)*O(I,2)*100:REM OPTION VALUE
820 OC#=04*(I):REM OPTION COST
830 IFO3$(I)="B" THEN GL# = CV# - OV# - OC# ELSE GL# =
OV# + OC# - CV#
840 O6*(I) = GL#: RETURN
```

CSAVE to save the changes.

STOCKPAK (26-1507)

Below are two sets of program changes to correct two separate problems with Stockpak.

1—Erratic performance, including syntax errors, reboot, etc., is exhibited when printing reports. Make the following corrections to the program PTREPORT. (Model III Version 3.0)

```
Line: 350 IFPEEK(16425)>50 THEN LPRINT CHR$(12);:POKE 16425,1:
QZ=QZ+1 ELSE RETURN
```

ADD line 7:

```
7 POKE 16412,255
```

SAVE "PTREPORT" to store the program changes.

Additional corrections need to be made to the program REPRUN/BAS.

Add the following lines.

```
Line: 5 POKE 16412,255
```

```
Line: 1503 POKE 16425,1
```

SAVE "REPRUN/BAS" to store the program changes.

2—The Eligibility for Long Term Gains Report is inaccurate when stocks with no commission are included. (Model III Version 3.0)

Make the following change to the program PTREPORT.

```
Line: 2520 FOR I=X8 TOK 7: FL=0: CH#=0: FOR I1=1 TO 3: FOR I2=0 TO 9
```

SAVE "PTREPORT" to store the program changes.

ACCOUNTS PAYABLE (26-1554)

Following are four sets of changes to correct separate problems with Accounts Payable.

1—Whether the end of period is performed or not, the number of invoices selected and posted is inaccurate after loading the end of period processing program. (Model I Version 3.0)

Make the following correction to the program PROCESS.

Add line 124:

```
124 LSETU1$(1)=MKI$(1):LSETU1$(5)=MKI$(15)
```

SAVE "PROCESS" to store the program change.

2—During End-of-Period Processing Error Code 5 in line 45 can occur. (Models I/III Version 3.0 and Prior)

Make the following changes to the program PROCESS.

VERSIONS PRIOR TO 3.0

```
Line: 350 W#=ABS(N#)*100+.1:X=W#/D1#:W#-W#-X*D1#
:V$=CHR$(X-(N#<0)*128):X=W#/D2#:W#-W#-X*D2#
:W#-W#-X*D2#:V$=V$+CHR$(X):X=W#/D3#
:W#-W#-X*D3#:V$=V$+CHR$(X)+CHR$(W#)
: RETURN
```

```
Line: 2530 PRINT:PRINTCHR$(30);" UNEXPECTED ERROR CODE
":ERR/2+1;" IN LINE";ERL:GOSUB 490:END
```

SAVE "PROCESS" to store the program changes.

Make the following changes to the program CHECKS.

```
Line: 350 W#=ABS(N#)*100+.1:X=W#/D1#:W#-W#-X*D1#
:V$=CHR$(X-(N#<0)*128):X=W#/D2#:W#-W#-X*D2#
:V$=V$+CHR$(X):X=W#/D3#:W#-W#-X*D3#
:V$=V$+CHR$(X)+CHR$(W#): RETURN
```

```
Line: 2530 PRINT:PRINTCHR$(30);" UNEXPECTED ERROR CODE ";
ERR/2+1;" IN LINE";ERL:GOSUB 490:END
```

SAVE "CHECKS" to store the program changes.

Make the following additional changes to the program POST.

```
Line: 350 W#=ABS(N#)*100+.1:X=W#/D1#:W#-W#-X*D1#
:V$=CHR$(X-(N#<0)*128):X=W#/D2#:W#-W#-X*D2#
:V$=V$+CHR$(X):X=W#/D3#:W#-W#-X*D3#
:V$=V$+CHR$(X)+CHR$(W#): RETURN
```

```
Line: 2530 PRINT:PRINTCHR$(30);" UNEXPECTED ERROR CODE ";
ERR/2+1;" IN LINE";ERL:GOSUB 490:END
```

SAVE "POST" to store the program changes.
Make the following changes to the program INVOICES.

```
Line: 350 N# = VAL(IN$): W# = ABS(N#) * 100 + .1: X = W# / D1#
      : W# = W# - X * D1#: V$ = CHR$(X - (N# < 0) * 128): X = W# / D2#
      : W# = W# - X * D2#: V$ = V$ + CHR$(X): X = W# / D3#
      : W# = W# - X * D3#: V$ = V$ + CHR$(X) + CHR$(W#): RETURN
```

```
Line: 2530 PRINT: PRINT CHR$(30); " UNEXPECTED ERROR CODE ";
      ERR / 2 + 1; " IN LINE"; ERL: GOSUB 490: END
```

SAVE "INVOICES" to store the program changes.

VERSION 3.0

Make the following changes to the program PROCESS.

```
Line: 45 W# = ABS(N#) * 100 + .1: X = W# / D1#: W# = W# - X * D1#
      : V$ = CHR$(X - (N# < 0) * 128): X = W# / D2#: W# = W# - X * D2#
      : V$ = V$ + CHR$(X): X = W# / D3#: W# = W# - X * D3#
      : V$ = V$ + CHR$(X) + CHR$(W#): RETURN
```

```
Line: 151 PRINT: PRINT CHR$(30); " UNEXPECTED ERROR CODE ";
      ERR / 2 + 1; " IN LINE"; ERL: GOSUB 65: END
```

SAVE "PROCESS" to store the program changes.
Make the following changes to the program CHECKS.

```
Line: 89 W# = ABS(N#) * 100 + .1: X = W# / D1#: W# = W# - X * D1#
      : V$ = CHR$(X - (N# < 0) * 128): X = W# / D2#: W# = W# - X * D2#
      : V$ = V$ + CHR$(X): X = W# / D3#: W# = W# - X * D3#
      : V$ = V$ + CHR$(X) + CHR$(W#): RETURN
```

```
Line: 315 PRINT: PRINT CHR$(30); " UNEXPECTED ERROR CODE ";
      ERR / 2 + 1; " IN LINE"; ERL: GOSUB 141: END
```

SAVE "CHECKS" to store the program changes.
Make the following corrections to the program POST.

```
Line: 59 W# = ABS(N#) * 100 + .1: X = W# / D1#: W# = W# - X * D1#
      : V$ = CHR$(X - (N# < 0) * 128): X = W# / D2#: W# = W# - X * D2#
      : V$ = V$ + CHR$(X): X = W# / D3#: W# = W# - X * D3#
      : V$ = V$ + CHR$(X) + CHR$(W#): RETURN
```

```
Line: 195 PRINT: PRINT CHR$(30); " UNEXPECTED ERROR CODE ";
      ERR / 2 + 1; " IN LINE"; ERL: GOSUB 107: END
```

SAVE "POST" to store the program changes.
Make the following changes to the program INVOICES.

```
Line: 79 N# = VAL(IN$): W# = ABS(N#) * 100 + .1: X = W# / D1#
      : W# = W# - X * D1#: V$ = CHR$(X - (N# < 0) * 128): X = W# / D2#
      : W# = W# - X * D2#: V$ = V$ + CHR$(X): X = W# / D3#: W# =
      : W# - X * D3#: V$ = V$ + CHR$(X) + CHR$(W#): RETURN
```

```
Line: 305 PRINT: PRINT CHR$(30); " UNEXPECTED ERROR CODE ";
      ERR / 2 + 1; " IN LINE"; ERL: GOSUB 133: END
```

SAVE "INVOICES" to store the program changes.
3—The General Ledger code numbers cannot be modified in SETUP. (Models I/III Version 3.1)

Make the following change to the program SETUP.

```
Line: 43 GOSUB 89: PRINT@470, "ALL CHANGES STORED "
      : ONRF + 1 GOSUB 199: GOTO 207
```

SAVE "SETUP" to store the program changes.
4—Adjustments are not handled correctly in Select/Hold by vendor. Make the following correction to the program APS. (Model III Version 3.1)

```
Line: 299 GOSUB 263: GET I, KR: PRINT@615, L2$, I1$; :
      IA = (IV$ = LEFT$(I1$, LEN(IV$))) AND I1$ <>
      "ADJUST" OR (IA AND I1$ = "ADJUST"):
      IF IATHENGOSUB 313
```

SAVE "APS" to store the change in the program.

ACCOUNTS RECEIVABLE (26-1555)

Following are four sets of program changes to correct separate problems in Accounts Receivable.

Correction 1—If any of the following symptoms occur, the Accounts Receivable index has been crashed, and the following Recovery program should be run. These correc-

tions replace those previously published in March 1982 Model I Version 3.0)

Unexpected Error Code 64, Line 2850 ("ARS")

Unexpected Error Code 64, Line 1430 ("ARS")

Bad Record Number on program line with "GET" or "PUT" statement

When sorting, the statement "duplicate account # 0 exists" appears or transactions appear under the wrong account - i.e. some transactions for account 3 are on account 20's bill, etc.

In order to correct these above errors please run the following program written to reconstruct "CUSINDEX" and "TRANSACTION" files of Model I Accounts Receivable program.

Quicker routines can be written, but this was written with the customer in mind. This routine will not recapture report numbers nor session numbers. It is merely intended to aid in recapturing data files rendered inaccessible because the Customer Index or Transaction files have been damaged or scrambled.

***** This is a last resort. All transactions must be checked afterwards and verified. This program will pick up transactions on the disk, including deleted transactions and previous period transactions not yet over-written in the current period.

Enter the following program in BASIC.

```
1 ' *** RECOVER *** (C) 1981 TANDY CORP
2 ' *** RECOVERS POINTERS/INDEX FOR ACCTS.
   RECEIVABLE
3 ' PROGRAM DISK SHOULD BE IN DRIVE #, DATA DISK IN
   DRIVE 1 --- ON 3 DRIVE SYSTEMS, SECOND DATA DISK
   IN DRIVE 2.
4 ' RECOVER WILL RUN SORT AND RETURN TO ARS ON
   COMPLETION.
5 ' **** FOR VERSION PRIOR TO 3.0, LINE 210 MUST BE
   MODIFIED: CHANGE "JD*128ASD$" TO "JD*127ASD$" AND
   "SASDES$" TO "4ASDES$"
10 CLEAR 500: DEFINT A-Z: CLS: INPUT "PASSWORD"; P$: INPUT "
   OR 3 DRIVE SYSTEM"; S
20 GOSUB 220: OPEN "R", I, PT$: NT = LOF(I) * 8: OPEN "R", 2,
   PD$: NA = LOF(2) * 2
30 DIM ITR(NT, 2), AC(NA, 2): CLS: PRINT@154, "FIRST
   PASS": PRINT@402, "WORKING ON TRANSACTION # "
40 KK = NT: FORK = 1: TONT: GOSUB 180: PRINT@426, K: TR(K, 0) =
   CVI(V1$): IF V0$ < "A" OR V0$ > "Z" THEN KK = K: NT
50 NEXT K: PRINT@154, "SECOND PASS": PRINT@426, "
   ": UN = KK: UT = KK - 1: U = UT:
   FORK = 1: TOUT: IF ITR(K, 0) < 0 THEN ITR(K, 0) = -TR(K, 0): U = U - 1
60 PRINT@426, K: NEXT K: CLS: PRINT@154, "FIRST
   PASS": PRINT@402, "WORKING ON ACCOUNT # "
70 FOR J = 1: TONA: GOSUB 200: PRINT@424, J: AC(J, 0) = CVI(DD$)
   : IF AC(J, 0) < 1 THEN F = F + 1 ELSE CV# = CV# + CVD(DB$)
80 NEXT J: TN = J - 1: TI = TN - F: JJ = J - 1: PRINT@154, "SECOND
   PASS": PRINT@424, "
   ": UX = UT + (TR(UT, 0) = 1)
90 FOR J = 1: TJJ: PRINT@424, J: PT = 0: FT = 0: AC = AC(J, 0)
   : FORK = 1: Toux: IF ITR(K, 0) = AC(IHENTR(PT, 2)) = K
   : TR(K, 1) = PT: PT = K: IFFI = 0 THEN FT = K
100 NEXT K: AC(J, 1) = FT: AC(J, 2) = PT: NEXT J: CLS
   : PRINT@402, "RESETTING TRANSACTION POINTERS"
110 FORK = 1: TOUT: GOSUB 180: LSET V8$ = MKIS(TR(K, 1))
   : LSET V9$ = MKIS(TR(K, 2)): PUT I, KR: NEXT K: CLOSE 1
120 CLS: PRINT@404, "RESETTING ACCOUNT POINTERS"
130 FOR J = 1: TONT: GOSUB 200: LSET DI$ = MKIS(AC(J, 1))
   : LSET DJ$ = MKIS(AC(J, 2)): PUT 2, JR: NEXT K: CLOSE 2
140 CLS: PRINT@410, "WRITING INDEX": OPEN "O", I,
   "CUSINDEX." + P$ + ".1": PRINT@1, TI: TN: F: U: UT
   : UN: 0: 1: 1: 1: 1: 1: 1: 1: STR$(CV#) + "D0"
150 FORN = 1: TONT - 1: CN = 1: IF AC(N, 0) < 1 THEN CN = -CN
160 PRINT@1, CN: CN: CN: NEXT: CLOSE 1: OPEN "O", I,
   "TRANSFER.1": PRINT@1, P1$: PRINT@1, PD$
   : PRINT@1, P5$: PRINT@1, P1$: PRINT@1, P5$
   : CLOSE
170 CLS: PRINT@410, "SORTING DATA": CLEAR 500: RUN "ARSORT"
180 KR = INT((K - 1) / 8) + 1: KD = K - 8 * INT((K - 1) / 8) - 1
190 FIELD I, KD * 31 AS VV$, 1 AS V0$, 23 AS V1$, 2 AS V8$, 2 AS V9$
   , 3 AS V$: GET I, KR: RETURN
200 JR = INT((J - 1) / 2) + 1: JD = J - 2 * INT((J - 1) / 2) - 1
210 FIELD J, JD * 128 AS D$, 10 AS D0$, 8 AS D8$, 4 AS DCS$,
   2 AS DDD$, 5 AS DES$, 2 AS DI$, 2 AS DJ$: GET 2, JR: RETURN
```

```

220 PIS="TRANSACT."*P$*":CHR$(47+S):P$=P$+
:1":PIS="CUSINDEX."*P$:P$="CUSDATA."*P$
:P$="CUSSETUP."*P$:P$="CLFILE."*P$
:RETURN

```

Type SAVE "RECOVERY" to save the above program for later use. Type RUN and press (ENTER) to start the recovery process. If the program is ever needed again load it from the disk with LOAD "RECOVERY"

Correction 2—When a correction entry is made to a payment, the correction entry replaces the previous payment on the statement. (Models I/III Versions 3.0 and Prior)

Make the following correction to line 2020 in version 3.0 of the program ARS. In versions prior to 3.0, this is line 1650.

```

Line: 2020 PR#-PR#-VE#:CB#-CB#-VE#:CV#-CV#-VE##
:G#(VJ)=G#(VJ)-VE#:G#(2)=G#(2)-VE# )

```

SAVE "ARS" to store the changes in the program.

Correction 3—These changes correct the following problems in versions prior to and including 3.1.:

Only one account can be deleted before posting. Accounts with transactions cannot be deleted immediately after posting.

The changes are made to the program ARS.

VERSIONS PRIOR TO 3.0

```

Line: 1300 IFDI<>DANDUD<>DTHENPRINT@832,CHR$(31);"CANNOT
DELETE THIS ACCOUNT UNTIL AFTER
POSTING":GOSUB3340:GOTO1210ELSE IFPT=1THEN
GOSUB3140

```

```

Line: 1990 U=#:UD=#:LPRINTLF$;LPRINT"END OF
REPORT":LPRINTLL$;LPRINTCHR$(12):CLOSE3:GOTO1630

```

```

Line: 3410 FOR N=1 TO TN-1:INPUT#1,P(N,0),P(N,1),P(N,2)
:NEXT:CLOSE1:UD=U

```

VERSION 3.0

```

Line: 1060 IFDI<>DANDUD<>DTHENPRINT@832,CHR$(31);"CANNOT
DELETE THIS ACCOUNT UNTIL AFTER POSTING"
:GOSUB2600:GOTO960ELSE IFPT=1THENGOSUB2450

```

```

Line: 2640 OPEN"1",1,PIS:INPUT#1,T1,TN,F,U,UT,UN,EP,R0,
R1,R2,R3,R4,R5,R6,R7,CV#:UD=U

```

```

Line: 1630 U=#:UD=#:LPRINT" ":LPRINT"END OF REPORT"
:LPRINTLL$:GOSUB2515:CLOSE3:GOTO1360

```

VERSION 3.1

```

Line: 1630 U=#:UD=#:LPRINT" ":LPRINT"END OF REPORT"
:LPRINTLL$:GOSUB2515:CLOSE3:GOTO1360

```

SAVE "ARS" to store the program changes.

Correction 4—Transactions cannot be entered or edited after statements are approved. (Versions 3.1 and prior)

Make the following corrections to the program ARS.

VERSIONS 3.0 & 3.1

```

Line: 1310 CLS:IFEP<3THENPRINT@474,"TRANSACTIONS":GOSUB240:
RUN"ENTRAN"ELSEPRINT@462,"END OF PERIOD
PROCESSING INCOMPLETE":GOSUB2600:GOTO230

```

VERSIONS PRIOR TO 3.0

```

Line: 1580 CLS:IFEP<3THENPRINT@474,"TRANSACTIONS":GOSUB340
:CLEAR50:RUN"ENTRAN":END:ELSEPRINT@462,"END OF
PERIOD PROCESSING INCOMPLETE":GOSUB3340:GOTO330

```

Type SAVE "ARS" (ENTER) to store the changes in the program.

DISK PAYROLL (26-1556)

In the following material is information regarding the way that Payroll calculates FICA and two corrections to existing program problems.

1—The way that Disk Payroll calculates FICA is correct. The Federal Government says FICA should be calculated by the following formula: GROSS *.0613. (Model I Non EIC Version)

Consider this Table:

Employee 1-Pay x .0613 equals Subtotal	555.53 x .0613 =	34.05
Employee 2-Pay x .0613 equals Subtotal	449.50 x .0613 =	27.55
Employee 3-Pay x .0613 equals Subtotal	99.63 x .0613 =	6.11
Employee 4-Pay x .0613 equals Subtotal	444.56 x .0613 =	27.25
Employee 5-Pay x .0613 equals Subtotal	999.50 x .0613 =	61.27

Gross Payroll Times .0613 equals Total or 2548.72 x .0613 = 156.24, BUT the Total of Each Employees FICA does not equal 156.24. It equals 156.23.

There are two ways of calculating FICA. One is to take a total sum of the gross wages of the employee and multiply it times .0613 or multiply the GROSS of each paycheck times .0613, then add those totals up to come up with the Tax. These two totals probably will not be equal due to the rounding errors involved in calculating the tax two different ways. When the employer fills out his forms, FICA is not calculated by multiplying 2549.18 x .0613 but by multiplying 2549.18 x .1226 to get the amount that he owes to the government.

2—If the weeks worked, workmen's comp, or net pay in the employee's personnel information section cannot be changed, make the following change to the program PR4ADD. (Model I Versions Prior to 2.0)

```

Line: 3900 FOR I=21TO22:MID$(G$,I*8-7,8)=MKD$(E#(1-23))
:NEXT I:MID$(G$,22*8-7,8)=MKD$(E#(17))
:LSETGG$=G$:LSETFF$=N$:PUT3,N:PUT4,1+2*(N-1)
+INT((2-2)/2):GOTO810

```

SAVE "PR4ADD" to store the program changes.

3—Two new steps have been added to the calculation process for Minnesota State Tax Tables. The additions involve a 7% surcharge. The necessary program changes can be implemented by making the following changes to the program STATETAX. (Models I/III Version 2.0)

```

Line: 7900 PRINT#1,"5900E#(1)=(E#(1)-CVS(MID$(N$,111,4)))
*1.084)/K#:IFE#(1)<.01THENE#(1)=0:GOTO6500ELSE
6500":CLOSE:GOTO2000

```

ADD the following line:

```

Line: 3160 PRINT#1,"5115G#-G#*.909"

```

SAVE "STATETAX" to store the changes in the program. Then RUN "STATETAX" and reenter the tax tables using the instructions in the manual.

MEDICAL OFFICE SYSTEM (26-1568)

Following are two sets of corrections for this program:

Correction 1—When option 2 of statement printing is selected, zero balance accounts will not be printed. Make the following corrections to the program BLLPRT. (Model I/III Version 1.0)

```

Line: 1100 PRINT@728,"SELECT OPTION: ";:FL=-1
:GOSUB210:Q=VAL(IN$):PRINT:IFQ<1ORQ>3
THEN1100ELSEPF=Q-1:X0=Q<>2:IFQ=1THEN
POKE16424,43ELSEPOKE16424,67

```

```

Line: 1130 IFA=AHTHEN1140ELSE IFX0AND(BA=0ORIS="N")
THEN1210ELSEGOSUB1215:IFPTHENLPRINTLF$
:LPRINTLF$:LPRINTLF$:IFBA<0THENLPRINT
TAB(40)"CREDIT BALANCE"ELSELPRINTTAB(42)
"PLEASE REMIT"

```

```

Line: 1140 IFX0AND(BA=0ORIS="N")THENGOSUB1250:
GOTO1210ELSEGOSUB1215

```

SAVE "BLLPRT" to store the program changes.

Correction 2—Medical Office System contains a bug which may cause problems if more than three patient data files are used.

**Changes to MOS are somewhat critical, the following process should be followed closely.

1. In BASIC, LOAD "ALPHASOR".
2. Type these lines as shown.

```
Line: 124# FZ=INT((AC-1)/Z1):IFFZ<>FOTHENFO=FZ:CLOSE3:
      FMS="PDATA"+CHR$(48+FO)+":3":NU=3:GOSUB400:CLS
```

```
125# LR=AC-FO*Z1:PR=INT((LR-1)/2)+1:SR=LR-2*
      INT((LR-1)/2)-1
```

3. SAVE "ALPHASOR"
 4. In BASIC, LOAD "MENU".
- Add these lines as shown.

```
12# FZ=INT((AC-1)/Z1):IFFZ<>FOTHENFO=FZ:CLOSE3:
      FMS="PDATA"+CHR$(48+FO)+":3":NU=3:GOSUB400
14# LR=AC-FO*Z1:PR=INT((LR-1)/2)+1:SR=LR-2*
      INT((LR-1)/2)-1
140# DEFFNA(X)=PEEK(X)+PEEK(X+1)*256:E=FNA(&H40A4)
141# S=FNA(E):IFFNA(S+2)<1000THENE=S:GOTO1410
142# PRINT"TP=TP+S-FNA(&H40A4)"IN LINE 1110":END
```

5. TYPE: GOTO1400

(The computer will pause and print 'TP = TP + NNNN in line 1110. The NNNN will be a number.)

6. Type LIST 1110 <ENTER>

(If the number after TP + TP + ... does not match the computer generated number of step 5, the line must be edited to contain the new value.)

7. Type EDIT 1170 <ENTER>

Line 1170 should look like this:

```
117# DATAMONSORT,PRINT STATEMENTS,BLIPRT,SET UP
      SYSTEM DISKS,INIT,16922,16923,16924,1390,4170,3,
      16452,16453,16454,660,3960,6
```

(If the numbers that appear after 16924 are not 1390,4170, edit them to conform to the new line 1170. There are no other changes in the line.)

8. Type SAVE "MENU" <ENTER>

**FORMAT A NEW DATA DISK FOR EACH PATIENT DATA DISK.

Type in this PROGRAM:

```
1# CLEAR2000:CLS:AC=1:FO=-1:B2=-1
2# IFPEEK(293)=73THENZ1=1390ELSEZ1=660
3# D$(1)="OLD":D$(2)="NEW"
4# CLS:INPUT"WHAT IS YOUR PASSWORD? (PRESS <ENTER> IF
      NONE)";PWS
5# IFPWS=""THENPWS="PASSWORD"
6# PWS=""
7# IFFO<>INT(AC/(Z1+1))THENCLOSE1:FO=
      INT(AC/(Z1+1)):D=1:DN=FO+1:GOSUB140:OPEN"R",1,
      "PDATA"+CHR$(FO+48)+PWS+":1"
8# PRINT339,AC
9# LR=AC-FO*Z1+1:PR=INT((LR-1)/2)+1:SR=LR-2*
      INT((LR-1)/2)-1
10# FIELD1,SR*127ASA1$,127ASAS:GET1,PR
11# B1=INT((AC-1)/Z1):IFB1<>B2THENB2=B1:CLOSE2
      :FMS="PDATA"+CHR$(48+B1)+":2":D=2:DN=B2+1:GOSUB140
      :OPEN"R",2,"PDATA"+CHR$(48+B1)+PWS+":2"
12# B=AC-B1*Z1:B3=INT((B-1)/2)+1:B4=B-2*
      INT((B-1)/2)-1
13# FIELD2,B4*127ASB1$,127ASB$:LSETB$=A$
      :PUT2,B3:AC=AC+1:GOTO70
14# IFAC>1THENM$="NEXT"
15# CLS:PRINT"PUT "D$(D)" PATIENT DATA DISK"DN"IN
      DRIVE"D
16# PRINT:PRINT"PRESS <ENTER> WHEN READY, OR <@> TO
      END"
17# W$=INKEY$:IFW$=""THEN170
18# IFW$=CHR$(13)THENCLS:PRINT@320,"WORKING ON RECORD"
      #":RETURN
19# IFW$=""THENCLOSE:ENDELSE170
```

FOLLOW THE PROMPTS!

Insert old and new disks as requested. These new disks will be used with the modified MOS disk. DO NOT attempt to use the old patient data disks with the modified program.

REAL ESTATE VOL. III (26-1573)

If the "End of Period Payment" is used, the first factor the factor column on the hardcopy option is deleted and factors are moved up by one. The screen display is all right (Models I/III Version 3.0)

Correct the problem by CLOADing the Variable Income Analysis tape and making the following change.

```
Line: 607# IFJ=1THENB=(1/(1+(B/N1)))^(1-1):GOTO6090
```

CSAVE "V" to save the changes on a new cassette.

NOTE On the Model III the "up-arrow" key appears as a left bracket on the screen, but it functions as exponentiation symbol in the program.

RUNCOBOL

COBOL (26-2203 Model III Version 1.3b)

COBOL RUNTIME (26-2206 Model I Version 1.3b)

COBOL RUNTIME (26-2207 Model III Version 1.3b)

The following problem exists in RUNCOBOL of all three of the above programs.

The OPEN EXTEND OPTION is not functional in the RUNCOBOL runtime program. An ERROR 30 message results when OPEN EXTEND is used with a file that has been previously opened and closed.

Below is a program which when run will result in error 30.

```
00010# IDENTIFICATION DIVISION.
00011# PROGRAM-ID.
00012# ENVIRONMENT DIVISION.
00013# CONFIGURATION SECTION.
00014# SOURCE-COMPUTER.
00015# OBJECT-COMPUTER.
00016# INPUT-OUTPUT SECTION.
00017# FILE-CONTROL.
00018# SELECT I-O-FILE
00019# ASSIGN TO RANDOM, "ERROR30/DAT";
000191# ORGANIZATION IS SEQUENTIAL;
000192# ACCESS IS SEQUENTIAL.
00020# DATA DIVISION.
00021# FILE SECTION.
00022# FD I-O-FILE
00023# RECORD CONTAINS 32 CHARACTERS
00024# DATA RECORD IS I-O-RECORD.
00025# 01 I-O-RECORD.
00026# 02 DATA-RECORD
00027# PROCEDURE DIVISION.
00028# BEGIN.
00029# DISPLAY "FIRST WRITE",
00030# LINE 01, POSITION 20, ERASE.
00031# OPEN OUTPUT I-O-FILE.
00032# MOVE SPACES TO DATA-RECORD
00033# WRITE I-O-RECORD.
00034# CLOSE I-O-FILE.
00035#
000351# DISPLAY "SECOND WRITE",
000352# LINE 02, POSITION 20, ERASE.
00036# OPEN EXTEND I-O-FILE.
00037# MOVE SPACES TO DATA-RECORD.
00038# WRITE I-O-RECORD.
00039# CLOSE I-O-FILE.
00040#
00041# STOP RUN.
EOF
```

Applying the following patches at TRSDOS READY correct this problem for all of the above packages.

```
PATCH RUNCOBOL/CMD (ADD=9A2A,FIND=000000000000,
      CHG=CDF0AACDFDCB
PATCH RUNCOBOL/CMD (ADD=9A30,FIND=000000000000,
      CHG=0266C8FDE5E1
PATCH RUNCOBOL/CMD (ADD=9A36,FIND=000000000000,
      CHG=111C0019EBDD
PATCH RUNCOBOL/CMD (ADD=9A3C,FIND=000000000000,
      CHG=E5FDE5CD
PATCH RUNCOBOL/CMD (ADD=9A40,FIND=000000000000,
      CHG=4844FDE1DDE1
```

```

PATCH RUNCOBOL/CMD (ADD=9A46,FIND=000000000000,
                    CHG=C9FDCB0266CA
PATCH RUNCOBOL/CMD (ADD=9A4C,FIND=0000000000,
                    CHG=569AE5D5
PATCH RUNCOBOL/CMD (ADD=9A50,FIND=000000000000,
                    CHG=CDF7ADD1E1C8
PATCH RUNCOBOL/CMD (ADD=9A56,FIND=0000000000,
                    CHG=CDEBADC9
PATCH RUNCOBOL/CMD (ADD=A43D,FIND=CDF0AA,
                    CHG=CD2A9A
PATCH RUNCOBOL/CMD (ADD=A7CF,FIND=FD,CHG=C3
PATCH RUNCOBOL/CMD (ADD=A7D0,FIND=CB0266,CHG=DEA700
PATCH RUNCOBOL/CMD (ADD=AB05,FIND=CDEBAD,CHG=CD479A

```

COBOL COMPILER (26-2203)

The following correction was published in the May issue of TRS-80 Microcomputer News but the last two patches were missing, and the change (CHG) string was missing on the third from the last patch. (Model I/III Version 1.3b)

When RUNCOBOL (MOD I/III) attempts to call an independent segment overlay, it will have a LOAD FAIL ERROR message and return to DOS. RUNCOBOL is unable to load and execute the machine language program (independent segment) and is forced to return to TRSDOS.

The following PATCHes will correct the above problem:

```

PATCH RUNCOBOL/CMD (ADD=9DB8,FIND=CD11AEC2419F,
                    CHG=000000000000)
PATCH RUNCOBOL/CMD (ADD=9DC4,FIND=2A,CHG=21)
PATCH RUNCOBOL/CMD (ADD=ADD0D,FIND=19,CHG=09)
PATCH RUNCOBOL/CMD (ADD=ACFE,FIND=D60A,CHG=C318)
PATCH RUNCOBOL/CMD (ADD=AD00,FIND=30,CHG=9A)
PATCH RUNCOBOL/CMD (ADD=9A18,FIND=0000000000,
                    CHG=FE0ADA04)
PATCH RUNCOBOL/CMD (ADD=9A1C,FIND=0000000000,
                    CHG=ADD611DA)
PATCH RUNCOBOL/CMD (ADD=9A20,FIND=0000000000,
                    CHG=11ADFE06)
PATCH RUNCOBOL/CMD (ADD=9A24,FIND=000000000000,
                    CHG=D211ADC302AD)
PATCH RUNCOBOL/CMD (ADD=9B65,FIND=2A69AE4E23,
                    CHG=46ED43D1AF)
PATCH RUNCOBOL/CMD (ADD=9B6A,FIND=46ED43D1AF,
                    CHG=CD8CA8205C)
PATCH RUNCOBOL/CMD (ADD=9DCB,FIND=489F,CHG=D49D)
PATCH RUNCOBOL/CMD (ADD=9B60,FIND=CD8CA82066,
                    CHG=2A69AE4E23)

```

Some Straight Talk About the Radio Shack Network Systems

Radio Shack's Network 2 and Network 3 systems add convenience to classroom computing. The following facts and figures have been prepared to show that Networking with the TRS-80 can be economical, too.

The Network 2 system keeps control of loading programs in the hands of the teacher. Up to sixteen student stations can be loaded with the same program simultaneously, or different programs can be loaded into different stations. Here is a summary of Network 2 costs for a 16-station network with a 48K two-disk host:

Radio Shack TRS-80 Network 2 System Optimum Configuration = 16 Student Stations

Equipment:	
16 16K Model III TRS-80s	\$15,984.00
1 Radio Shack Network 2 Controller	\$499.00

1 TRS-80 48K two-drive disk system	\$2,295.00
	<hr/>
	\$18,778.00
Cost per year for use over a 5-year period (excluding service contract/maintenance)	\$3,755.60
Cost per student station per year	\$234.73
Cost per student hour (1000 hours per station per year)	\$.23

Network 3 gives non-disk student stations most of the capabilities of a disk system. The system lets students sitting at the student stations load and save programs using the host disk. A printer at the host computer can be shared by the student stations. With 32K or 48K RAM in the student stations, many Radio Shack educational software programs will run on the Network 3. (Business programs or programs in compiler languages will NOT run on the Network 3.) The following is a summary of costs for a Network 3 system in which each student station has 48K RAM:

Radio Shack TRS-80 Network 3 System Maximum Configuration = 16 Student Stations

Student Stations:	
16 48K Model III TRS-80 tape systems (includes 16 16K Model IIIs @999.00, 32 16K RAM kits @49.00, installation fees @15.00 per RAM kit, 16 RS-232 boards @99.00 and installation fees @17.50 per board)	\$19,896.00
Network 3 Controller, Operating System, and Cables (typical cable configuration)	\$1,614.00
Host Computer: 48K Model III dual disk system	\$2,295.00
Network 2 Controller (used for simulta- neous host-to-student-station program transfer and loading Network 3 commu- nications software)	\$499.00
Total Network 3 System	<hr/>
	\$24,304.40

Cost per year for use over a 5-year period (excluding service contract/ maintenance)	\$4,860.88
Cost per student station per year	\$303.81
Cost per student hour (1000 hours per station per year)	\$.30

Note that the Network 3 system cost is less than that for 16 Radio Shack disk systems with 48K:

16 48K single-disk student stations (each \$1849.00)	\$29,584.00
Cost per year for use over a 5-year period (excluding service contract/maintenance)	\$5,916.80
Cost per student station per year	\$369.80
Cost per student hour (1000 hours per station per year)	\$.37

News From The Education Division

The Radio Shack Education Division has grown dramatically both in personnel and in the number of different types of programs we're producing, in the past year alone. Because our commitment to the educational user is a source of pride at Radio Shack, we'd like to give you an overview in this educational issue of the exciting things we've been doing to support educational use of the TRS-80 microcomputer.

Although it isn't generally known, many microcomputer manufacturers don't develop courseware for their own machines. Instead, they rely on outside developers. Radio Shack was one of the first manufacturers to see what the microcomputer could do for education and to get involved in courseware development. Students at all grade levels and in all subjects are benefiting from the extra practice, instruction, or exploration available from the computer. With good courseware, the computer is easy to use and non-judgmental. It is allowing students to work at their own pace, free to learn from their mistakes without embarrassment and free to do their own best work without peer pressure. Best of all, the computer has unending patience, rewards students with positive feedback messages, and makes students view problem-solving as a game. From the start of the Radio Shack Education Division two and a half years ago, our purpose has been to develop programs that are constructive in all of these ways.

The first program we released was the Radio Shack K-8 Math Program, Volume One. K-8 Math is a classic drill-and-practice CAI (Computer Assisted Instruction) program. In this case, the program provides varied and intensive skill-building work by randomly generating problems in numeration, addition, subtraction, multiplication, and division for grades K through 8. Field tests prove that K-8 Math works. By motivating, encouraging, and drilling students on math basic skills, K-8 Math has been effective in raising math competency levels. The quality of K-8 Math has become a standard for other Radio Shack courseware programs.

Since K-8 Math, we've expanded our product development efforts. Many more kinds of CAI programs are now being released, including tutorial programs, simulation and problem solving programs, and basic skills practice programs in other areas (like reading and business). Many of our latest products have been featured in recent TRS-80 MICROCOMPUTER NEWS articles. (You'll also find descriptions of many in this month's Computer Customer Service article.) In addition to the area of CAI, our current product development efforts are in these categories: (1) computer literacy, (2) computer managed instruction, (3) authoring systems, (4) resource materials, and (5) special hardware.

COMPUTER LITERACY

Computer literacy is a fairly new subject area in many schools. On one level, computer literacy deals with the history of computers and their place in society—roughly compara-

ble to teaching about the Wright brothers and the history of air and space travel. To teach students about computers is to teach them about something that will influence their lives directly or indirectly. Radio Shack's Basic Computer Literacy package COMPUTERS PAST AND PRESENT is a complete teaching package for this kind of computer literacy course.

On another level, computer literacy deals with the ability to use, and perhaps to program, the computer. This area can emphasize mainly the practical skills, as when word processing is taught as part of a vocational program. Or it can get into programming and the realm of intellectual problem-solving with the computer as laboratory. Radio Shack's COMPUTER EDUCATION SERIES helps meet the need for this kind of computer literacy with three complete teaching packages designed for secondary and post-secondary students. Taken consecutively, these courses lead students and teachers from a first introduction to the computer through advanced programming in the BASIC computer language.

The COMPUTER EDUCATION SERIES is designed so it can be used by teachers who may not know a lot about computers or programming. This addresses a problem seen frequently in the schools: wanting to offer some computer programming courses but not having teachers who are experienced in that field. More volumes in the series are planned.

Educators at all types of schools have affirmed the importance of computer literacy and the fact that the well-educated adult of the future will have to know something about computers. Elementary educators have recently spoken of computer literacy as "the fourth R" (in addition to Reading, Writing, and Arithmetic). And Harvard University now requires all undergraduates to meet a "Quantitative Reasoning" requirement that includes "use of the computer". Harvard students usually meet part of the requirement by writing and running a short computer program. Clearly the computer in education is an idea whose time has come.

COMPUTER MANAGED INSTRUCTION

The computer can save today's overworked teacher some valuable time by providing some of the supplementary drill work that many students need so badly. Another time saver is computer managed instruction. Several Radio Shack programs, including the K-8 MATH WITH STUDENT MANAGEMENT program, NUMERIC DATA ENTRY PRACTICE, C.A.R.D. I: SENTENCES, the STUDENT RECORDS SYSTEM for the HIGH MOTIVATION READING SERIES, QUICK QUIZ: A MINI-AUTHORING SYSTEM, feature automatic storage of student scores on diskette. This saves time when the lesson is taken because the teacher can have permanent records without having to record performance information off the screen when the student finishes the lesson. And it saves time later because the computer calculates the score—the teacher doesn't have to sit up all night grading quizzes and exercises.

AUTHORING SYSTEMS

The Radio Shack Education Division is busy providing authoring systems which enable teachers to design and create their own courseware lessons in any subject without having to have a knowledge of programming. We offer a series of these systems, appropriate for everything from test generation to development of a complete instructional series with record keeping, conditional branching, and student management. Examples are TRS-80 MicroPILOT™ and TRS-80 AUTHOR I.

RESOURCE MATERIALS FOR EDUCATORS

Educational Resource Materials has been an important area of recent emphasis for our Publications group. The RADIO SHACK PROPOSAL WRITING GUIDE is a detailed step-by-step guide to writing successful proposals for CAI funding. The brand new EDUCATIONAL SOFTWARE SOURCEBOOK lists many TRS-80 educational software products available from numerous companies and individuals.

To introduce educators to microcomputer applications, Radio Shack publishes the MICROCOMPUTER INFORMATION HANDBOOK FOR EDUCATORS. A reference source for classroom use is the RADIO SHACK K-8 MATH CROSS-REFERENCE, which matches lesson numbers from the Radio Shack K-8 Math Program with unit, chapter, and page numbers from six of the major mathematics basal series.

SPECIAL HARDWARE

Our special hardware for the classroom is designed to provide teachers and schools with the most convenient, cost-effective configurations possible. Our first educational hardware product, the NETWORK 1 CONTROLLER, allowed up to 16 TRS-80 cassette-based student stations to be connected to a single host TRS-80 disk system, thereby allowing the stations to share the disk and printer on the host. The more recent NETWORK 2 CONTROLLER replaces the original Network 1 and supports the higher communication rate possible with the Model III, while still retaining compatibility with the many Model I's in schools.

The new NETWORK 3 is a polled system that performs different networking functions from the Network 2. Among the new features of Network 3 is the ability to use student management programs with the Network configuration.

WHO IS THE EDUCATION DIVISION ?

Supporting educational applications of the TRS-80 microcomputer to help YOUR school's microcomputer ideas become reality is the purpose of the Radio Shack Education Division. The Education Division includes six different areas: (1) Educational Product Development (including the areas of Software and Publications), (2) Educational Consultants, (3) Educational Sales, (4) Publisher Relations, (5) Industrial Training, and (6) National Bid.

The Educational Product Development Department produces the instructionally sound, effective, and properly validated microcomputer-based education materials that we market. The number of writers and programmers in this department has more than doubled in the past year. This department is divided into the two areas of Software Development and Publications.

The Software Development area uses in-house programmers and works with contract programmers to develop the computer programs in Radio Shack's educational courseware packages. Operation and content of our software are carefully polished through testing, revision, and retesting. Programmers, testers, and writers give special attention to user-friendliness, making sure that program options are clearly defined, that the program is easy to use, and that the student is constantly reinforced with positive messages or with guiding hints and corrections. Radio Shack Consultants, experts in various fields of education, are called upon to evaluate and polish program curricula. Screen displays are scrutinized for neatness, accuracy, and clarity.

Once the software is ready, the Publications staff writes and edits the user's manuals, which are essential parts of the courseware packages. Traditionally, a "walk-through" or hands-on demonstration of the program is at the core of Education Division manuals to provide an easy introduction to the program's main features. Additional features of the program may be outlined in a "reference section." Where appropriate, a summary of the content of each lesson is included. Our goal is to insure that our courseware content, manual text, binders, artwork, and so on, measure up to the textbook standards used by other educational publishers.

In addition to working on user's manuals, Publications staff produce workshop and training materials and educational resources for classroom and administrative use.

Another important resource is our Educational Consultants. These consultants provide expert guidance in curriculum design and other education-related issues. Among our consultants are experts in reading and mathematics/science instruction, district-wide computer applications, courseware design and the use of authoring systems, grants and federal funding, and computer literacy instruction. University professors, former school district superintendents, and other educators experienced both in teaching and administration are among our consultants. They work to help us keep the educator's viewpoint constantly in sight.

Radio Shack Regional Educational Sales Coordinators are the Educational Sales personnel in the field. These are the people who support Radio Shack stores in helping schools and districts determine how to meet specific educational computing needs. Educational Sales has grown rapidly, from 9 full-time Regional Coordinators this time last year to a current team of 21, located around the country.

These Regional Coordinators know that before a microcomputer purchase is made, schools have to spend a lot of time in planning their applications, comparing configurations and costs, and providing inservice training for the teachers who will be using these systems in their classrooms. Therefore, a major part of the Coordinator's job is serving as resource person for school districts. They frequently give workshops and inservice training sessions on computer use, they work to help schools and districts find answers to their particular microcomputer needs, and they help educational users get in touch with other Radio Shack support people in the area.

The people working in the Publisher Relations area have the important job of maintaining cooperative relationships between the Education Division and major educational textbook publishers. By encouraging educational publishers to produce electronic learning materials for the TRS-80, we're

helping to increase the applications available to you, the user. No microcomputer manufacturer can produce the software to meet every need. Established educational publishers will be an important source of educational software in the years ahead.

Just a few of the publishers the Education Division has cooperative agreements with include South-Western Publishing Co., A Division of SFN; Educational Testing Service of Princeton, N.J.; Random House School Division; Psychological Corporation, A Division of Harcourt Brace Jovanovich; and Harper and Row. By agreements like these, Radio Shack provides marketing, sales, and technical support for courseware development by educational publishers.

Another area, Industrial Training, will become increasingly important in the coming years. Computer technology offers great potential in this kind of post-secondary education, and Radio Shack is currently working to develop such programs. Authoring systems will be especially important in creating training materials for industries of all types.

The final area, National Bid, was established to provide fast, accurate responses to all of the bid invitations and requests for price quotation that Radio Shack receives from educational institutions. Through National Bid, schools can be offered special prices on Radio Shack hardware and software, either in quantity or under the terms of numerous purchase agreements with state education agencies and major school districts.

OUTLOOK: SPECIAL PROJECTS AND PRODUCTS

Now let's look at some of the special projects and products that the Education Division is currently involved in.

CHILDREN'S TELEVISION WORKSHOP SOFTWARE

One exciting new development announced by the Education Division this year is the agreement with Children's Television Workshop, the creators of Sesame Street and The Electric Company. Sometime over the next year, CHILDREN'S TELEVISION WORKSHOP will develop and release a series of ten game-style educational software packages for use exclusively with the TRS-80 Color Computer. Later, two instructional series will be developed and released. One will be a reading classroom package for grades 1-4, emphasizing visual discrimination, matching skills, and reading comprehension.

NEW EDUCATIONAL HARDWARE

In the area of hardware, Radio Shack this year added a third classroom network system to the catalog. The NETWORK 3 CONTROLLER, featured in a July TRS-80 MICROCOMPUTER NEWS article, brings new disk-based capabilities to the classroom. Network 2 provides a fast, convenient way for teachers to download instructional programs through a system of connected tape-based student stations. But now Network 3 gives these tape-based student stations many of the capabilities of individual disk systems, including the ability to run student management programs.

HOME INSTRUCTION SERIES

Radio Shack's HOME INSTRUCTION SERIES is a new courseware development that has just begun to become a reality. The series is designed to provide children with enter-


taining education through programs that are relatively inexpensive and run on the minimum 4K Color Computer target system. Packages released so far emphasize language skills for children reading at about grade levels 4-6. For READING IS FUN packages are now available (these are spin-offs of the classroom HIGH MOTIVATION READING SERIES). Two VOCABULARY TUTOR programs are soon to follow. Watch for many new additions to this promising series.

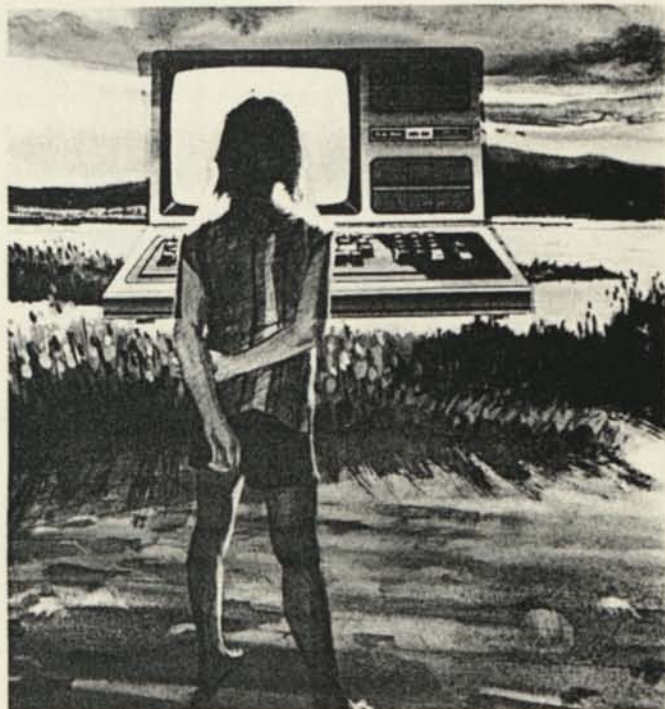
Color LOGO

Later this year, Radio Shack will release Color LOGO, the TRS-80 version of the well-known LOGO program. Color LOGO will run on a 32K disk-based or 16K tape-based TRS-80 Color Computer. Color LOGO features a "turtle" that moves around on the screen and leaves a trail of color in response to directions that are typed in at the keyboard. When children tell the computer how to move, they learn about direction, distance, and angle. More importantly, they have the chance to be creative.

Color LOGO allows you to choreograph fairly sophisticated turtle movements and save these mini-programs on tape or disk. Adults as well as children should find hours of fascination in this program.

AND IN THE FUTURE . . . ?

The products and activities you've just read about are only a few of the things that are new at the Radio Shack Education Division. Also coming forth will be a comprehensive series of Chemistry simulations for the TRS-80 Color Computer and for the TRS-80 Model I and Model III. Also, the second volume of our popular K-8 Math Program, more Vocabulary Tutor programs for the Color Computer, more volumes in the Radio Shack Computer Education Series . . . and the list goes on and on. In the future, you can expect to see Radio Shack continue to produce more education-oriented microcomputer products than any other microcomputer manufacturer. 



Programming Techniques

Auto FIELD Routine for Model II

Max Lupul
10448 Amigo Avenue
Northridge, CA 91326

I have numerous programs which use multiple random access files containing a hodge-podge of fields and I have gotten tired of writing lengthy codes to field these files. I have solved my problem with the enclosed 'all purpose file fielder' which uses a string 'A\$' to contain all the specs for a specific file.

The first two characters in 'A\$' contain the buffer number I wish to use, the balance of 'A\$' contains two-character specifications of the field lengths. Thus line 10 in the listing specifies that buffer #1 will contain a field of 14 bytes, followed by one of 12, followed by 15 2-byte fields, 4 4-byte fields and one 35-byte field. Line 30 specifies that buffer #2 will contain a 22-byte, a 21-byte and a 2-byte field.

'L%' in the subroutine computes the length of the buffer; 'F%' computes the length of the masking dummy variable, while X\$ is a double subscripted buffer variable, with the first subscript identifying the position in the buffer and the second the identity of the buffer. The DIM X\$ statement should dimension X\$ to the largest buffer size in the first subscript and to the number of buffers in the second subscript.

I am providing this with the thought that some other users might find this approach viable, or might want to modify it to suit his/her application.

```

5 CLEAR 10000
  : DIM X$(24, 2)
10 A$="0114120202020202020202020202020202020202020404
  040435"
20 GOSUB 5000
30 A$="02222102"
40 GOSUB 5000
50 STOP 'PROGRAM WOULD CONTINUE HERE
5000 L%=0
  : FOR I=3 TO LEN(A$) STEP 2
  : L%=L%+VAL(MID$(A$, I, 2))
  : NEXT I
5010 OPEN "D", VAL(LEFT$(A$,2)), "TEST"+LEFT$(A$,
  2)+"/DAT", L%
5020 F%=0
  : FOR I=1 TO LEN(A$)/2-1
  : FIELD VAL(LEFT$(A$, 2)), F% AS D$,
  VAL(MID$(A$, I*2+1, 2)) AS X$(I,
  VAL(LEFT$(A$, 2)))
  : NEXT I
5030 RETURN

```

String Machine Language Programs

Christopher P. Hamkins
11 1/2 Locust Street
Waterloo, NY 13165

I have discovered an interesting way to access user-defined machine language subroutines from BASIC on my Model II.

The object code is encoded in a string variable, and then a jump to the string contents is executed. This can be accomplished from BASIC.

String variables are stored in two parts, a 3-byte descriptor and the string itself. The descriptor has the form:

length	memory location
_____	_____
1st byte	2nd 3rd bytes

From BASIC, I can get the address of the string descriptor using the 'VARPTR' function. This does not allow me to jump directly to the string, but I can use this to jump to the string descriptor location by defining DEFUSRn=VARPTR-(STRING\$). If I arrange things so that the first byte of the descriptor is a "Jump" code, jumping to the descriptor location will execute a jump to the next two bytes, which are the location of the string. This allows a complete access to the machine language program without ever leaving BASIC, and makes the routine completely relocatable. The disadvantage is that absolute program addressing cannot be used.

I have used this technique to make a BASIC subroutine which invokes the "scroll" service call to protect the top of the screen from scrolling. The listing is enclosed.

P.S. The "Customer Service" section is by far the best and most informative. Please continue it and expand it.

```

110 GOSUB 230
120 'THIS PROGRAM USES THE SERVICE CALL NUMBER
  27--'SCROLL' TO PROTECT THE
130 'TOP OF THE SCREEN FROM SCROLLING ACTION.
  THIS IS DONE WITH A USER-
140 'DEFINED MACHINE LANGUAGE SUBROUTINE ENCODED
  IN THE STRING SC$. THE
150 'TRANSFER TO THE SUBROUTINE IS ACCOMPLISHED
  BY SETTING THE LENGTH OF THE
160 'STRING TO 195, WHICH RESULTS IN A 'C3' CODE
  FOR THE FIRST BYTE OF THE
170 'STRING DESCRIPTOR. SINCE THE 'C3' CODE IS
  THE 'JP pq' STATEMENT, AND
180 'THE NEXT TWO BYTES OF THE STRING DESCRIPTOR
  ARE ACTUAL MEMORY LOCATION
190 'OF THE STRING, JUMPING TO THE STRING
  DESCRIPTOR LOCATION WILL DO A

```

```

200 'JUMP TO THE BEGINNING OF THE ACTUAL STRING.
    THE MACHINE LANGUAGE
210 'SUBROUTINE IS ENCODED BYTE BY BYTE IN THE
    STRING BEFORE THE JUMP.
220 'THE PARAMETER LL IS THE NUMBER OF LINES TO
    BE PROTECTED.
230 INPUT LL
240 IF (LL<0) THEN LL=0
250 IF (LL>22) THEN LL=22
260 SC$=CHR$(6)+ CHR$(LL)+ CHR$(62)+ CHR$(27)+
    CHR$(207)+ CHR$(201)+ STRING$(189,CHR$(118))
270 LZ=VARPTR(SC$)
280 DEFUSR0=LZ
290 DUMMY=USR0(DUMMY)
300 RETURN

```

```

230 IF F$="P" THEN SYSTEM "DUAL ON"
    : GOTO 70
240 IF F$="O" THEN SYSTEM "DUAL OFF"
    : GOTO 70
250 IF F$="E" THEN PRINT CHR$(26); "TC";
    CHR$(25); AS
    : END
260 IF F$="C" THEN CLS
    : GOTO 70
270 PRINT CHR$(26); "ERROR - UNKNOWN FUNCTION";
    CHR$(25)
    : GOTO 70

```

Calculator

Allen Banick, Jr.
2612 Redlands Drive
Costa Mesa, CA 92627

```

10 'BASIC CALCULATION PROGRAM THIS PROGRAM
    WRITTEN FOR THE TRS-80 MOD-II BY ALLEN
    BANICK, JR. 2612 REDLANDS DR. COSTA MESA CA.
    92627 THIS PROGRAM MAY BE PUBLISHED AS I
    CLAIM NO COPY RIGHTS
20 'THIS IS A SIMPLE ADDING MACHINE PROGRAM USING
    SOME OF THE SPECIAL FEATURES OF THE MOD-II.
30 'FUNCTIONS
    <F1>      ==      <F2>      ==
    <LEFT ARROW>=*    <RIGHT ARROW>=/
    <UP ARROW> =SUB-TOT <DOWN ARROW> =TOTAL
    <E>       =END    <C>       =CLEAR
                                SCREEN
    <P>       =PRT ON <O>       =PRINT OFF
40 ' ADDITIONAL FUNCTIONS CAN EASILY BE ADDED
    WITHIN LINES 180-270
50 DEFDBL A-D
60 CLS
70 PRINT
    : A=0
    : F$=""
80 PRINT "F>";
90 F$=INKEY$
100 IF F$="" THEN 90
110 IF F$=CHR$(01) THEN F$="+"
120 IF F$=CHR$(02) THEN F$="-"
130 IF F$=CHR$(28) THEN F$="*"
140 IF F$=CHR$(29) THEN F$="/"
150 IF F$=CHR$(30) THEN F$="S"
160 IF F$=CHR$(31) THEN F$="T"
170 IF F$="+" THEN PRINT F$;
    : INPUT A
    : AS=AS+A
    : GOTO 70
180 IF F$="-" THEN PRINT F$;
    : INPUT A
    : AS=AS-A
    : GOTO 70
190 IF F$="*" THEN PRINT F$;
    : INPUT A
    : AS=AS*A
    : GOTO 70
200 IF F$="/" THEN PRINT F$;
    : INPUT A
    : AS=AS/A
    : GOTO 70
220 IF F$="S" THEN PRINT CHR$(26); "ST";
    CHR$(25); AS
    : GOTO 70

```

Read Screen

Alfred Pacheco
349 Westmoreland
Richland, Washington 99352

The Model II BASIC INTERPRETER is missing one command which most other versions of BASIC have. With the Model II BASIC you cannot read an existing character from the screen. You can, however, read a character that is about to be placed on the screen using the INKEY\$ instruction. Unfortunately, if your program needs to know if a certain position on the screen is vacant or not, there is no command that enables you to do this.

The following machine language program was written for this very purpose. It can be entered using the DEBUG utility program and saved using the DUMP command and recalled later by your BASIC program.

The only variable needed by the subroutine is the position of the screen you wish to read. This position must be in Scroll Mode as described in the User Manual. If you are using the graphics mode, the coordinates must first be converted to Scroll mode before entering the subroutine. This can be done quite easily using the following function:

```

DEF FNVH (V,H) = 80 * V + H
HV = FNVH(V,H)

```

where H is the horizontal position (0-79), and V is the vertical position (0-23). HV is then the desired position on the screen in the scroll mode. H,V and HV are in integer format.

USE OF THE SUBROUTINE IN A BASIC PROGRAM

When you enter BASIC, make sure that you leave 16K memory above EF00 reserved for the machine language program. Do this by typing in:

```
BASIC-M:61184
```

when entering BASIC.

Next, load and define the subroutine with the following:

```
SYSTEM"LOAD SCREEN/MAC"
DEFUSR0 = &HEF00
```

To use the subroutine the command

```
HV = USR0(HV)
```

is used. HV is the screen position you wish to read in scroll mode. HV is in integer format.

The Hex value of the character will be returned in H. If there was no character there, then HV=32. The Hex code can be converted to the actual character by command

```
AS = CHR$(HV)
```

LOADING THE PROGRAM

To enter the machine language program, follow these steps:

- 1). When in TRSDOS READY, enter "DEBUG ON". After the response "DEBUG ON", enter "DEBUG".
- 2). When in DEBUG mode, enter "M" <ENTER> "EF00" <ENTER>.
- 3). Press the <F1> key.
- 4). Enter the hex code as shown in Figure 1.
- 5). Press the <ESC> key.
- 6). Press the <S> key.
- 7). Type in "DUMP SCREEN/MAC START = EF00, END = EF24".

The program that follows (Figure 2) illustrates the use of this subroutine. The program will ask you to input the coordinates of the spot on the screen you wish to read and echo the character at the present cursor location.

TRS-80 MODEL II DEBUG PROGRAM

```

EF00 EB ED 53 23 EF 1A 6F 13 1A 67 BE 50 D6 01 3E 17
EF10 CF 45 16 01 2A 23 EF 3E 08 CF 2A 23 EF 23 36 00
EF20 3E 02 C9 00 00 00 00 00 00 00 00 00 00 00
EF30 00 00 00 00 00 00 00 00 00 00 00 00 00 00
EF40 00 00 00 00 00 00 00 00 00 00 00 00 00 00
EF50 00 00 00 00 00 00 00 00 00 00 00 00 00 00
EF60 00 00 00 00 00 00 00 00 00 00 00 00 00 00
EF70 00 00 00 00 00 00 00 00 00 00 00 00 00 00
    
```

Figure 1

```

10 CLS
20 PRINT "THIS IS A SAMPLE OF THE TEXT ON THE
   SCREEN"
30 PRINT "TO READ ANY LETTER IN THIS TEXT CHOSE
   THE PROPER"
40 PRINT "COORDINATE OF THAT LETTER USING THE
   GRAPHIC COORDINATE SYSTEM"
50 SYSTEM "LOAD SCREEN/MAC"
60 DEFINT H, V
70 DEFUSR0=4NEF00
80 DEF FNHV(V, H)=80*V+H
90 INPUT "H": H
100 INPUT "V": V
   : PRINT ""
110 HV=FNHV(V, H)
120 HV=USR0(HV)
130 PRINT CHR$(HV)
140 GOTO 90
    
```

Figure 2

```

00100 EX DE,HL ;PUT ASA ADDR. IN DE REG
00200 LD (ADDR),DE ;SAVE ASA ADDRESS IN ADDR
00300 LD A,(DE)
00400 LD L,A ;GET ARGUMENT AND STORE
00500 INC DE ;IT IN REG PAIR
00600 LD A,(DE) ;HL
00700 LD H,A ;
00800 LD C,00 ;CONVERT SCROLL COOR. TO GRAPHIC COOR.
00900 LD B,1 ;BY DIVIDING ARGUMENT
01000 LD A,23 ;BY 00
01100 RST 8
01200 LD B,L ;LOAD ROW POSITION INTO REG B. COLUMN
01300 LD D,1 ;POSITION IN REG C. READ ONE CHARACTER
01400 LD HL,(ADDR) ;AND STORE IN ASA
01500 LD A,11 ;
01600 RST 8
01700 LD HL,(ADDR) ;SET HIGH ORDER BYTE OF ASA
01800 INC HL ;TO 0
01900 LD (HL),0
02000 LD A,2
02100 RET ;RETURN TO BASIC
02200 ADDR: DW 0
02300 END
    
```

Figure 3

General Electric ASCII Sequential Files

Richard Halloran
155 Jackson Street No. 1704
San Francisco, CA 94111

TRSDOS for Models I, II and III supports two types of data files—Sequential and Random Access. There is a third

type called General Electric ASCII sequential files or merely "terminal" files as they were used on devices similar to the Telex terminals.

Terminal files work on the TRS-80s and are very simple to set up and use. To illustrate, go into BASIC and type "AUTO" and then type in the following:

```

10 DICK,145,MYSELF
20 EDWARD MILES HALLORAN,282,MY NEPHEW
30 EDWARD,167,MY BROTHER
    
```

Now save that "program" to disk using the command: SAVE "NAMES",A

which saves the "program" in ASCII form as a data file!

Now type NEW and enter the following program to access the "NAMES" data file.

```

10 CLEAR 500
   : CLS
20 OPEN "I",1,"NAMES"
30 FOR N=1 TO 3
40 INPUT #1,N$,W,R$
   : 'FIRST TIME THE FIELDS ARE NAMED !
60 TW=TW+W
70 PRINT MID$(N$,4) TAB(24);W TAB(30) R$
80 NEXT
90 CLOSE
100 PRINT
110 PRINT USING "THE TOTAL WEIGHT IS ### POUNDS": TW
120 PRINT USING "THE AVERAGE WEIGHT IS ### POUNDS
   ":TW/3
    
```

Now, save this program to disk as "NAMES1" and then run it. The results should be as follows:

```

DICK                145 MYSELF
EDWARD MILES HALLORAN 282 MY NEPHEW
EDWARD              167 MY BROTHER

THE TOTAL WEIGHT IS 514 POUNDS
THE AVERAGE WEIGHT IS 171 POUNDS
    
```

Now let's see what is going on here. First we entered our data list consisting of three records with three fields each - a name, the persons weight, and the persons relationship to me. Then we saved the list in ASCII to disk and thus created a data file called "NAMES".

Next let us look at our BASIC program "NAMES1" which accesses and manipulates the data in "NAMES." Line 10 reserves string memory and clears the screen. Lines 30 and 80 set up a loop to input all the records. Line 40 accesses "NAMES" in sequential input mode and "declares" the fields N\$,W, and R\$. Line 60 accumulates the numeric variables fielded as W. Line 70 tabulates and prints the records. Note that the MID\$(N\$,4) prints all of the remainder of the field N\$ after the third character. This suppresses the printing of the BASIC line numbers. Line 90 closes the file and lines 110 and 120 formats the printout of the results of the calculations. (Note that the numeric variable "W" does not have to be "converted" or "made" as in random access.)

What are some of the advantages of terminal files?

1. Very simple to learn how to use. Variable length fields
2. Data can be entered faster than if using input prompts.
3. The data file can be loaded into BASIC and edited.
 - A. Line (Record) Editing
 - B. Add, insert, delete records, add fields
 - C. Scroll for checking data, etc.
4. Merge and append data files.
5. Insert EOF flags to make jump to successive drives.
6. No restrictions on the "driver" programs.
7. Efficient use of disk space

Model II Bugs, Errors, and Fixes

NOTE TO USERS:

The following program changes and corrections are provided for your information. If you have an applications program which is working correctly, you should probably NOT make any changes to it. If you feel that the changes should be made, but you do not feel qualified to make the changes yourself, please contact your local Radio Shack Computer Center or Expanded Computer Department for assistance. If you do not have access to one of these stores, then you may want to call Computer Customer Services in Fort Worth for assistance.

CHANGES TO BASIC PROGRAMS

There are general procedures that need to be followed when any corrections are made.

1. Make a backup of the disk that contains the program to be corrected. Changes should be made on the backup copy.
2. Load the program to be changed by typing LOAD "filename" where filename is the name of the program to be modified.
3. Make the line changes indicated in the fix. For existing line numbers, edit or retype the line to match the one in the fix. New lines should be entered.
4. Save the corrected program (the one now in memory). Type SAVE "filename" <ENTER> where filename is the name of the program that has been modified.
5. Now make a backup of the corrected diskette.

PATCHES

PATCHes are entered from TRSDOS READY and are used to make corrections to files stored on the disk.

1. Before making a PATCH, back up the diskette that requires modification and make the PATCHes to the backup copy of the diskette.
2. Apply PATCHes according to the information given in your TRSDOS manual.

RUNCOBOL

Users of the following programs:

- 26-4601 General Ledger
- 26-4604 Accounts Receivable
- 26-4605 Accounts Payable
- 26-4607 Order Entry/Inventory Control
- 26-4608 Sales Analysis
- 26-4703 COBOL Development System
- 26-4704 COBOL Runtime

All of the COBOL programs in the above list include the program RUNCOBOL. There are several sets of PATCHes that need to be made to the program RUNCOBOL. The first PATCH should only be applied to the programs specified in Correction 1.

The PATCHes in the other corrections should be applied to all the programs in the above list.

Correction 1—If one of the following programs is abnormally terminated, a problem may occur. An abnormal termination is one in which you suffer a power loss, press the <RESET> key, press the <BREAK> key, or turn your machine off while running the program. Conversely, a normal termination is one where the program is exited according to program instructions.

When an abnormal termination occurs, any data files you were working with at the time are rendered unuseable, even though they may appear to still be good. While some programs detect an abnormal termination as an ERROR 98, the following programs do not.

- 26-4601 General Ledger
- 26-4604 Accounts Receivable
- 26-4605 Accounts Payable
- 26-4607 Order Entry/Inventory Control
- 26-4608 Sales Analysis

You must—GO TO YOUR BACKUP. For Hard Disk users, RESTORE your system.

If you continue processing, the results will be unpredictable. Remember that the first thing you should do after bringing up backup data files as your new working data files is create a new set of backup data files.

We have developed the patch below for RUNCOBOL. This patch will assist you in determining if you have a problem resulting from an abnormal termination. However, it is not a complete answer. When this patch has been applied and the program determines that the error has occurred, the following message will be displayed:

(DATA FILE NAME) ERROR 34

By pressing <ENTER> all files that are currently open will be closed, and the program will return to TRSDOS Ready. This way you will know that you should go to your backup.

PATCH RUNCOBOL A=6E90, F=98, C=34 14

NOTE: This PATCH should only be used with the programs indicated in Correction 1. It could cause inaccurate error detection if used with any other COBOL programs.

Correction 2—When attempting to add records to indexed or relative file, COBOL improperly handles a disk condition (error 24). The indices of indexed files are improperly changed. As a consequence, the next attempt to write record produces an invalid index error (error 98).

At this point, the user can only revert to his most recent back-up copy, because the invalidated file cannot be recovered.

At TRSDOS READY, apply the following PATCHes to correct the above problem.

PATCH RUNCOBOL A=761D F=3014FD7071FD71702111
C=D021250019EB3E2CCF:F
PATCH RUNCOBOL A=7626 F=250019EB3E2CCF280414
C=2007FD7071FD7170C9:9

GENERAL LEDGER (26-4501)

The batch total does not equal the Document Total at document printing. This problem can also occur during printing. The problem is corrected by making the following changes to the program Txentry. (Version 1.2)

Line: 1450 IFHI<LOTHEN1405ELSEFORI=0TONA-11
:S0(I)=20:NEXT:BT#20

Add the following line:

6345 IFIN0>0THENBT#-BT#-INV
6740 IN#-BT#:LPRINTTAB(28)"BATCH TOTAL"TAB(42);)
:GOSUB5100:LPRINT " ":L=L+1:RETURN

DELETE Line 6730
 SAVE "Txentry" to store the program changes.
 Make the following changes to the program Txpost.
 Line: 420 FORJ=BTOMA-1:50(J)=20:NEXT:BT#=20:RETURN

Add the following new lines:

```
740 IN#-BT#:LPRINTTAB(28)"BATCH TOTAL"
  TAB(42);:GOSUB5100:LPRINT" ";L=L+1:RETURN
3345 IFIN#>0THENBT#-BT#+IN#
```

DELETE LINE 730
 SAVE "Txpost" to store the program changes.

PAYROLL (26-4503)

Following are corrections to separate problems with Payroll.

Correction 1—The Federal unemployment insurance rate factor has been reported to be to the third decimal place. The Payroll program will allow changes of up to the second decimal place. (Version 1.0)

Make the following changes to the program CHNGCO.

```
Line: 1620 CLEAR2000:GOSUB400:CLS:PRINTCFS
:PRINT@ (2,16),RS" OTHER RATES
AND L I M I T S "NS:DIMFT(9),FT(11)
:F9$="###,###,##-":F8$="####":F7$=
LEFT$(F9$,10)+"-"
```

```
*Line: 1670 FORI=1TO8:PRINT@(I+6,0),RS;USING" ## ";
I;:PRINTNS;TAB(10);C6$(I);:FORK=1TO2
:J=VAL(MID$(C7$(I),2*K-1,2));IFJ>0ANDI=5ANDK=1
THENPRINTTAB(K*13+12);USINGF7$;FT#(J);
ELSEIFJ>0THENPRINTTAB(K*13+12);USINGF9$;FT#(J);
```

```
Line: 1760 PRINT@(19,0),EL$:PRINT@(19,8),P$;C6$(I);" ";
:FL=8:GOSUB300:IFCF=1ORCF=2THENK=2:NEXT
:GOTO1710ELSEIFLEN(IN$)>0THENFT#(J)=VAL(IN$)
:IFI=5ANDK=1THENPRINT@(I+6,K*13+12),USINGF7$;
FT#(J);ELSEPRINT@(I+6,K*13+12),USINGF9$;FT#(J);
```

SAVE "CHNGCO" to save the changes in the program.

Correction 2—Two new steps have been added to the calculation process for Minnesota State Tax Tables. The additions involve a 7% surcharge. The necessary program changes can be implemented by using the following procedure. (Version 2.0 & Prior)

Make the following changes to the program STATETAX.

To change Step 6 of the Minnesota Tax Calculation edit line 1820 to read as follows:

```
Line: 1820 NEXT:PRINT#1,USINGL$;4000:PRINT#1,
"##(I)=INT(E#(1)*1.084#100+.5)/1000
:IFE#(1)<0THEN#(I)=0:RETURNELSERETURN"
```

To change Step 2 of the Minnesota Tax Calculation add the following line.

```
Line: 1785 PRINT#1,USINGL$;SN+35:PRINT#1,"CP=CP+.909D0"
```

SAVE "STATETAX" to save the changes in the program.

Correction 3—In the Prepare Checks section of the Model II payroll program there is a problem. If you enter (choose to prepare a check for) an employee number that is greater than the total number of employees in your Payroll system the number entered will replace that of the employee currently displayed. (Version 2.0 and Prior)

Make the following changes to the program INPUT.

```
Line: 510 IFCF<>5THENIFCF<>0THEN500ELSEC=VAL(IN$)-EO
:IFC<10RC>LOF(3)THEN500ELSEN=C
Line: 520 IFN<10RN>LOF(3)THEN500ELSEIFCF=0THENN=
INSTR(C1$,CHR$(N))
```

SAVE "INPUT" to store the program changes.

ACCOUNTS PAYABLE (26-4505)

Following are two sets of corrections for Accounts Payable.

Correction 1—Occasionally you may need to edit YTD purchases/payments. Make the following changes to the program APS/BAS. (Version 1.0)

```
Line: 200 IFW=1THEN220ELSEIFW$=CHR$(0)THENPRINTW$;FL$;B1$;
:W=W-1:MID$(IN$,W,1)=FL$:GOTO140
```

```
Line: 1800 GOSUB8500:PRINT@(23,42),:;FL=1:SZ=1
:GOSUB100:SZ=0:IFCF=1THEN1000ELSE
IFCF=5THEN1800
```

ADD the following new lines.

```
Line: 222 IFSZ<>1THEN230ELSEIFASC(W$)=19THENPRINT@
(22,20),"YTD PURCHASES & CREDITS";:INPUTN#
:MI#=A1#-N#:A1#-N#:N#A1#:GOSUB5000:AA$=V$
:GOTO226
```

```
Line: 224 IFASC(W$)=16THENPRINT@(22,20),"YTD PAYMENTS &
DEBITS";:INPUTN#:MI#-N#:A2#-N#:GOSUB5000
:AB#=V$ELSE230
```

```
Line: 226 GOSUB9000:PRINTNS;:GOSUB8400:GOSUB9200
:IN$="X":RETURN
```

```
Line: 5000 CP#=CP#-MI#:W#=ABS(N#)*100+.1:X=W#/D1#
:W#-W#-X*D1#:V$=CHR$(X-(N#<0)*128)
:X=W#/D2#:W#-W#-X*D2#:V$=V$+CHR$(X)
:X=W#/D3#:W#-W#-X*D3#:V$=V$+CHR$(X)+
CHR$(W#):RETURN
```

SAVE "APS/BAS" to save the changes in the program.

Run the program using the following options which are accessed from the VENDOR FILE maintenance menu:

CTRL S TO Edit PURCHASES

CTRL P TO Edit PAYMENTS

Correction 2—The check number does not increment if a check is voided (pre-printed checks only). Make the following changes to the program APCHECKS/BAS. (Version 2.0 and Prior)

```
Line: 2060 IC=IC+1:IFIC=14THENLPRINTL1$:GOSUB8000
:IC=0:CF=0:LC=LC+1
```

ADD this line:

```
Line: 3075 IC=IC+1:IFIC=15THENLPRINT"":LPRINTL$
:LPRINT"***** CHECK NUMBER "CN" VOIDED
***** - CONTINUED ON CHECK NUMBER
"CN+1:LPRINTLL$:CN=CN+1:IC=0
```

SAVE "APCHECKS/BAS" to save the changes in the program.

MAILING LIST II (26-4507)

Once a given capacity is reached, the Mailing List program will not expand, and an error message is displayed. (Version 2.0)

Make the following changes to the program SETUP/BAS.

```
Line: 1270 IFRV=<0ORRV>CPHEN1250
```

Add the following line:

```
Line: 985 IFM=85THENC=3000ELSECP=2000
```

SAVE "SETUP/BAS" to save the changes in the program.

If you have already set up your Mailing List file then you need to complete the additional following steps.

At TRSDOS READY type in SETUP (ENTER).

Type (R) followed by the name of the system.

Press the (F2) key.

Press the (F1) key.

The system will show maximum capacity but will expand on the disk, one record at a time.

STATISTICAL ANALYSIS (26-4540)

Variable names do not change as variables change in the Correlation Matrix. (Version 1.0)

The problem is corrected by making the following changes to the program CORRL/BAS.

```
Line: 120 PRINT:LLZ=MEM/8-1000:DIMX$(LLZ),B$(300),
      VS$(12),XS$(12),VNS$(12),NNS$(12)
```

```
Line: 2060 ONERRORGOTO0:NRZ=CVI(NRS):NVZ=CVI(NVS)
      :FORJ=1TO12:VNS(J)=VS(J):NNS(J)=VS(J):NEXTJ
```

ADD the following new line:

```
Line: 2140 FOR DD=1 TO NV:VNS(DD)=NNS(WVZ(DD)):NEXT DD
```

Type SAVE "CORRL/BAS" to save the changes in the program.

ACCOUNTS RECEIVABLE (26-4554)

If option 10 is chosen at the MAIN MENU so that system initialization is exited and you are returned back to the MAIN MENU, a BASIC Error Code 8 in line 51060 will occur. This is not a fatal error, and you may continue execution of the program by pressing <ENTER>. To prevent this error message from occurring make the following change to the program ARSI/BAS. (Version 1.1)

```
Line: 51060 IFRS<>"END"THEN RETURN ELSE GOTO 40000
```

Type SAVE "ARSI/BAS" to save the change.

RSCOBOL GENERATOR (26-4707)

Due to some misspelled words, COBOL Generator has a flaw in the code it produces. While this problem is not crucial, it does necessitate using an Editor to make corrections prior to compiling which brings about another problem.

The system is capable of generating 500K of code, but the CEDIT module only accommodates 32K of code thereby limiting the editing capability to 32K. Putting the system on Hard Disk does not increase the capacity of the Edit function. Our Text Editor (26-4710) does not work on Hard Disk. However, if you are using the Hard Disk in generating your code you can FCOPY it to 2.0a and then use the Text Editor (26-4710) to edit the code. Then FCOPY the file back to Hard Disk and compile it.

BISYNC (26-4715)

The BiSync package will lock up during Log On when interfacing with TSO running with TCAM. The following patch is to be used with version 3.1 of the BiSync package to correct the problem.

```
PATCH BIS3270 A=7203 B=FE27 C=FE26
```

Computer Clubs

THE COMPUGUILD
c/o Ben F. Davis
3513 Tripp
Amarillo, TX 79121
1-806-372-1898
1-806-358-2202

COMPUTER CLUB/TRS-80 USERS GROUP
c/o Harriette Gross
John C. Hart Memorial Library
1130 Main Street
Shrub Oak, NY 10588
1-914-245-5262

JACKSON AMATEUR COMPUTER SOCIETY (JACS)
c/o C.B.C.C. Inc.
2355 Camp Baker Rd.
Medford, OR 97501
1-503-535-6883

MICRO-80 GROUP TRS-80 USERS
c/o Bob Walters
P.O. Box 1472
Eugene, OR 97440-1472

NORTH IOWA TRS-80 USERS GROUP (NITUG)
c/o Chuck Prickett, Jr.
1705 South Coolidge
Mason City, IA 50401
1-515-423-3198

NORTHERN ILLINOIS COLOR COMPUTER CLUB
c/o John Bowman
9346 Landings Square
Des Plaines, IL 60016

RICHMOND TRS-80 USERS' GROUP
c/o MC Studio
4115 Hopkins Rd.
Richmond, VA 23234

SILICON VALLEY COMPUTER CLUB
GTE Systems Bldg. 3
Central Expressway & Whisman Rd.
Mountain View, CA 94043

SILVERLAKE DISTRICT COMPUTER CLUB
2475 Moreno Dr
Los Angeles, CA 90039
1-213-666-2116

SOUTH BAY TRS-80 USERS GROUP (SBUG)
Dysan Bldg.
5401 Patrick Henry Dr.
Santa Clara, CA 95050

TRS-80 MODEL I/III USERS CLUB
c/o Timothy I. Moder, Ph.D.
1932 McGee Avenue
Berkeley, CA 94703
1-415-841-8686

TRS-80 USERS GROUP OF NORTH UTAH
c/o Marcel Kinard
3106 East 125th North
Layton, UT 84041
1-801-544-8970

TRS-80 MODEL I USERS GROUP OF WEST LOS ANGELES
c/o Mike Miller
10210 Woodbine St. #1
Los Angeles, CA 90034
1-213-836-4103

Pocket Computer II Start-Up Instructions

There have been several complaints of defective PC-2s "out of the box".

The complaints are similar in nature and generally take the form of "I can't get it to do anything without producing an error" or "I can enter a program while in the program mode, but when I switch to the run mode it has forgotten the program."

Both of these complaints are traceable to a single source. Page 14 of the PC-2 Owner's Manual is not as explicit as it should be regarding the steps required after replacing the batteries or pressing ALL RESET. The manual should state:

After inserting the batteries and turning the unit over, your display should show:

"NEW0:CHECK?"

Respond by performing the following:


Press <CLEAR>

Type NEW0 and press <ENTER>

Press <SHIFT MODE>

Type NEW and press <ENTER>

These two operations clear the PC-2 main and reserve areas of memory.

It may be necessary for you to perform these operations whenever you change batteries again or press ALL RESET. 

Homework

Brian Cutler
P.O. Box 124
Cairo, NY 12413

This little program is another one of my pocket Memo programs. In school, I have a very hectic schedule. So, I found myself frequently forgetting things. Well, with this program, that soon became a thing of the past. If I keep this program loaded into my computer at all times, I am able to come up with a list of what I have to do each day. The program below still has my course names included in it. This is so that people can see what it does, and they can then modify the headings to suit their own needs.

You can input ten lines of seven characters each for each heading. The last line (line 10) must be an "E." The menu at line 30 allows you to ADD information, PRINT OUT the information stored, or END the session.

```
1 REM HMWK
2 REM VERSION 1.2
3 REM 11/1/81
```

```
10: A$(10)="ENG 232"
    A$(11)="CSI 404"
    A$(12)="CSI 210"
    A$(13)="PHI 210"
20 A$(14)="PHY 104"
    A$(15)="WORK"
    A$(16)="MEMO"
30 INPUT "ADD(A),PRINT(P),END(E);A$
    : IF A$="P" GOTO 88
40 IF A$="A" GOTO 200
45 IF A$="E" GOTO 500
50 GOTO 30
88 PRINT "*****"
89 PRINT "* HOMEWORK AND *"
    : PRINT "* MEMOS *"
    : PRINT "*****"
    : PRINT " "
90 D=9
91 LET B=10
92 D=D+1
    : IF D=17 GOTO 30
93 PRINT"-----"
    : C=B+10
94 PRINT" ( ) ";A$(D)
95 PRINT " "; A$(C)
96 C=C+1
97 IF A$(C)="E" LET B=B+10
    : GOTO 92
98 GOTO 95
200 INPUT "DATE:";F$
    : D=9
201 LET B=10
202 D=D+1
    IF D=17 GOTO 30
203 C=B+10
204 PAUSE A$(D)
205 INPUT "HMWK:";A$(C)
206 IF A$(C)="E" LET B=B+10
    GOTO 202
207 C=C+1
208 GOTO 205
500 PRINT " "
    PRINT " "
    PRINT "DATE:";F$
501 PRINT " "
    : PRINT " "
    : PRINT " "
    : BEEP 10
    : END
```

Partial Sample Printout

```
*****
* HOMEWORK AND *
* MEMOS *
*****
```

```
-----
( ) ENG 232
    READ
    SHORT
    STORY
    END
```

Use of the TRS-80 PC in Celestial Navigation

Donald Parson
4475 N. Ocean Blvd., #43-C
Delray Beach, FL 33444

PROGRAM ONE

For those interested in celestial navigation, here is a program for the PC to solve the spherical triangle and come up with intercept and azimuth without the use of tables. All that is required is sextant, watch and almanac.

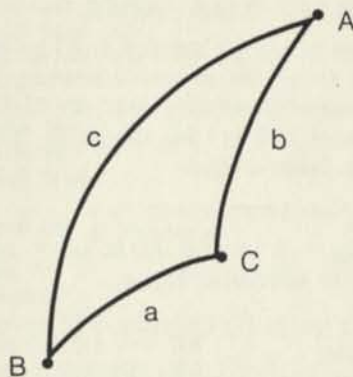


Figure 1

The spherical triangle (Fig. 1) may be solved as follows:
 $\cos a = \cos b * \cos c + \sin b * \sin c * \cos A$ (1)
 Solving for "a" will yield "Hc", the computed altitude.
 $\cos C = (\cos c - \cos a * \cos b) / (\sin a * \sin b)$ (2)
 Solving for "C" will yield "Z", the azimuth angle.

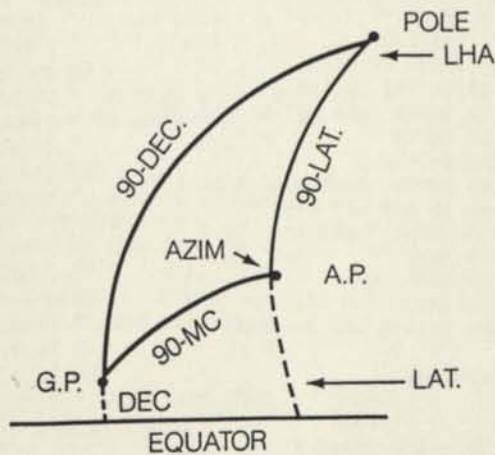


Figure 2

Fig 2. shows how the various inputs are worked into the triangle. Fig. 3 shows what happens when the declination and latitude are of 'contrary name'. "A.P." is the assumed position of the observed body, as taken from the almanac.

This program will compute LHA directly from GHA and longitude, thus disposing of another chore and another chance of human error.

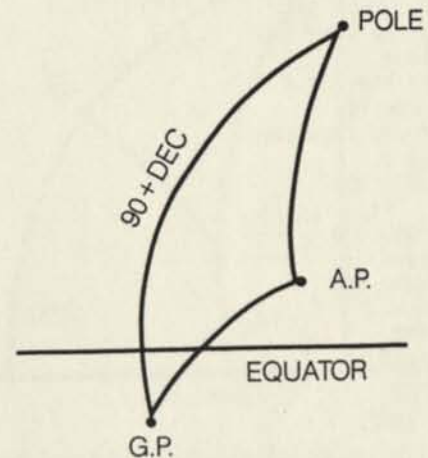


Figure 3

```

10 CLEAR
20 INPUT "LAT, DEG? "; A
21 INPUT "   MIN? "; X
22 A=A+X/60
   : X=0
25 INPUT "IF NORTH LAT, DIAL 1 "; W
30 INPUT "DEC, DEG? "; B
31 INPUT "   MIN? "; X
32 B=B+X/60
   : X=0
33 INPUT "IF SAME NAME, DIAL 1 "; X
34 IF X<>1 THEN 550
35 INPUT "GHA, DEG? "; O
36 INPUT "   MIN? "; P
37 O=O+P/60
38 INPUT "LONG, DEG? "; P
39 INPUT "   MIN? "; U
40 P=P+U/60
41 INPUT "IF WEST, DIAL 1 "; U
42 IF U<>1 LET P=-P
43 IF O<P LET O=O+360
44 C=O-P
45 INPUT "HO, DEG? "; N
46 INPUT "   MIN? "; Y
47 Q=60*(Y-INT Y)
48 N=N+(INT Y/100)+Q*.0001
50 D=90-A
   : E=90-B
   : DEGREE
60 F=COS D*COS E
   : H=COS C
70 G=SIN D*SIN E
80 I=ACS(F+G*H)
   : J=DMS(90-I)
81 Q=100*J-INT(100*J)
82 R=INT(100*J)+Q/.6
83 S=INT(R/100)
84 T=100*(R/100-S)
   : BEEP 2
90 PAUSE "HC="; S; "DEG "; USING "###.##"; T; "MIN"
91 IF J<N THEN 97
92 V=60*(DEG J-DEG N)
96 PRINT V; "   AWAY"
   : GOTO 105
97 V=60*ABS(DEG J-DEG N)
100 PRINT V; "   TOWARD"
105 K=COS E-(COS I*COS D)
110 L=SIN I*SIN D
    
```



```

120 M=ACS(K/L)
130 PAUSE " Z= "; M
    : GOTO 210
140 IF C<180 THEN Z=360-M
210 IF W=1 THEN 300
220 IF C>180 THEN 500
230 Z=180+M
250 PRINT "ZN= "; Z
    : GOTO 10
300 IF C>180 THEN 400
310 Z=360-M
    : GOTO 250
400 Z=M
    : GOTO 250
500 Z=180-M
    : GOTO 250
550 B=-B
    : X=0
    : GOTO 35
600 END

```

PROGRAM TWO

The first program was for sight reduction only. This new one includes Great Circle Distance and initial azimuth (course) from present position to destination. It also furnishes the numbers required for the use of a star finder (HO 2102-D).

Line 10 starts the Great Circle program, line 690 the Star Finder and 905 the Sight Reduction. You will note that the two navigational programs share some of the operations, inasmuch as they use the same basic formulae.

COMMENTS ON "F", SIGHT REDUCTION:

Note that this program can yield an intercept from the DR position, instead of from an 'assumed position' as required when using the tabular solution. This is a big benefit. In line 916 the Z is held for the later determination of azimuth from azimuth angle C in the triangle (see Fig. 4). This determination occurs in line 310, via a "GOTO". Line 926 provides the proper sign for longitude when it is (eventually) used in producing LHA (Angle A in the triangle). Line 940 deals with the situation when the celestial body is on the other side of the equator from the observer. Line 210 converts latitude and declination to b and c in the triangle. In line 262, V would have been zero, had we been using the Great Circle program.

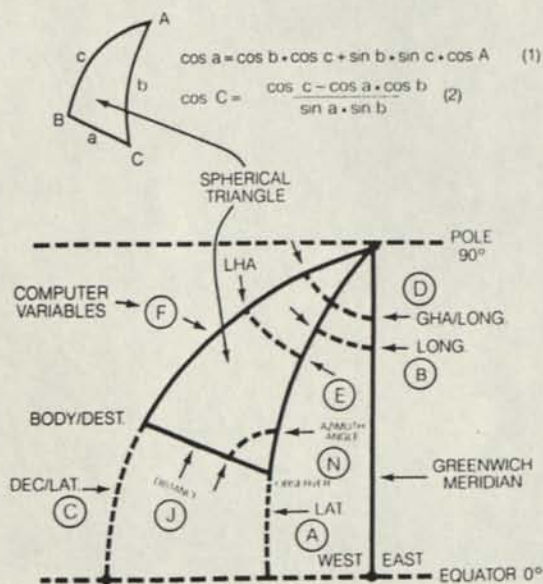


Figure 4

Thus we can adapt the use of the triangle solution according to our requirements. V being 1, we enter the H.O. data and subtract it from (compare it to) the triangle solution. This yields the Intercept, either 'toward' or 'away'. We then jump to line 280 for the azimuth answer, which would be common to the two programs.

COMMENTS ON "D", GREAT CIRCLE:

Line 10 "D" merely changes the wording for some of the inputs, substituting the destination for the celestial body. Line 200 corrects the situation where, in the celestial solution, GHA is always measured westward, whereas in the Great Circle solution the longitude can go either way. V is made to be zero, so that later on (line 270) we print the proper things. In line 260 we multiply the angle by 60 to get miles.

COMMENTS ON "S", STAR FINDER:

"S" of course could be omitted, but I have found that it reduces the star finder set up time from, say, 20 minutes to about 5. It also eliminates many chances for mistakes.

Lines 720 and 730 set the longitude of the observer. They could have been written:

```

700: INPUT "LONGITUDE, DEGREES"; A
710: INPUT "          MUNUTES"; B
720: A=A+B/60
730: INPUT "IF WEST, DIAL 1"; B

```

I happen to be at 80 degrees 3.5' W, and expect to remain for a while.

I hope that the foregoing is of interest. It certainly was interesting to compose it. I wound up with only 13 steps and 1 memory left!

```

5 PRINT " SHFT
  : S, F OR D"
10 "D"
  : CLEAR
20 INPUT "DEP LAT DEG "; A
30 INPUT "          MIN "; B
40 A=A+B/60
  : B=0
50 INPUT "N, DIAL 1 "; Z
60 INPUT "LONG DEG "; B
70 INPUT "          MIN "; C
80 B=B+C/60
90 INPUT "W, DIAL 1 "; Y
100 IF Y=0 LET B=-B
110 INPUT "ARR LAT DEG "; C
120 INPUT "          MIN "; D
130 C=C+D/60
140 INPUT "SAME NAME, DIAL 1 "; X
150 IF X=0 LET C=-C
160 INPUT "LONG DEG "; D
170 INPUT "          MIN "; E
180 D=D+E/60
190 INPUT " W DIAL 1 "; W
200 IF W=0 LET D=-D+360
  : V=0
210 E=90-A
  : F=90-C
  : DEGREE
220 G=COS E * COS F
230 H=D-B
240 IF H<0 LET H=H+360
250 I=SIN D * SIN F * COS H
260 J=ACS (G+I)
  : K=60 * J
262 IF V=1 THEN 400
270 PRINT "GRT CIR DIST="; USING "#####.#"; K; MI
280 L=COS F - (COS J * COS E)

```

```

290 M=SIN J*SIN E
300 N=ACS (L/M)
310 IF Z=1 THEN 350
320 IF H>180 THEN 380
330 O=180+N
340 PRINT "AZIMUTH="; USING "###.#"; O
341 END
350 IF H>180 THEN 370
360 O=360-N
: GOTO 340
370 O=N
: GOTO 340
380 O=180-N
: GOTO 340
400 INPUT " H.O.DEG. "; P
410 INPUT " MIN. "; Q
420 P=P+Q/60
: Q=0
430 R=90-J
: Q=60*ABS (R-P)
440 IF R<P THEN 460
450 PRINT USING "###.#"; Q; "A"
: GOTO 280
460 PRINT USING "###.#"; Q; "T"
: GOTO 280
690 "S" CLEAR
720 A=80+3.5/60
730 B=1
740 IF B=0 LET A=-A
750 INPUT " GHA ARIES, DEG "; C
760 INPUT " MIN "; D
770 C=C+D/60
780 D=C-A
790 IF D>360 LET D=D-360
800 INPUT " GHA, BODY, DEG "; E
810 INPUT " MIN "; F
820 E=E+F/60
830 F=C-E
835 IF E>C LET F=F+360
840 PRINT " LHA, ARIES="; USING "###.#"; D
850 PRINT "R.A. BODY="; F
860 INPUT " GHA, BODY, DEG "; E
870 INPUT " MIN "; F
880 E=E+F/60
890 F=C-E
895 IF E>C LET F=F+360
900 PRINT " R.A. BODY="; F
: GOTO 860
905 "F" CLEAR
910 INPUT " DR LAT DEG "; A
912 INPUT " MIN "; B
914 A=A+B/60
916 INPUT " N DIAL 1 "; Z
918 INPUT " DR LONG DEG "; B
920 INPUT " MIN "; C
922 B=B+C/60
924 INPUT " W DIAL 1 "; Y
926 IF Y=0 LET B=-B
928 INPUT " GHA DEG "; D
930 INPUT " MIN "; E
932 D=D+E/60
934 INPUT " DEC DEG "; C
936 INPUT " MIN "; E
938 C=C+E/60
940 INPUT " SAME NAME DIAL 1 "; X
942 IF X=0 LET C=-C
944 V=1
: GOTO 210

```

PROGRAM THREE

For you navigators, here is a reserve program which will give you the coordinates of Latitude and Longitude directly from the altitudes of Polaris and any other body.

This we can accomplish with only a sextant and the almanac; no reduction tables required.

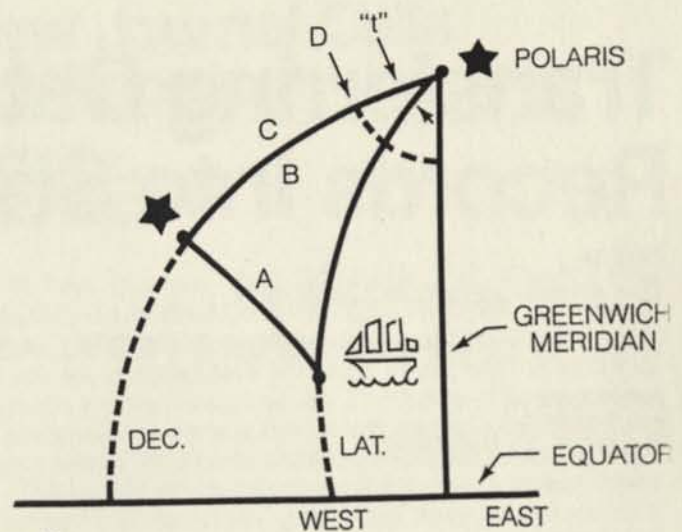


Figure 5

Shown in figure 5 is a spherical triangle, which is involved in our problem. What we do is to solve for the angle "t" in the relationship:

$$\cos "t" = \frac{\cos A \cdot \cos B \cdot \cos C}{\sin B \cdot \sin C}$$

Our reserve program is written:

SHFT A: ACS((COS A - COS B * COS C)/(SIN B * SIN C))

The inputs are as follows:

- A = 90 - Altitude of Body
- B = 90 - Altitude of Polaris
- C = 90 - Declination
- D = Greenwich Hour Angle

You must also know whether you are East or West of Greenwich, whether the body bears Easterly or Westerly and that you are in the Northern Hemisphere. If the Declination is of opposite name, then:

$$C = 90 + \text{Declination}$$

To operate the program, the keying would be like this:

1. In West Longitudes, with the Bearing Easterly:

<D> <+> <SHFT><A> <ENTER>
2. In West Longitudes, with the Bearing Westerly:

<D> <-> <SHFT><A> <ENTER>
3. In East Longitudes, with the Bearing Easterly:

<3><6><0> <-> <D> <-> <SHFT><A> <ENTER>
4. In East Longitudes, with the Bearing Westerly:

<3><6><0> <-> <D> <+> <SHFT><A> <ENTER>

In each case, you will read the Longitude after the last <ENTER>. If the result exceeds 360, you simply subtract 360.

The Latitude, of course, is equal to the Altitude of Polaris.

Transferring Data From Long Records Into Shorter Records

Profile

The small Computer Company

Written for the users of PROFILE II, PROFILE + and PROFILE III +

PROFILE Editor
P.O. Box 2910
Fort Worth, TX 76113-2910

Mr. Robert D. Lynch of Glassboro, New Jersey, writes: "I have a TRS-80 Model II with PROFILE II. Before I understood your patches described in the June 1981 'News' which allows me to shorten segments and increase my number of records, I already had a sizeable file which is now limited to approximately 1800 records. With the patches and shortened segments I have been able to create a new file which will open to approximately 3000 records. However, I have not been able to successfully transfer the records from my old file to the new file. How do I do it? In my old file I have used the second and third segments but have limited the totals of the field lengths to 128 bytes."

What is needed is a short BASIC program that will write the old 256 byte records into 128 byte records. This entails taking the first 128 bytes from the old records and writing them into 128-byte records in a new file.

Reading PROFILE files from BASIC was described in the November 1981 column. Using that information as a jumping-off point, you should be able to use this sample program to write the old to the new file:

At TRSDOS READY, open 2 files in BASIC: BASIC -F:2
Type in the following program:

```
100 OPEN "D",1, "FILENAMEEXT
: n "
110 OPEN "D",2, "FILENAMEEXT
: n ",128
120 FIELD 1,128 AS IN$
130 FIELD 2,128 AS OT$
140 FOR R=1 TO LOF(1)
150 GET 1,R
160 LSET OT$=IN$
170 PUT 2,R
180 NEXT R
```

In place of "FILENAME" use the name of your file; if the name is less than 8 characters, pad it with zeroes—i.e., "SHORT000/EXT."

PROFILE data file extensions ("EXT") are /DAT for segment 2; /DA2 for segment 3; and /DA3 for segment 4. To move segments 2 and 3, Mr. Lynch will have to type in the program twice, using FILENAME/DAT the first time, and FILENAME/DA2 the second time.

Note: You cannot move the KEY—the first segment—with this program because key segments are written in blocks

of 3 (1 from each of 3 records), and un-blocking and re-blocking the segments is quite another process. However, in the above example, the old key segment can be used for the new file, too.

The "n" at the end of the extension is the drive number. If you have only one drive, or want to write the old file into the new file on the same drive, your new file name must be different from the old file name. After the records are written into the new file, change the new name to the old name using the TRSDOS "RENAME" command.

Once you've verified that the records have been written successfully into the new file, you can kill the old file. If the old and new files have the same name, we suggest you take the diskette(s) holding the new file out of the system first, to avoid killing it by mistake.

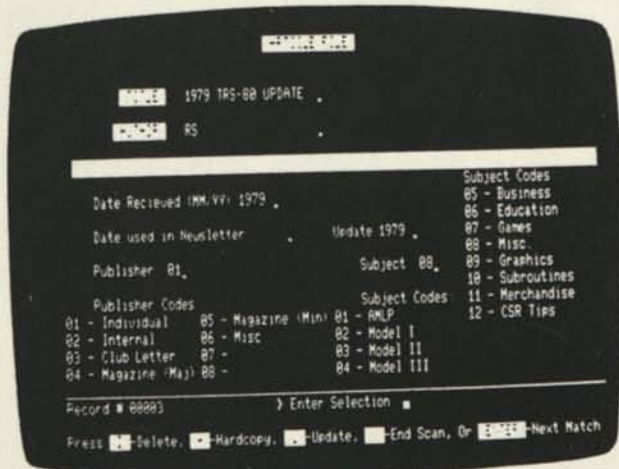
One question we often hear from PROFILE II and PROFILE PLUS users is, "What **are** all those programs on my diskette—what do they do?"

To answer this question once and for all—until we write more add-on programs, that is—we've come up with the following table.

Included are PROFILE II, PROFILE PLUS. Where you see "FILENAME," substitute your own file's name.

Program	Menu Choice & Function
PROFILE II	
M	PROFILE II menu
CREATION PROGRAMS	
CREATE/EFC	Menu choice 1—define files; creates FILENAME/MAP, KEY, DAT, DA2, DA3
CREATEX/EFC	Menu choice 2—define screens; creates FILENAME/PMP, PM1-5
LPFORM/EFC	Menu choice 3—define report formats; creates FILENAME/PRT, PR2-5

LBFORM/EFC	PROFILE II menu choice 4—define label formats; creates FILENAME/LB1-5
FIELDER/EFC	PROFILE II menu choice 5—define selections for SCRIPSIT; creates FILENAME/SL1-5 (SL = Selection List)
RUNTIME PROGRAM	
EXPAND/EFC	PROFILE II menu choice 6—expand files FILENAME/KEY, DAT, DA2, DA3; in single-segment files only, creates FILENAME/KX1-3.
CLERK/EFC	PROFILE II menu choice 7, Inquire, Update, Add; uses the indexes; uses screen formats FILENAME/PMP, PM2-5
PRINT/EFC	PROFILE II menu choice 8, print reports; reads in FILENAME/PRT, PR2-5
LABEL/EFC	PROFILE II menu choice 9, print labels; uses FILENAME/LB1-5
SELECTOR/EFC	PROFILE II menu choice 10, run record selection for SCRIPSIT; reads in FILENAME/SL1-5 and creates FILENAME/SR1-5 (SR = Select Records)
PROFILE PLUS	
MENUPLUS/EFC	The PROFILE PLUS menu
MAKEMENU/EFC	PROFILE PLUS menu choice 1—create a user's menu
K/EFC	A key file; it allows you to pass parameters to programs in a user's menu
CREATEM/EFC	PROFILE PLUS menu choice 2—define math formulas; creates FILENAME/MTH
INDEX/EFC	PROFILE PLUS menu choice 3—build index; creates FILENAME/IX1




New Journal Calls For Papers

Gerald H. Block
A LEARNING PLACE
6472 Moraga Avenue
Oakland, CA 94611

A new quarterly, THE JOURNAL OF COMPUTERS READING & LANGUAGE ARTS (CRLA), is ready to receive papers. The journal's purpose is to support the rapidly growing interest in computers and their relationship to reading, language arts and related issues. The theme of the journal will be pragmatic in perspective. It will emphasize presenting papers which have clear classroom/teaching implications.

The journal will be interdisciplinary and directed toward an audience of reading-language arts teachers, educational specialists, classroom teachers, educators of teachers and educational researchers.

Papers or requests for information should be sent to Gerald H. Block, CRLA, P.O. Box 13039, Oakland, CA 94661. 

Radio Shack Agricultural Software Sourcebook

Radio Shack invites you to list your agricultural computer software in our new Agricultural Software Sourcebook.

Listing your software in the TRS-80 Agricultural Software Sourcebook could be the most effective advertising expenditure you will make in 1982: a listing will introduce a description of your programs to the owners of thousands of TRS-80 computers in the agricultural market, through the more than 6,000 Radio Shack Stores which will sell the sourcebook. No one else can make you an offer like this one.

To obtain a single listing in the forthcoming Radio Shack Agricultural Software Sourcebook (26-2774) for a period of one year, complete the registration form, submit a \$10.00 check or money order to "Radio Shack", and mail them to the following address:

RADIO SHACK AGRICULTURAL SOFTWARE SOURCEBOOK
DEPARTMENT AX10—ONE TANDY CENTER
FORT WORTH, TEXAS, 76102

Registration forms can be obtained by writing department AX10 at this address. If you wish to list ten or more of your agricultural programs in the Radio Shack Agricultural Software Sourcebook, you can do so for only \$5.00 per listing. This is a savings of 50% per program listing off the standard fee of \$10.00 per listing.

To accept our special listing offer, just send 10 or more separate forms together with a check or money order for all the listings at the \$5.00 per listing rate.

We invite you to join us by listing your agricultural software. The deadline for the first edition is November 30, 1982.

Planetarium

J.W. Hurley
3403 Onyx Street
Torrance, CA 90503

Have you ever wondered what the stars looked like but did not want to go out in the rain? Have you ever wondered which planet is visible and where? Have you ever wondered where Sirius is located? Have you ever wondered why I do not get on with it?

Here are some of the keys to the program.

Line 20	Insert your own value for LA based on your latitude. This runs from about 30 in the Southern United States to around 50 in the North. Not critical.
Line 50	Input either a carriage return or a three letter abbreviation of a constellation. Example: enter "SCO" and all the stars of Scorpius will be shown over-size.
Line 60	As above except the individual stars named will be accented. If "SUN" is entered part two will draw the sun and accent the planets.
Line 80	Prints instructions of what to do after the chart is complete.
Line 160	Converts date into days into the year. Lines 270-190 chew up the rest of the data so the next read starts with the star data.
Lines 220-790	Four part data statements, first the constellation name, then the star name if available, or a "Z" for a null, then the right ascension in hours and finally the north-south elevation. This information is taken from reference books and more stars can be included. We have named the 21 first magnitude stars and included most second magnitude stars as unnamed ones.
Line 790	When data "XXX" is read the program knows the last information has been read.
Lines 820-840	Draws a cross that is directly overhead and a circle to indicate the horizon and labels East and West.
Lines 850-950	Reads data, plots position based on an arc around the North Star. PSETs a point. "T" is a factor that increases star time four minutes each day into the year as compared to solar time which we use.
Lines 970-980	Accent stars as requested at the start of the program.

Line 990	Says go back and read another set of data if "XXX" has not been read.
Line 1000	Says go do the solar system and print the data.
Line 1020	After all the above has been done the computer waits for your instruction. There is no prompt.
Line 1010	Just in case you wish to redraw at a later date or hour the program chews up the first data set so it can reread the star data.
Lines 1110-1250	Prints the time and sounds a signal that all is done as requested.
Lines 1260-1290	Plots how far above or below the celestial equator the sun is for the time of year. This factor, "E," is also used to locate the planets. They are not all right on the line of the sun but are very close. If all is right the sun will be all the way North about Jun. 22 and South for winter December 22.
Line 1290	Locates the sun by the hour and then goes sub to plot x-y coordinates again based on an arc as the stars were.

The balance of the program reads data based on the number of months into the time from 1-1-82. If a date outside the 82-85 requested this part is skipped.

Data is read for the middle of each month. Only in a few cases is this far off. Note that the data for VA (Venus) is given in degrees from the sun while the slower planets use the same right ascension as the stars used earlier.

I hope other readers enjoy improving the program and adding to it as much as I have enjoyed developing it almost line by line. I see quite a few areas for improvement but you have to stop somewhere. It shows that the CC is good for many things beside making pretty patterns.

I have had my setup about a year and have taught it many things from playing Blackjack to figuring income tax. I am retired and dollars are scarce but this computer has been well worth the money in enjoyment received. I hope you will print many different kinds of programs to spark interest.

```
10 'ENTER VIEWING LATITUDE FOR LA
20 LA=33
30 CLS
40 PRINT
   : PRINT
   : PRINT "          STAR MAPS"
   : PRINT
   : PRINT
```

```

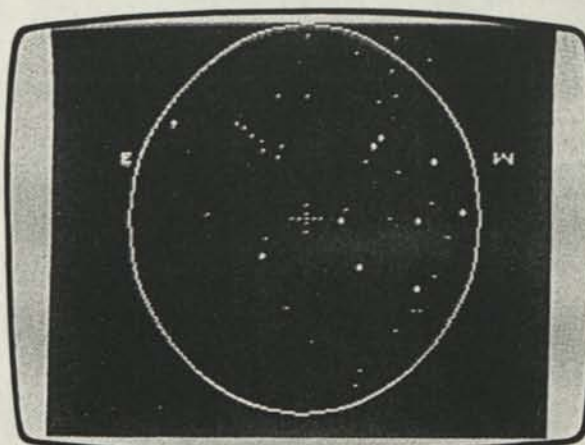
50 PRINT "TO ACCENT ANY CONSTELLATION TYPE
   IT'S FIRST THREE LETTERS"
   : INPUT E$
60 PRINT "TO ACCENT A STAR-TYPE IT'S FIRST THREE
   LETTERS. TYPE <SUN> TO ACCENT SOLAR
   SYSTEM."
   : INPUT C$
70 C$=RIGHT$(C$, 3)
80 PRINT "AFTER CHART IS DISPLAYED HIT SPACE
   BAR TO ADVANCE ONE HOUR <E> TO DRAW
   CELESTIAL EQUATOR <D> TO ENTER NEW DATE
   <Q> TO QUIT
90 INPUT "DATE(MM,DD,YY (82-85))"; M, D, YY
100 IF M<0 OR M>12 THEN 90
110 IF D<0 OR D>31 THEN 90
120 INPUT "CHART TIME-USE STANDARD TIME AND 24
   HOUR FORMAT TO NEAREST HOUR."; H
130 H=INT(H)
   : DY=0
140 IF H<0 OR H>24 THEN 120
150 DATA 0, 31, 28, 31, 30, 31, 30, 31, 31, 30,
   31, 30, 31, -99
160 FOR J=1 TO M
   : READ DD
   : DY=DY+DD
   : NEXT
170 READ Z
180 IF Z=-99 THEN GOTO 200
190 GOTO 170
200 DY=DY+D
210 PMODE 4, 1
   : PCLS
   : SCREEN 1, 1
220 DATA "ERI", "ACH", 1.6, -57
230 DATA "UMA", "Z", 10.8, 61.8
240 DATA "UMA", "Z", 10.8, 57
250 DATA "UMA", "Z", 11.7, 54
260 DATA "UMA", "Z", 12.1, 57
270 DATA "CAS", "Z", .6, 56.4
280 DATA "CEP", "Z", 21, 62.5
290 DATA "GRU", "Z", 22.6, -47
300 DATA "AUR", "Z", 5.9, 45
310 DATA "CMA", "Z", 6.8, -26.8
320 DATA "CRU", "ACR", 12.4, -63
330 DATA "UMA", "Z", 12.7, 55
340 DATA "UMA", "Z", 13.3, 53
350 DATA "UMA", "Z", 13.6, 50
360 DATA "TAU", "ALD", 4.6, 16
370 DATA "AQU", "ALT", 19.8, 9
380 DATA "SCO", "ANT", 16.5, -26
390 DATA "BOO", "ARC", 14.2, 19
400 DATA "CRU", "BEC", 12.7, -60
410 DATA "ORI", "BET", 5.9, 7
420 DATA "CAR", "CAN", 6.4, -63
430 DATA "AUG", "CAP", 5.2, 46
440 DATA "CYG", "DEN", 28.7, 45
450 DATA "FAU", "FOM", 22.9, -30
460 DATA "CNE", "HAD", 14, -60
470 DATA "GEM", "POL", 7.7, 28
480 DATA "CMI", "PRO", 7.6, 5
490 DATA "LEO", "REG", 10.1, 12
500 DATA "ORI", "RIG", 5.2, -8
510 DATA "CEN", "RIG", 14.6, -61
520 DATA "CMA", "STR", 6.7, -17
530 DATA "VIR", "SPI", 13.4, -11
540 DATA "LYR", "VEG", 18.6, 39
550 DATA "PEG", "Z", .1, 29
560 DATA "PEG", "Z", .1, 15
570 DATA "PEG", "Z", 23, 29
580 DATA "PEG", "Z", 23, 15
590 DATA "UMI", "Z", 1, 90
600 DATA "UMI", "Z", 14.7, 74
610 DATA "PHO", "Z", 4, -42, "CET", "Z", .4, -18,
   "AND", "Z", 1, 35, "AND", "Z", 2, 42,
   "ARE", "Z", 2, 23, "CET", "Z", 2.3, -3
620 DATA "ORI", "Z", 5.7, -9.6, "AUR", "Z", 5.9,
   45, "CMA", "Z", 6.3, -18, "GEM", "Z", 6.4,
   16.4, "CMA", "Z", 6.9, -26, "CMA", "Z", 7.3,
   -29
630 DATA "PER", "Z", 3, 41, "PER", "Z", 3.3, 50,
   "ORI", "Z", 5.4, 6.3, "TAU", "Z", 5.4, 28.6
640 DATA "GEM", "Z", 7.4, 32, "PUP", "Z", 8, -40,
   "VEL", "Z", 8.1, -47
650 DATA "HYD", "Z", 9.3, -8.6, "LEO", "Z", 10.1,
   20, "LEO", "Z", 11.8, 14.7, "CRU", "Z",
   12.4, -57, "CEN", "Z", 12.7, -49, "CEN",
   "Z", 13.4, -53
660 DATA "CEN", "Z", 14.1, -36.2, "CEN", "Z",
   14.5, -42, "LUP", "Z", 14.6, -47, "BOO",
   "Z", 14.6, 27
670 DATA "CBO", "Z", 15.5, 26.3, "SCO", "Z",
   15.9, -22.5, "SCO", "Z", 16.7, -34.2,
   "OPH", "Z", 17.1, -15.7, "SCO", "Z", 17.5,
   -37, "OPH", "Z", 17.5, 12.5
680 DATA "SCO", "Z", 17.5, -43, "SCO", "Z", 17.6,
   -39, "DRA", "Z", 17.7, 51.5, "SAG", "Z",
   18.3, 67, "SAC", "Z", 19.2, -26
690 DATA "CYG", "Z", 20, 40, "PAY", "Z", 20.4,
   -56, "CYG", "Z", 20.6, 33.9, "PEG", "Z",
   21.6, 9.7, "GRU", "Z", 22.3, -47
700 DATA "PER", "Z", 3.3, 50
710 DATA "CAS", "Z", .1, 58
720 DATA "CAS", "Z", .5, 55
730 DATA "CAS", "Z", 1, 60
740 DATA "CAS", "Z", 1.4, 58
750 DATA "CAS", "Z", 1.9, 62
760 DATA "ORI", "Z", 5.5, 0
770 DATA "ORI", "Z", 5.65, -2
780 DATA "ORI", "Z", 5.8, -4
790 DATA "XXX", "XXX", 0, 0
800 FOR J=120 TO 136 STEP 4
   : PSET(J, 90, 1)
   : NEXT
810 FOR J=84 TO 98 STEP 4
   : PSET(128, J, 1)
   : NEXT
820 CIRCLE(128, 90), 90
830 DRAW "BM34, 60R4L4D2R2L2D2R4"
840 DRAW "BM226, 60; D4E4F4U4"
850 READ F$, B$, A, B
860 IF B$="XXX" THEN 1000
870 A=24-A
880 T=DY/15+H
   : A=A+T+12
   : B=ABS(B-90)
890 IF A>24 THEN A=A-24
   : GOTO 890
900 A=A*15
910 Y=SIN(A/57.296)*B+LA
920 X=128-(COS(A/57.296)*B)
930 IF Y<0 OR Y>180 THEN GOTO 850
940 IF X<38 OR X>218 THEN 990
950 PSET(X, Y, 1)
960 IF B$<"Z" THEN CIRCLE(X, Y), 1
970 IF F$=E$ THEN CIRCLE(X, Y), 1
980 IF B$=C$ THEN CIRCLE(A, B), 2
990 GOTO 850
1000 GOSUB 1260
   : GOSUB 1110
1010 RESTORE
   : FOR J=1 TO 14
   : READ Z
   : NEXT
1020 I$=INKEY$
   : IF I$="Q" THEN END
1030 IF I$="E" THEN CIRCLE(128, LA), 90, 1, 1,
   .05, .45
   : GOTO 1020
1040 IF I$="D" THEN RESTORE
   : GOTO 90

```

```

1050 IF I$="" THEN 1020
1060 PCLS
      : H=H+1
      : GOTO 790
1070 IF H=7 OR H=17 THEN DRAW "BM200, 150; R8G12"
1080 IF H=8 OR H=18 THEN DRAW "BM200, 150;
      R8D6L8D6R8U6 L6U6"
1090 IF H=9 OR H=19 THEN DRAW "BM200, 150;
      R8D12U6L8U6"
1100 IF H=10 OR H=20 THEN DRAW "BM200, 150;
      R8D12L8U12"
1110 IF H>24 THEN H=N-24
      : DY=DY+1
1120 IF H=1 OR H=11 OR H=21 THEN DRAW "BM200,
      150; D12"
1130 IF H=2 OR H=12 OR H=22 THEN DRAW "BM200,
      150; R8D2G6 D4R8"
1140 IF H=3 OR H=13 OR H=23 THEN DRAW "BM200,
      150; U2R8D6 L4R4D6L8U2"
1150 IF H=4 OR H=14 OR H=24 THEN DRAW "BM200,
      150; D6R12L6 U6D12"
1160 IF H=5 OR H=15 THEN DRAW "BM200, 150;
      R8L8D6R8D6L8 U2"
1170 IF H=6 OR H=16 THEN DRAW "BM200, 150;
      R6L6D12R8U6L6"
1180 IF H=7 OR H=17 THEN DRAW "BM200, 150; R8D12"
1190 IF H=8 OR H=18 THEN DRAW "BM200, 150;
      R8D6L8D6R8 U6L8U6"
1200 IF H=9 OR H=19 THEN DRAW "BM200, 150;
      R6D6L6U6R8 D12"
1210 IF H=10 OR H=20 THEN DRAW "BM200, 150;
      R8D12L8U12"
1220 IF H>9 AND H<20 THEN DRAW "BM180, 150; D12"
1230 IF H>19 THEN DRAW "BM180, 150; R8D2G6D4R8"
1240 SOUND 200, 8
1250 RETURN
1260 Y1=YY-82
1270 DE=DY+9
      : IF DE>365 THEN DE=DE-365
1280 DE=DE/57.296
      : E=COS(DE/365*360)*23
1290 A=H-6
      : IF H<6 OR H>18 THEN SCREEN 1, 1
      : GOTO 1330
1300 IF C$<>"SUN" THEN 1330
1310 GOSUB 1570
1320 SCREEN 1, 0
      : CIRCLE(X, Y), 4
      : PAINT(X, Y), 1, 1
1330 IF YY<82 OR YY>85 THEN RETURN
1340 MM=M+Y1*12
1350 FOR Z=1 TO MM
      : READ YA, MA, JA, SA
      : NEXT
1360 IF YA=0 THEN 1400
1370 A=VA/15+(H-6)
      : GOSUB 1570
      : GOSUB 1630
      : IF Y<LA THEN GOTO 1390
1380 LINE(X, 182)-(X+2, 188), PSET
      : LINE-(X+4, 182), PSET
1390 IF MA=0 THEN 1430
1400 A=MA+T
      : GOSUB 1570
      : GOSUB 1630
      : IF Y<LA THEN GOTO 1420
1410 LINE(X, 188)-(X, 182), PSET
      : LINE-(X+2, 188), PSET
      : LINE-(X+4, 182), PSET
      : LINE-(X+4, 188), PSET
1420 IF JA=0 THEN 1460
1430 A=JA+T
      : GOSUB 1570
      : GOSUB 1630
      : IF Y<LA THEN GOTO 1450
1440 LINE(X+2, 182)-(X+2, 188), PSET
      : LINE-(X-2, 188), PSET
      : LINE-(X-2, 184), PSET
1450 IF SA=0 THEN RETURN
1460 A=SA+T
      : GOSUB 1570
      : GOSUB 1630
      : IF Y<LA THEN RETURN
1470 LINE(X+2, 182)-(X-2, 182), PSET
      : LINE-(X+2, 188), PSET
      : LINE-(X-2, 188), PSET
1480 RETURN
1490 DATA 0, 12, 8, 14, 4, 13, 4, 33, 13, 2,
      14.5, 13.4, 45, 13, 14.5, 13.3, 46, 12.4,
      14.3, 13.2, 42, 12.1, 14.1, 13.1, 36, 12.4,
      13.9, 13
1500 DATA 29, 13.2, 13.9, 13, 21, 13.9, 14.1,
      13.2, 13, 15.6, 14.4, 13.4, 0, 17.1, 14.8,
      0, 0, 18.8, 0, 13.8, -11, 20.4, 15.7, 14
1510 DATA -18, 22, 16.1, 14.1, -25, 23.5, 16.5,
      14.2, -31, .8, 16.6, 14.1, -31, 2.3, 16.6,
      14, -43, 0, 16.4, 13.9, -45, 0, 16.1, 13.8,
      -41, 6.8, 16, 13.8, -15, 8.2, 16, 13.9, 29,
      9.6, 16.2, 14
1520 DATA 44, 10.7, 16.5, 14.2, 46, 11.9, 16.9,
      14.5, 42, 13, 0.14.7
1530 DATA 37, 14, 17.9, 14.9, 31, 15, 18.4, 15,
      24, 15.6, 18.7, 15, 15.7, 18.9, 14.9,
      0, 15.1, 18.9, 14.7, 0, 14.6, 18.7, 14.6
1540 DATA 0, 14.8, 18.5, 14.5, -16, 15.7, 18.2,
      14.6, -25, 17, 18.2, 14.7, -32, 18.5, 18.5,
      14.9, -39, 20.1, 18.8, 0, -44, 21.7, 19.3,
      15.4
1550 DATA -47, 23.1, 0, 15.6, -45, .6, 20.3,
      15.7, -28, 1.8, 28.7, 15.8, 18, 3.3, 21,
      15.7, 41, 4.7, 21.3, 15.5, 46, 0, 21.3, 15.4
1560 DATA 43, 0, 21.1, 15.3, 38, 0, 20.9, 15.3,
      30, 10.4, 20.7, 15.4, 23, 11.1, 20.7, 15.6,
      16, 12.7, 20, 8, 0, 0, 13.9, 21.2, 16.1
1570 B=90
      : A=A*15
1580 X=128-(COS(A/57.296)*B)
1590 IF X<38 OR X>218 THEN RETURN
1600 Y=SIN(A/57.296)*B
1610 Y=Y+LA+E
1620 RETURN
1630 IF X<38 OR X>218 THEN RETURN
1640 IF Y<LA THEN RETURN
1650 CIRCLE(X, Y), 1
      : IF C$="SUN" THEN CIRCLE(X, Y), 2
      : RETURN
1660 RETURN

```



TRS-80®

Volume 4, Issue 9

October, 1982

Price \$1.50

Microcomputer News

Information Published for TRS-80 Users.

Data Base Issue
Model I Double Density
Color Graphic Printer
Micro Research





In addition to the software and hardware necessary to make your TRS-80® computer operate like a terminal, Radio Shack now offers three separate terminals which are available at Radio Shack Computer Centers or the expanded computer departments of selected Radio Shack stores and dealers.

VIDEOTEX

The Videotex terminal requires the addition of a phone line and television to enable you to access national information networks like Dow Jones and CompuServe. It is the least expensive of the three Radio Shack terminals (16K for \$499.00).

PT-210 PORTABLE DATA TERMINAL

The new TRS-80 PT-210 Portable Data Terminal (76-1001, \$995.00) incorporates a full "typewriter" keyboard, a quiet thermal printer and a 110/300 baud (Bell 103A compatible) acoustic telephone coupler in a compact, attractive, portable package.

It provides exceptional value for applications where hard copy of in-computer information or information provided by Videotex or other on-line services is needed. An optional add-on RS-232-C Interface Module (76-1002) for \$69.95 can be user (plug-in) installed. This facilitates use of the terminal as a local (to a computer or selected peripheral) "front end" dumb terminal and/or printer. RS-232-C is also a convenient way to connect the terminal to a direct connect modem.

The PT-210 features a full-size ASCII keyboard, generating a total of 99 codes. 110 baud or 300 baud operation may be switch selected, as may half-duplex or full-duplex operation and odd-parity/even-parity/no-parity modes.

Its quiet non-impact thermal printer uses a 35-element (5 x 7) matrix and offers variable contrast control. 71 characters are printable, with lower case letters automatically printed as their upper case equivalents. Each 8-inch line can include up to 80 characters (10 characters per inch), and carriage return is automatic at the 81st column on any line. Printing speed is 50 characters per second with 6 lines per vertical inch on 100-foot rolls of 8 1/2-inch-wide thermal paper.

The PT-210 is housed in a sleek silver-grey case measuring 14 1/2 x 14 1/2 x 5 inches and weighing 15 pounds with paper installed. It is FCC registered and UL-listed for 120 VAC 60 Hertz operation. The line cord is detachable for easy portability.

DATA TERMINAL 1

Radio Shack also recently announced TRS-80 DT-1 (26-6050, \$699.00) which is an advanced video data terminal. The DT-1 can be configured to emulate any of four popular terminals: the Televideo 910, Lear-Siegler ADM-5, ADDS 25 or Hazeltine 1410. Its configuration is keyboard selectable (other multiple-protocol terminals use tiny DIP-switches for this selection which is often an awkward proce-

dure). The selected protocol is maintained independently of power through the use of a breakthrough in memory technology: the EEPROM (Electrically Erasable Programmable Read-Only Memory).

The custom-programmed, high speed 8-bit micro-processor inside the DT-1 is the basis of its high performance, flexibility, and low price.

The legibility and flexibility of the DT-1 display are remarkable. The 12-inch (diagonal measure) CRT provides a full 80 character by 24 line (80 x 24) video display. Upper-(shift key) and lower-case characters are available, and control characters can be displayed without using special symbols. A variety of video attributes are available, including normal, reverse, invisible, blinking, underlined, and half-intensity video. And four types of cursors may be selected from the keyboard—steady or blinking, block or underline. The DT-1's 70-key keyboard has been designed for smooth, convenient operation with full, ergonomic keyboard "feel". It includes the full set of special keys used by the TRS-80 Model II and Model 16 computers.

In addition to a standard typewriter-style keyboard area, the DT-1 has a 12-key data pad which functions in three ways. It can be used calculator-style for quick entry of numbers and the ENTER command; it can be used with the SHIFT key for cursor up, cursor down, cursor left, cursor right and home commands, plus special characters; and it can be used with the CONTROL key for control code commands. Any keystroke can be repeated automatically by holding down the desired key; after the first several repeats, the DT-1 will automatically repeat at a faster rate.

The TRS-80 DT-1 is equipped with both a built-in RS-232-C serial interface and a Radio Shack printer-compatible parallel interface, either of which may be used as a printer port for hard copy output. A local monitor mode can be used to enable printer ports and to position the video cursor using Local Control and Escape modes. The CARRIAGE RETURN key can be programmed to operate with or without sending a Line feed command. CONTROL key termination codes are also user-programmable. Ten keyboard selectable communications rates from 75 to 19,200 baud are available. All keyboard selectable options are stored in the EEPROM, and they may be changed or temporarily overridden from the keyboard at any time. An electronic bell is standard. The DT-1 is U.L. listed for 120 VAC 60 Hz operation; a 50 Hz option is also available.



The very portable Radio Shack PT-210—the newest in our line of data terminals—includes many outstanding features.

TRS-80 Microcomputer News: 1982
Tandy Corporation,
Fort Worth, Texas 76102 U.S.A.
All Rights Reserved

Reproduction or use, without express written permission from Tandy Corporation of any portion of the Microcomputer News is prohibited. Permission is specifically granted to individuals to use or reproduce material for their personal, non-commercial use. Reprint permission for all material (other than William Barden's article), with notice of source, is also specifically granted to non-profit clubs, organizations, educational institutions, and newsletters.

TRS-80 Microcomputer News is published monthly by Radio Shack, a division of Tandy Corporation. A single six month subscription is available free to purchasers of new TRS-80 Microcomputer systems with addresses in the United States, Puerto Rico, Canada and APO or FPO addresses. Subscriptions to other addresses are not available.

The subscription rate for renewals and other interested persons with U.S., APO or FPO addresses is twelve dollars (\$12.00) per year, check or money order. Single copies of the Microcomputer News may be purchased from Radio Shack Computer Centers or Computer Departments for \$1.50 suggested retail each. The subscription rate for renewals and other interested persons with Canadian addresses is fifteen dollars (\$15.00) per year, check or money order in U.S. funds. All correspondence related to subscriptions should be sent to: Microcomputer News, P.O. Box 2910, Fort Worth, Texas 76113-2910.

Retail Prices in this newsletter may vary at individual stores and dealers. The company cannot be liable for pictorial and typographical inaccuracies.

Back issues of Microcomputer News prior to January, 1981 are available through your local Radio Shack store as stock number 26-2115 (Suggested Retail Price \$4.95 for the set). Back issues of 1981 copies are not available.

The TRS-80 Newsletter welcomes the receipt of computer programs, or other material which you would like to make available to users of TRS-80 Microcomputer systems. In order for us to reprint your submission, you must specifically request that your material be considered for reprinting in the newsletter and provide no notice that you retain copyrights or other exclusive rights in the material. This assures that our readers may be permitted to recopy and use your material without creating any legal hassles.

Material for publication should be submitted on magnetic media (tape, disk, or CompuServe). If you submit material on tape or disk, and it is accepted for publication, we will send you two cassettes or diskettes for each one you sent us. Cassettes will come from our box of mixed blank cassettes. If you submit material on CompuServe, and we think we may use the material, we will extend your Microcomputer News subscription by six months for each article accepted. If you are submitting material over CompuServe, please include your name and address or your subscription number so we can find you. If the material is very short, send it to us in a mail. If you have more than a few lines, you need to place the material in the ACCESS area of CompuServe and then let us know it is there by leaving a message on E-Mail.

Material may be submitted by mail to P.O. Box 2910, Fort Worth, Texas 76113-2910, or through CompuServe. The Microcomputer News' CompuServe user ID number is 70007.535.

Notes to Program Users:

Programs published in the Microcomputer News are provided as is, for your information. While we make reasonable efforts to ensure that the programs we publish here work as specified, Radio Shack can not assume any liability for the accuracy either of the programs themselves, or of the results provided by the programs.

Further, while Microcomputer News is a product of Radio Shack, the programs and much of the information published here are not Radio Shack products, and as such can not be supported by our Computer Customer Service group. If you have questions about a program in the Microcomputer News, your first option is to write directly to the author of the program. When possible, we are now including author's addresses to facilitate communications. If the address is not published, or if you are not happy with the response you get, please write us here at Microcomputer News. We will try (given the limited size of our staff) to find an answer to your question and, in many cases, will publish the answer in an up-coming issue of Microcomputer News.

Trademark Credits

CompuServe [™]	CompuServe, Inc.
Dow Jones [™]	
NEWS/RETRIEVAL	
Service [*]	Dow Jones & Co., Inc.
Time Manager [™]	Image Producers
VisiCalc [™]	VisiCorp, Inc.
Program Pak [™]	Tandy Corporation
SCRIPSIT [™]	Tandy Corporation
TRSDOS [™]	Tandy Corporation
TRS-80 [*]	Tandy Corporation

Microcomputer News

Information Published for TRS-80 Users.

Contents:

Assembly Language Programming	5
by William Barden, Jr.	
Corrections to June Barden Article	9
Color Computer	
Programs	
Hex to Decimal	10
by C. Thonen	
Landline	10
by R.M. Hallam	
Loadcopy and Loaddata	44
by G. Fraser	
Mailing Labels on CoCo	46
by D. Banaszak	
Print to the Screen and Wraparound	10
by G. Plueard	
Computer Clubs	17
Computer Customer Service	
Data Base Management Systems	11
Data Bases	
CompuServe	13
Welcome to Someday	
Dow Jones	18
The Academic American Encyclopedia—A to Z Electronically	
Profile	25
Profile III + : Introducing the Flexible Data Base Management System	
Education	
Introducing "Number Theory"	32
Microcomputer-Based Data Management for Small Libraries	32
microResearch Problems	27
Fort Worth Scene	2
General Interest	
Coded Message	21
by P. Vogel	
Heart Health Programming Contest	24
Ode to Micro-Idiom	43
by J. Carr	
Scripsit Ideas	21
by W. Ashworth	
Model I/III	
Bugs, Errors, and Fixes	
COBOL Development System (26-2203)	31
Model III Disk System Manual (26-2111)	31
Programs	
A Baker's Dozen of Helpful Hints	30
by J. Rogers	
Model I Double Density	28
Model II	
Bugs, Errors, and Fixes	
Scripsit 2.0 (26-4531)	37
Programs	
A Patch Writing Program	35
by R. Burke	
The Firmament	38
by W. Boyd	
Notes on Previous Newsletters	4
June, 1981 SAFE	
January, 1982	
One Armed Dime Bandit	
Shoot 'em Again	
Lines	
February, 1982	
Joystick Draw Routine	
Peripherals	
Color Graphic Printer	22
Pocket Computer	
Bugs, Errors, and Fixes	
Calendar (26-3529)	43
Programs	
Data File Program	39
by A. Barclay	
Pearson (R) Correlation	43
by G. Currah	
Reverse Polish Notation on the Pocket Computer	40
by J. Morrison	

Prices shown in TRS-80 Microcomputer News are in U.S. Funds.

Notes on Previous Newsletters

JUNE 1981 SAFE

In the SAFE program for the instructions indicate that the Left joystick will be used when the right joystick is assigned in line 110 A=JOYSTK(0). Change the word left to right in line 50.

DECEMBER 1981 PATCH and SPELLING CHECKER

Gilbert & Kathy Keith
Route 1, Box 285
Rineyville, KY 40162

It is with reading your article on page 31, PATCH, that really gave me the idea that possibly I should do some kidding about your article on page 30. On page 31, THE PATCH states that a vocabulary has surfaced that discovers the more serious types of bugs, such words as: bomb, blow up, silent death, hang, crash, limbo. I would like to add a few more of my own to this most serious type of problem, such as: little goof, goof, big goof, and big splash.

On page 30 SPELLING CHECKER, whoever wrote this article made a big splash since the dictionary for Model III is not out yet, I cannot speak for it, but I can speak for the dictionary for the Model II. The statement that was made in the article was "and it will be checked for spelling errors at about 2000 words per minute." The most that we have ever been able to check is about 800 to 900 words per minute. This is with SCRIPSIT, Version 2.0 and the new dictionary.

The reason for bringing this to your attention is that we are proud owners of both the Model II and the Model III computers, and we do not want erroneous information put out that would mislead the public.

We are your supporters, and we do not want you to take this as a personal affront. Our Radio Shack computers hold a warm spot in our heart, and we appreciate the good work that you are doing.

Editor's Note: I just spell checked a document containing 2757 words. There were 99 possible errors in the file, and dictionary took 1 minute 33 seconds to go through the file. That figures out to 1779 words/min. The speed depends on a lot of factors. I have had files checked at 2000+ word/minute, but the average is somewhat slower, as the Keiths point out.

JANUARY 1982 ONE-ARMED DIME BANDIT

Newell H. Claudy
1811 Filbert Way, Apt. A
Elkhart, IN 46514

This program suggests a clever application of the Pocket Computer. Unfortunately, the program won't run correctly the way it is listed.

Variables R,S, and T must be reset to zero each time before entering the FOR-NEXT loop. Otherwise, values accumulated in line 95 will be meaningless. The necessary correc-

tion to the program can conveniently be made by adding to line 50, resulting in:

```
50 M=M-.1
   : V=0
   : R=0
   : S=0
   : T=0
```

Delete the expressions R=0, S=0, and T=0 in line 30, where they function only the first time the program is run.

The display is esthetically more attractive if the value of A\$(30) specified in line 20 is changed to have five dots, rather than three, corresponding to the five letter since the words "APPLE", "GRAPE", and "LEMON". This keeps spacing of the display the same while the "drums spin."

One final comment. To be fair to the player, the payback for a win should be 90 cents instead of 60 cents or M = M + .9 in line 130. The true probability of winning is 1/9, so out of nine tries (on the average) the player should spend 90 cents to play, getting back 90 cents on one "spin." The 60 cent payback yields a "house" percentage of 33.3% (the "house" keeps 30 cents out of each 90 cents the player spends). This one-armed bandit lives up to its name!

Donald Parson
Apt. 43-C
4475 N. Ocean Blvd.
Delray Beach, FL 33444

Mr. Bond is to be congratulated on his ingenious and elegant program.

I did however find that it was paying off when it shouldn't and vice versa.

I fiddled around for awhile and finally made the following changes:

```
95 Delete
125 Q=(H=J)+(H=K)+(J=K)
130 If Q=3 etc.
```

Line 125 now tests directly for 'three-of-a-kind'.

SHOOT 'EM AGAIN


Levin J. Muth
700 First Avenue
Harvey, LA 70058

I am dropping this line to make minor corrections to J.W. Myers article.

```
Change
110 OT=T
Delete
112
Add
384 IF W=A6 THEN 500
```

LINES

Mrs. Fern Christensen
Northwestern State University of Louisiana
1017 Oma Street
Natchitoches, LA 71457

Several have written about making lines, as teachers are so inclined to do for test making. My favorite is to have tabulation "t" set at appropriate spacing (i.e. such as 10 or 15 spaces). Use F1 SHIFT HYPHEN and tabulate or arrow to the place chosen to stop, SHIFT HYPHEN, and you will have your line drawn for a student to fill in "with the correct answer." 

The 6809 Divides and Conquers

by William Barden, Jr.
©William Barden, Jr. 1982

One of the problems with assembly language is that it's even more basic than BASIC. In BASIC you can compute numbers with relative ease, spewing integers, fractions, and exponents all over your listings. The BASIC interpreter handles adds, subtracts, multiplies, and divides with ease, and one takes the computing power for granted.

THE 6809E MULTIPLY

Not so with assembly language. All arithmetic operations are done at rudimentary levels, and you're lucky to be able to handle adds and subtracts of even 16-bit integers. A case in point: The 6809E in the CoCo has several types of adds, several types of subtracts, and a multiply. Just having the multiply instruction alone is a terrific selling point for the 6809E microprocessor—its predecessor, the 6800, had no multiply, and neither did comparable microprocessors such as the 8080, Z-80 (Mods I, II, III), and the 6502 (Apple II, III, Vic 20).

Now that we have a built-in multiply in the 6809E, we do not have to write a "software" multiply. Software multiplies typically take 100 times longer than a hardware multiply. The 6809E MUL instruction multiplies the contents of the A and B registers to get a 16-bit product (with the result going into the D register, A and B combined) in 11 machine cycles, or about 12.4 microseconds. A comparable software multiply would take about 150 microseconds.

NOW FOR THE DIV INSTRUCTION . . .

What about the DIV instruction? Let's see DAA, DEC, EOR . . . What! No divide? Ha, ha . . . They must have made a mistake and left it off of the Instruction Set Summary "crib card". Surely it's in the EDTASM+ manual . . .

Don't bother looking any further! There is no DIVIDE. If you want a divide, you'll have to implement it yourself. In this column we'll look at how we can write a software divide in lieu of a built-in DIV and see how closely we can come to that MUL instruction time of 12.4 microseconds. You may be surprised. Or appalled . . .

WHAT TYPE OF DIVIDE?

First of all, consider what type of divide should be implemented. The MUL is an "8 by 8" multiply. It takes two 8-bit operands, multiplies them, and puts the result into the D register. (If you're not familiar with the 6809E registers, you'd better go back to earlier columns, or the EDTASM+ manual.)

Of course, in 8 bits we can hold values from 00000000 (0) through 11111111 (255). The product of the MUL will be 0000000000000000 through 1111111100000001 (\$FE01 or 65025 or 255 times 255).

Note that the multiply is an "unsigned" multiply. The two operands are considered unsigned numbers, rather than

"two's complement" numbers. This is typical for rudimentary multiplies, as we can always test operands for their sign, take the absolute value, multiply, and then negate them for the proper product if necessary.

The MUL could have been a "16 by 8", or a "16 by 16", but larger multiplies can be built from the basic "8 by 8", and the "8 by 8" is convenient in terms of cpu register sizes.

The reason I'm going into detail on the MUL is that it seems reasonable to implement a divide that is somewhat equivalent to the MUL in terms of size of operands. Let's see, if we divided an 8-bit operand by another 8 bit operand, we couldn't do much. How about a "16 by 8" divide, dividing a 16-bit operand by an 8 bit operand? That seems reasonable. It would allow us to hold a maximum dividend of up to 65,535 (1111111111111111) and work with a maximum divisor of 255 (11111111).

Like the MUL, the divide should probably be an "unsigned" divide; we can test the dividend and divisor for sign initially, take the absolute value, divide, and then negate the quotient if necessary. Are there any other considerations? One thing that we have with the DIV that we didn't have in the MUL is a "second" result. We not only have a quotient, we have a "remainder", representing the fractional portion of the result. (Don't forget that the MUL and DIV will operate on "integer" values and not mixed numbers.) It would be nice to make some provision for retaining this remainder, if it's convenient.

What can we say about the sizes of the quotient and remainder? It appears that the maximum quotient will be 16 bits, as 65,535 divided by 1 is a legitimate division. The maximum remainder will be 254, produced when the divisor is 255 for various dividends. We can therefore hold the remainder in 8 bits.

Next question: Now that we have the divide "roughed out", how do we implement it?

WHICH REGISTERS SHALL WE USE?

The first order of business is to decide where to hold the operands. We could put them in RAM somewhere, but why not take advantage of the 6809E registers? In general, if we can hold operands in registers, we'll be able to speed up the program. Let's see, we have a somewhat limited choice of registers . . . Let's assume that the dividend is in the X register and that the divisor is in the A register on entry to the divide subroutine. On exit from the subroutine, we'll put the 16-bit quotient in X and the 8-bit remainder in the A register. The dividend and divisor will be destroyed, and we'll leave it up to the "calling program" to save them if necessary.

We now have the scheme shown in Figure 1. The DIV subroutine will be a machine-language program "callable" by other assembly-language code or by BASIC through the USR call.

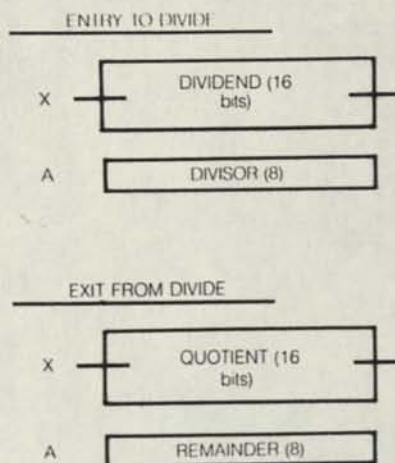


Figure 1.

WHAT'S THE ALGORITHM?

That was the easy part. The next question is: How do we actually do the divide?

There are some standard ways of doing software divides, developed over a number of years, so don't feel bad if obvious algorithms don't immediately spring to mind.

One algorithm we could use is to do "successive subtraction". In this method of division, the divisor is subtracted from the dividend until the result "goes negative". The quotient is the number of times that the divisor could be subtracted from the dividend before the result went below 0. The remainder is the "residue" on the last subtract. An example is shown in Figure 2.

DIVIDE 100/23		QUOTIENT	REMAINDER
	100	0	
	-23		
	77	1	
	-23		
	54	2	
	-23		
	31	3	
	-23		
	8	4	
	-23		
RESIDUE "WENT NEGATIVE" HERE →	-15	4	
	+23		
	8		8

Figure 2.

A divide by this "successive subtraction" is shown in Figure 3. The actual divide subroutine is in lines 160 through 370. The preceding code is a short "driver" routine to interface to BASIC. Let's look at the actual divide first.

```

0000 0E 3FF0 00100 DRIVER LDX 3FF0 GET DIVIDEND
0003 06 3FF2 00110 LDA 3FF2 GET DIVISOR
0006 0D 07 00120 BSR DIVRSB DO DIVIDE
0009 0F 3FF3 00130 STX 3FF3 STORE QUOTIENT
000E 07 3FF5 00140 STA 3FF5 STORE REMAINDER
000E 39 00150 RTS RETURN
00160 *****
00170 * UNSIGNED DIVIDE, DIVIDES 16 BIT UNSIGNED NUM-
00180 * BER BY 8 BIT UNSIGNED NUMBER, DIVIDE BY 0 NOT
00190 * ALLOWED, AS IT WILL PRODUCE SAME RESULT AS
00200 * $FFFF/$01, QUOTIENT WILL BE $0000 - $FFF,
00210 * REMAINDER WILL BE $00 - $FE (AS IN $00FE/$FF).
00220 * ENTRY: X=DIVIDEND, $0000 - $FFFF
00230 * A=DIVISOR, $01 - $FF
00240 * EXIT: X=QUOTIENT, $0000 - $FFF
00250 * A=REMAINDER, $00 - $FE
00260 *****
00270 DIVRSB PSHS A SAVE DIVISOR
00280 TFR X,D DIVIDEND NOW IN D (A+B)
00290 LDX #1 INITIALIZE QUOTIENT
00300 DIV010 LEAX +1,BUMP QUOTIENT
00310 SUBB +5 SUBTRACT DIVISOR
00320 SBCA #0 NOW ANY BORROW
00330 BCC DIV010 GO IF RESULT <

```

```

00340 ADDB +5,ADD BACK TO GET REMAINDER
00350 TFR B,A REMAINDER
00360 LEAS +1,S RESET STACK
00370 RTS RETURN
00380 END
00000 TOTAL ERRORS
DIV010 0016
DIVRSB 000F
DRIVER 0000

```

Figure 3.

Coming into the DIVRSB subroutine, we have the dividend in X and the divisor in A. The PSHS A instruction transfers the divisor to the "stack". The dividend in X is now moved to the D register (A and B combined). The quotient will be held in X at the end, and X is loaded with -1 as an initial value.

We're now at DIV010. Lines 300 through 330 make up the loop for the successive subtract. For every "iteration" through the loop we do the following:

- Bump quotient in X by 1 (goes to 0 on first time)
- Subtract the divisor in the stack from B
- Subtract a possible "borrow" from A
- Branch back to the beginning if the result is not below 0

Some of these operations need explanation:

The divisor was stored in the stack and is accessed by "S", which says use the stack pointer register S as an index register to get the operand to be subtracted from the contents of the B register. The stack pointer always points to the last byte stored, and the stack, of course, "builds down" in memory. Why didn't we store the divisor in another cpu register? We only have Y left, and we can't use it to hold an operand for the subtract.

The subtract was an 8 bit subtract. The "residue", the result of each successive subtract, is held in D, which is A and B together, with A holding the most significant 8 bits, and with B holding the least significant 8 bits. As the divisor is only 8 bits, we can do the subtract by subtracting the divisor from B and then subtracting a possible "borrow" from the A register. This is done by doing a "subtract with carry (borrow) with an immediate operand of 0 (SBCA #0), which subtracts only a (possible) borrow from A. In other words, only 0 or 1 is subtracted from the upper 8 bits of the residue.

If the residue after the SBCA goes below 0, the carry bit in the condition codes will be set to a 1, and the loop ends. At this point an add of the divisor must be done to restore the proper remainder, and this is done by ADDB ,S. The remainder in B is then transferred to the A register. The X register holds the proper quotient. The LEAS +1,S resets the stack pointer by adding 1 to its contents, causing it to bypass the divisor byte and point back to the return address for the RTS.

The driver portion of this subroutine is an interface to BASIC. It assumes that the dividend has been stored in locations \$3FF0 and \$3FF1, and the divisor has been stored in location \$3FF2. It then calls DIVRSB and on return stores the quotient in locations \$3FF3 and \$3FF4 and the remainder in location \$3FF5.

The entire code for DIVRSB is "position independent", or "relocatable". It can be put anywhere in RAM and run properly, as long as the dividend and divisor are in the \$3FFx area.

Figure 4 shows the code incorporated as BASIC program DATA values and moved to the \$3F00 area. An excellent way to do this is as follows: Assemble the program using the EDTASM+ (Don't have one? Get one as an indispensable tool for assembly language, and I'm not just helping Radio Shack sell ROM packs either. EDTASM+ is a subset

of the excellent EDTASM+ for the Model I and allows in-memory assembly, "symbolic" debugging, and a high degree of interactivens. More on EDTASM+ in another column . . .)

```

100 DATA 190,63,240,182,63,242,141,7,191,63
110 DATA 243,183,63,245,57,52,2,31,16,142
120 DATA 255,255,48,1,224,228,130,0,36,240
130 DATA 235,220,31,152,50,97,57
140 FOR I=&H3F00 TO &H3F24
150 READ A: POKE I,A
160 NEXT I
170 DEFUSR0=&H3F00
180 INPUT DD,DV
190 POKE &H3FF0,INT(DD/256)
200 POKE &H3FF1,DD-INT(DD/256)*256
210 POKE &H3FF2,DV
220 A=USR0(0)
230 PRINT DD1/*1DV1:"S/B"1INT(DD/DV)1"REMAINDER"1DD-INT(DD/DV)*DV
240 PRINT "HAS "1PEEK(&H3FF3)+256+PEEK(&H3FF4)1" WITH REMAINDER "1PEEK(&H3FF5)
250 GOTO 100
260 FOR I=&H3F00 TO &H3F24
270 PRINT HEX$(PEEK(I)),
280 NEXT I

```

Figure 4.

When you've assembled, check to see that all code is position independent. (Use BSRs and LBSRs in lieu of JMPs or JSRs.) Now assemble in memory and then go to Zbug by entering the "Z" response.

Once in Zbug, enter "O10" for decimal "output", "byte" mode ("B"), and numeric ("N"). Now display your program from the program start and record the decimal values (designated by a "T" suffix).

Use those decimal values in DATA statements in the BASIC program. Use code similar to Figure 4 to move the values to a protected area (by an initial "CLEAR"). Before execution, dump the area in BASIC by using a PRINT HEX\$(PEEK(I)) command. Compare the values from the print with your original assembly-language listing. If they compare exactly, you have successfully incorporated the machine code and relocated it in a BASIC program.

To use this Extended Color BASIC program, CLEAR 200, &H3EFF to protect the \$3F00 area and execute. The BASIC program will then relocate the program and ask you for a dividend and divisor. It will then call the DIVRSB program and compare the expected results with the actual.

Try the program with various values, and you'll see that it works for any legitimate operands.

A VOICE FROM THE AUDIENCE . . . "HOW FAST *IS* IT?"

That's an interesting question. If you think about it, you'll see that the speed of DIVRSB is related to the number of subtracts. Discounting "overhead", the speed of 10000/100 should be about 100 times faster than 10000/1, as 100 times the number of subtracts need to be done for the latter case. Can we measure the actual speed?

We could go through and calculate the speed based on the number of "cpu cycles" from Motorola documentation, but here's an easier way. Figure 5 shows a short BASIC program to POKE the operands and then call DIVRSB. It assumes that the program has been relocated to the \$3F00 area. The program as shown POKEs a "57" into location \$3FF6. A 57 is the "op code" value for an RTS, or Return From Subroutine instruction.

```

2000 DEFUSR0=&H3FF6
2010 POKE &H3FF0,3
2020 POKE &H3FF1,&HEB
2030 POKE &H3FF2,100
2040 POKE &H3FF6,57
2050 FOR I=1 TO 10000
2060 A=USR0(0)
2070 NEXT I

```

Figure 5.

As it stands, the program does 10,000 calls to the USR subroutine at location \$3FF6, which is simply an RTS. If we time the execution of this program, we'll be able to measure the BASIC "overhead" before measuring the actual speed of DIVRSB. It turns out that this program takes about 55.5 seconds to go through 10,000 calls.

Now, substitute "&H3F00" for the "&H3FF6" in the DEFUSR and measure the time for 10000 iterations. The POKEs in the program setup a dividend of 1000 and a divisor of 100, so we'd have 10 subtracts using the successive subtraction method. If you measure it, you'll find that the divide takes about 58.1 seconds for 10,000 iterations (that's 10,000 divides of 1000 by 100). Subtracting 55.5 seconds (the BASIC "overhead" time from 58.1 seconds gives about 2.6 seconds to execute 10,000 divides in the subroutine itself, or about 260 microseconds per divide. The 260 microseconds is about 20 times more slow than the equivalent hardware multiply! Wait, it gets worse . . .

If you modify the BASIC code at 2000 to POKE different values for the dividend and divisor, you'll find that a divide of 1000 by 10 takes about 1660 microseconds per divide! What is the absolute worst case? This occurs when 65,535 is divided by 1 and 65,535 subtracts are done, and the time required is about 95,000 microseconds or about 1/10 second! And you thought assembly language was fast!

This algorithm for division is therefore not too efficient, and Microsoft probably will not pick it up for inclusion into BASIC interpreter software. If the average quotient is 255 (statisticians may prove me wrong), the average division speed will be about 3 milliseconds, or about 333 divides per second. Can't we do better? Although assembly-language code doesn't have to be "tight" in most programs, it does have to be fast and efficient for commonly used operations such as divides.

ANOTHER ALGORITHM

Is there a faster algorithm that we can use for a "16 by 8" divide, or for that matter, for any divide? No, next question . . .

Just kidding. One that has been used for many years is the "restoring" divide. It emulates a paper and pencil divide. (Hardware designers sometimes use a "non-restoring" divide to speed up hardware divides in larger microprocessors or computers.)

The restoring divide works on paper as shown in Figure 6. The divisor is subtracted from the first group of digits. If the divide "doesn't go", then the subtract is not done and a shift to the next digit position is done, where the process is repeated.

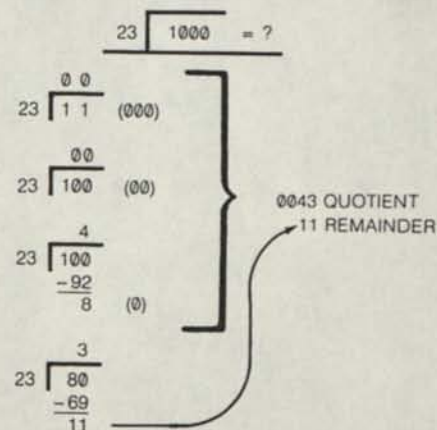


Figure 6.

We can do the same type of thing in the CoCo in binary. (See Figure 7.) Here, however, the computer can't inspect the numbers and tell that the subtract can't be accomplished. In this case the subtract is done, and if a negative result occurs the divisor is added back to the "residue". If the residue is still positive or 0 (0 is a positive number in the 6809E and other microprocessors), then the residue is not "restored" and a 1 bit is put into the quotient.

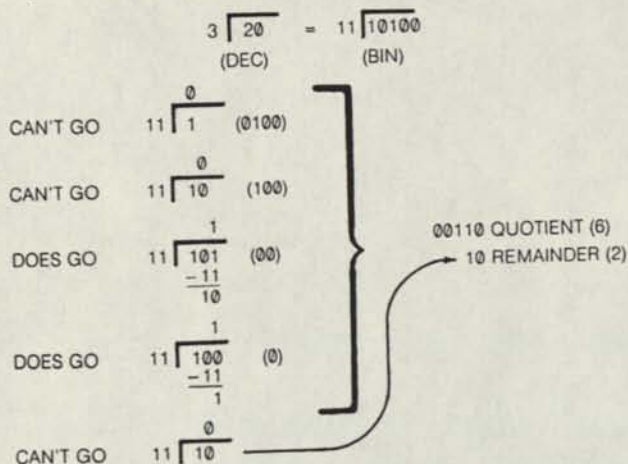


Figure 7.

As you can see from the figure, the operation is essentially a subtract, restore if necessary, put a 0 or 1 bit into the quotient, and shift the residue. Even within this method, there are a number of ways to do the divide. Either the divisor or residue may be shifted. We'll choose to shift the residue, as shown in Figure 8.

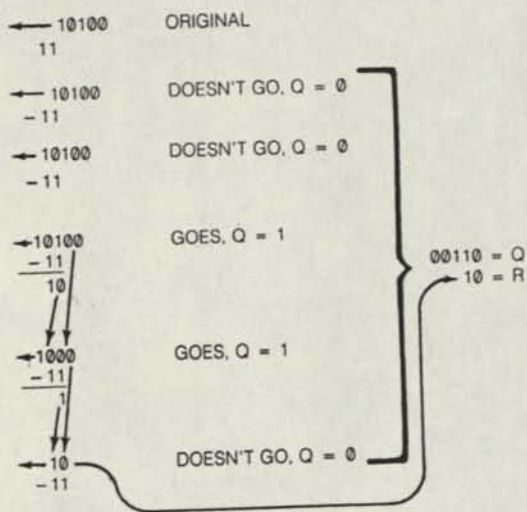


Figure 8.

Figure 9 shows the register setup for the DIV168, or the "16 by 8" divide. The residue is held in the D register which is the A and B registers taken together as one 16-bit register. The divisor is held in the "S" stack, the normal hardware stack. I could have used the "U" stack, or the user stack, but I guess I'm a traditionalist

The D register is shifted left one bit at a time by shifting B left, and then shifting A and the "carry" from B left. After each shift, the divisor from the stack is subtracted from the A register. We don't have to subtract anything from B because it cannot change until it is shifted into the upper 8 bits.

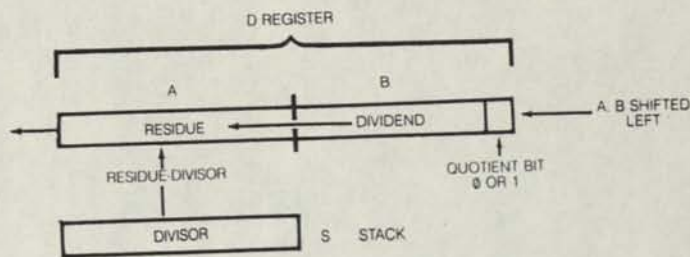


Figure 9.

After the subtract and possible restore, the quotient bit is put into the least significant bit of the B register. The shift of D, therefore, not only shifts the "residue", but the accumulation of the quotient. The quotient can be stored in B because the shift leaves nothing in the 1s bit of the B register, and the residue is being shifted inexorably towards the left. Although at first glance it might seem that I'm being too darn clever with this scheme, it is a not uncommon way of accumulating the quotient, especially when one doesn't have an overabundance of registers, as in the 6809E.

Eight bits of the dividend are handled at one time. At the end of 8 iterations, the B register holds the upper 8 bits of the quotient. This is saved, and the B register is then loaded with the lower 8 bits of the divisor. After 8 more iterations, the 8 quotient bits in B are stored as the 8 1s quotient bits. At this point, the A register holds the remainder, the "dregs" of the residue.

So much for the algorithm. It seems complicated, but you might try actually working it out by playing computer with ones and zeroes. Its really not very profound. On to the actual code . . .

DIV168 CODE

The DIV168 code is shown in Figure 10. Lines 160 through 470 represent the actual subroutine which can be called by other assembly-language code. Lines 100 through 150 are a driver program to permit easy interfacing to BASIC.

```

0000 BE 3FF0 00100 DRIVER LDX 33FF0 GET DIVIDEND
0003 86 3FF2 00110 LDA 33FF2 GET DIVISOR
0006 8D 07 00120 BSR DIV168 DO DIVIDE
0008 BF 3FF3 00130 STX 33FF3 STORE QUOTIENT
0009 87 3FF5 00140 STA 33FF5 STORE REMAINDER
000E 39 00150 RTS RETURN
00160 *****
00170 * UNSIGNED DIVIDE. DIVIDES 16 BIT UNSIGNED NUM-
00180 * BER BY 8 BIT UNSIGNED NUMBER, DIVIDE BY 8 NOT
00190 * ALLOWED, AS IT WILL PRODUCE SAME RESULT AS
00200 * $FFFF/8. QUOTIENT WILL BE $0000 - $FFFF.
00210 * REMAINDER WILL BE $00 - $FE (AS IN $00FE/$FF).
00220 * ENTRY: X=DIVIDEND, $0000 - $FFFF
00230 * EXIT: X=QUOTIENT, $0000 - $FFFF
00240 * A=DIVISOR, $01 - $FF
00250 * A=REMAINDER, $00 - $FE
00260 *****
00270 DIV168 PSHS X:A DIVIDEND, DIVISOR
00280 CLRA CLEAR 1/2 OF DIVIDEND
00290 LDB +1,S GET MSB OF DIVIDEND
00300 BSR DIVIDE DO 8 DIVIDES
00310 STB +1,S REPLACE 1ST 1/2 OF D'DEND
00320 LDB +2,S GET LSB OF DIVIDEND
00330 BSR DIVIDE DO 8 DIVIDES
00340 STB +2,S REPLACE 2ND 1/2 OF D'DEND
00350 PULS B:X DISCARD D'ISOR, GET Q
00360 RTS RETURN
00370 DIVIDE LDX #0 SETUP COUNTER
00380 DIVD10 SHIFT A,D LEFT ONE BIT
00390 ROLA
00400 ORB #1 PRESET Q BIT TO 1
00410 SUBA +2,S DO SUBTRACT
00420 BCC DIVD20 GO IF + OR 0
00430 ANDB #$FE RESET Q BIT
00440 ADDA +2,S RESTORE
00450 DIVD20 LEAX -1,X DECREMENT COUNT
00460 BNE DIVD10 GO IF NOT 0
00470 RTS RETURN TO DIV168
00480 END
00000 TOTAL ERRORS
00000
DIVD10 0024
DIVD20 0030
DIV168 000F
DIVIDE 0021
DRIVER 0000
DRIVER 0000

```

Figure 10.

DIV168 is entered with the dividend in the X register and the divisor in the A register. The PSHS pushes X and A into the S stack. After the PSH, the stack looks as shown in Figure 11, with the S pointer pointing to the last byte (A) pushed.

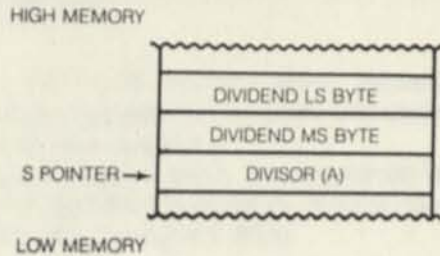


Figure 11.

The CLRA clears the A register. The LDB +1,S loads the upper 8 bits of the dividend from the stack. Subroutine DIVIDE is now called.

DIVIDE does 8 iterations of the divide. The X register is loaded with a count of 8. A shift of D is done by two shifts, the first of B and the second of A. The LSLB does a logical shift of B to the left, with a 0 going into the 1s bit of B, and the ms bit of B going into the carry condition code. The ROLA "rotates" A to the left; this has the effect of shifting A left one bit with the carry going into the 1s bit of A.

Now the Q bit in the 1s bit of B is set to a 1. This action makes the assumption that the subtract will be successful. If it is not, this bit will be reset.

A subtract of the divisor is now done from A. If the subtract left a result of 0 or greater, the C flag will be "clear" or 0, and the restore is not done. If the result was less than 0, the Q bit is reset and the divisor is added back to the A register by ADDA +2,S.

We're now at DIV020. At this point the Q bit has been stored. The count of 8 iterations in X is now decremented by the LEAX -1,X. If this count is not 0, the DIV010 loop is repeated. After 8 iterations, an RTS is done back to DIV168 code.

One tricky point in DIVIDE: Note that the subtract and add used +2,S. This is a difference of 2 from Figure 11 because the BSR to DIVIDE pushed 2 more bytes onto the stack and adjusted the S pointer. We're now back at line 310. At this point the first 8 iterations have been done and B holds the upper 8 bits of the quotient. This is now stored in the upper 8 bits of the divisor in the stack, as we'll no longer be needing them. The lower 8 bits of the divisor is now put into the B register and another call to DIVIDE is made. On the return from this call, the lower order quotient bits replace the lower-order divisor bits.

The last instruction before the return puts the divisor in the B register and the quotient in the X register.

DIV168 IN BASIC

The DIV168 subroutine can be called from other machine-language code, or it can be used from BASIC. If used from BASIC, the DRIVER code before DIV168 picks up the dividend and divisor before the divide and stores the quotient and remainder after the divide.

The "parameter block" area allocated for the input and output arguments is at RAM location \$3FF0. Locations \$3FF0 and \$3FF1 hold the dividend. Location \$3FF2 holds the divisor. Locations \$3FF2 and \$3FF3 hold the 16-bit quotient, while location \$3FF5 holds the remainder.

A BASIC program that incorporates the machine-language code of DIV168 is shown in Figure 12. This program holds the codes as DATA values and relocates the code to the \$3F00 area. Protect this area before execution by a CLEAR 200,&H3EFF.

```

100 DATA 198,63,248,182,63,242,181,7,191,63
110 DATA 243,183,63,245,57,52,18,79,238,97,141
120 DATA 11,231,97,238,98,141,5,231,98,53
130 DATA 28,57,142,8,8,88,73,282,1,168
140 DATA 98,38,4,198,254,171,98,48,31,38
150 DATA 248,57
160 FOR I=&HCF00 TO &HCF34
170 READ AT PEEK I:A
180 NEXT I
190 DEFUSR0=&HCF00
200 INPUT DD:DV
210 PEEK &HCF00:INT(DD/256)
220 PEEK &HCF1+DD-INT(DD/256)*256
230 PEEK &HCF2:DV
240 A:USR0(I)
250 PRINT DD;" / "DV;" = "B*INT(DD/DV);" REMAINDER" DD-INT(DD/DV)*DV
260 PRINT "MAG "PEEK(&HCF3)+256+PEEK(&HCF4);" WITH REMAINDER "PEEK(&HCF5)
270 GOTO 200

```

Figure 12.

The BASIC program asks for the dividend and divisor, calls DIV168, and then compares the expected result with the actual result. You really don't have as much faith as a grain of mustard seed, do you . . .

DIV168 works, and works quite a bit faster than the successive subtraction program. Unlike the latter program DIV168 is about the same for the best case and worst case and runs between about 510 and 640 microseconds per divide. You can see from Figure 4 that the only difference in speed would be caused by the number of times that the "restore" was done.

Is this fast enough for a divide? It's pretty slow to my mind, although it's not that badly written (if one can be subjective about these things . . .). Can you do better? Why not? How about flexing your programming muscles and coming up with a faster divide? Good luck. You may advance the cause of the CoCo immeasurably . . .

Corrections to June Barden Article

Our apologies to Mr. Barden but there were typographical errors in the programs in Listing 1 (page 14) and Listing 2 (page 16) of the June article

Corrections to Listing 1

Change the following lines to read:

```

371A 66 C4 ##31#      LDB      0      USE SMALLER LENGTH
32F2 27 B4 ##42#      BEQ     BRTB45  GO IF EQUAL
3756 AE E1 ##64#      LDX     1,6    GET # ENTRIES

```

Corrections to Listing 2

Change lines 130 and 150 to read:

```

130 DATA 198,63,254,52,16,238,228,174,94,48,31,79,52,
      18,166,196,39,42,166,196,238,69,168,69,36,2,238,
      196,52,1,174,66,16,174,71
150 DATA 198,238,69,231,196,167,69,234,69,234,228,
      231,228,91,69,174,97,48,31,175,97,38,176,166,228,
      59,99,38,161,58,98,57

```


Hex to Decimal

Charles M. Thonen
8005 B Gen Grant Dr.
Black River, NY 13612

Unlike the HEX\$ function which converts the Hexadecimal number to decimal, the Color Computer has no built in Decimal to Hex function.

This conversion is easily accomplished by use of the value function and the &H symbols. This can be illustrated by the following short program listing.

```
10 INPUT "HEX NUMBER "; H$
20 PRINT "DECIMAL="; VAL("&H" + H$)
```

Landline

Rodney M. Hallam, Age 15
265 Rolfe Road
De Kalb, IL 60115

This program which I call Landline generates a "tree-dimensional" image. To me it looks like a landscape tableau. To my mom, it looks like someone "shaking a rug" Anyway, they are both good descriptions of an image that is hard to describe.

What the program actually consists of is the repetitive plotting of a sin wave. Each time it is drawn, however, the amplitude (AM) and pitch (P) change slightly. To make the image seem to come at you from left to right, change line 60:

```
60 P = P - .001
```

to

```
60 P = P + .001
```

This program also makes a great example for showing off the "Screen Print" routine on a Radio Shack printer capable of handling graphics.

If one wished to hear a little music to go along with the plotting, delete line 130. (You call that "music"?)

```
10 PMODE 4,1
20 P = (RND(6) + 1) / 100
30 AM = RND(2) + 1
40 PCLS (1)
50 AM = AM + 4
60 P = P + .001
70 COLOR 0,1
80 SCREEN 1,0
90 LINE (0,94) - (0,95), PSET
100 FOR T = 0 TO 255 STEP RND(15) + 5
110 A = SIN(T * P) * AM
120 LINE - (T, INT(A) + 95), PSET
130 GOTO 150
140 SOUND 255 - (INT(A) + 95), 1
150 NEXT T
160 IF AM < 90 THEN 50
170 FOR T = 1 TO 1000
: NEXT T
180 GOTO 20
```

Print to Screen and Wraparound

Gerald Plueard and Family
501 North 143rd Street
Seattle, WA 98133

I have the Line Printer VII and programs that use the printer. But I do not want to print all the time and have found that, with my Color BASIC 1.0 Computer, if the switch is put in the 8BS (UP) position, the computer will run and the information is seen on the screen. Then if I want to print on the printer all I have to do is run the program again but this time with the switch in the 7BS (MID) position.

```
420 PRINT
: PRINT " TYPE YOUR SENTENCE NOW"
450 A$=INKEY$
: IF A$ = "" THEN 450
480 IF A$ = CHR$(13) THEN 450 'CHR$(13) = <ENTER>
KEY
500 IF A$ = CHR$(8) THEN 450
510 PRINT A$;
520 A$(X) = A$(X) + A$
530 IF LEN(A$(X)) > 65 AND A$ = CHR$(32) THEN 540
ELSE 450
540 X = X + 1
: PRINT#-2, A$(X-1)
560 GOTO 450
```

The next program allows the Line Printer VII to print without cutting words off on the right edge of the paper.

```
2510 FOR Y = 1 TO 50 'NUMBER OF SENTENCES
2610 LINE INPUT D$(Y)
2670 INPUT "DO YOU WANT TO PRINT IT?"; S$
: IF LEFT$(S$,1) = "Y" THEN 2680 ELSE 2810
2680 LL = LEN(D$(Y))
2690 IF LL > 255 THEN LL = 255
2700 FOR Z = LL TO 1 STEP -1
2710 IF LEN(D$(Y)) < 1 THEN 2830
2720 IF LEN(D$(Y)) < DW1 THEN 2800 'DW1 = 40 FOR
DOUBLE WIDTH, 80 FOR NORMAL
2730 PLAY "L235;12"
2740 IF MID$(D$(Y),Z,1) = CHR$(32) THEN IF
LEN(MID$(D$(Y),1,Z)) < DW2 THEN 2760
2750 NEXT Z
2760 PRINT#-2, MID$(D$(Y),1,Z)
2770 D$(Y) = MID$(D$(Y),Z,LL)
2780 LL = LEN(D$(Y))
2790 IF LEN(D$(Y)) > 1 THEN 2700
2800 PRINT#-2, D$(Y)
2810 CLS
2820 X2 = Y
: PRINT "SENTENCES TYPED" X2
: PRINT "OUT OF 50 "
2830 NEXT Y
```

Data Base Management Systems

This month we will discuss Data Base Management Systems. This field of "data-processing" has become a science in itself, so we obviously can't cover the subject completely in one short article.

The definition of a data base system is a computer-based record keeping system whose overall purpose is to record and maintain information. The information stored in the system can be anything that is considered to be important or of significance to the company or organization that is using the system. The manipulation of the information in the data base is done by the data base management system software. With a bit of programming knowledge, additional reports can be extracted from the data base.

Why should a company use a data base system, and what are some of the advantages of such a system? The primary reason for a company or organization to use a data base is centralized control of information. One of the primary advantages of a data base system is the elimination or reduction of redundant files. Inconsistencies in data can also be avoided to a great extent by reducing the number of times data is entered. This goes hand-in-hand with the elimination of redundant files. Another advantage to a data base is that data can be shared, i.e., the information captured by one group can be used by another.

What type of information should be stored in a particular data base? Any information that would normally be put into an organized filing system is one type of information that could be put into a data base. A data base could also include complete company records—information to be used in the decision making processes of a company or organization. The information contained in the data base can range from a simple personnel file to a complete set of files containing employee information, project information, suppliers, inventory information, parts used in a project, and departments involved in a project. Many companies and organizations use more than one data base. Let's look at some of Radio Shack's data base management systems.

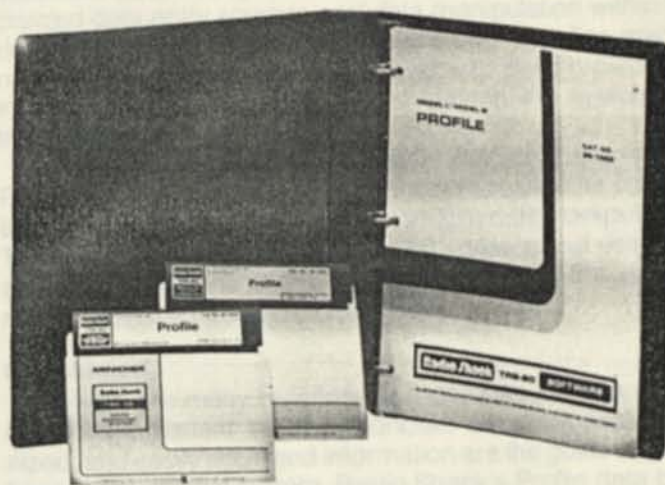
MODEL III DATA BASE SYSTEMS

Profile—Catalog No. 26-1562

Profile for Model I/III allows the user to design a data entry screen to his or her own specifications, with up to 32 fields per record. Records can be variable in length up to a maximum of 254 characters in each record. With this type of versatility a company can create a custom record keeping system with ease.

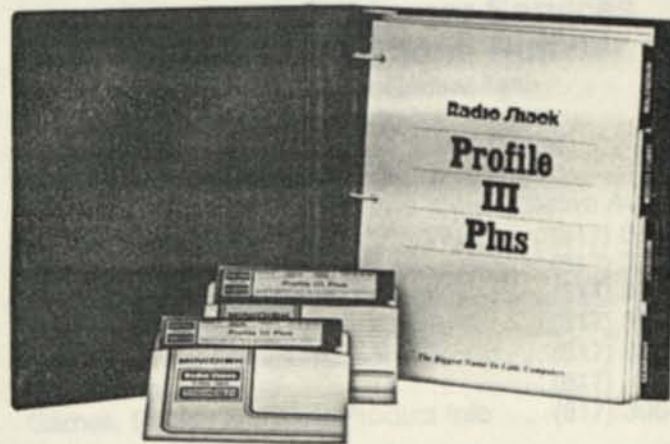
One important consideration in choosing a data base system is the ease with which information can be retrieved from the system. Profile allows the user to find information by any field in the record. The system will also sort the records by any field in the record. Printed reports are also defined by the user, putting a report title and column headings at the top of

the report with the system printing the data below the headings. The data chosen can be selected information or all of the information stored in the data base. Another feature of Profile is the accessibility of the data through the BASIC programming language. This feature allows the user to manipulate the data in any way that is required.



Profile III Plus—Catalog No. 26-1592

The Profile III Plus data base system (Model III only), like Profile, allows the user to define his/her own data entry screens. Each data base can have up to five separate screens to provide for additional flexibility. Profile limited the user to 32 fields on a single screen whereas Profile III Plus allows a total of 99 fields, with a maximum of 36 fields on any one of five separate screens. In Profile the user was limited to 254 characters of information, but Profile III Plus allows multiple segment records, providing space for more than 1000 characters of information per record.



Profile III Plus provides many of the same features of Profile with many advanced data manipulation features such as the ability to perform mathematical functions with the data in specified fields. The searching and sorting features of Profile III Plus are much the same as Profile with the ability to search or sort by any of the 36 fields in the first segment. Profile III Plus gives the added advantage of being able to associate fields from the first segment with one another. A totally customized data base can be created with the user menu feature of Profile III Plus.

Information is easily accessed with Profile III Plus as the user may define up to five different reports where Profile allows only one report format to be stored. Additionally Profile III Plus can be accessed by other Radio Shack software packages such as VisiCalc and SuperScripts adding to its report generating capabilities.

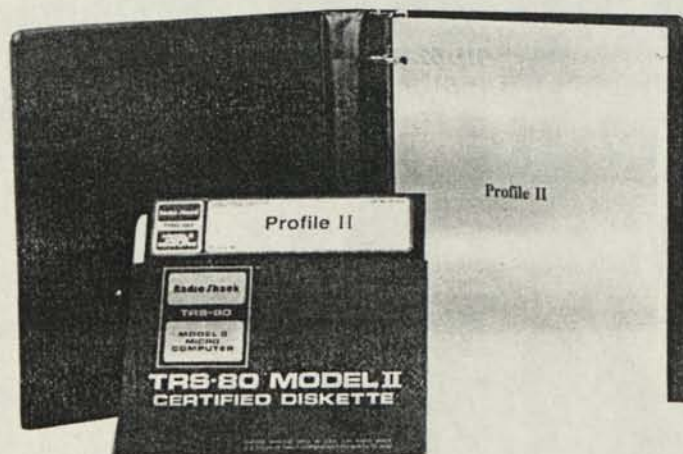
MODEL II DATA BASE SYSTEMS

Profile II—Catalog No. 26-4512

Profile II is a very versatile data base system that lets the user format up to five data entry screens, each screen containing up to 36 fields of information with a maximum number of 99 fields. Profile II also has the ability to use multiple segment records. Multiple segment records allows the user to store up to 850 characters of information per record. The Model II owner with four floppy disk drives can store up to 1800 four segment records or a total of one and a half million characters of information.

Finding information is easy with Profile II. Profile II has the ability to find information by record number or search on any one of the 36 fields in the first segment. Profile II will sort the information in the data base by any of the fields in the first segment. Many times information for reports must meet more than one criterion and Profile II allows two fields to be associated with each other for greater searching and sorting capability. The system allows the addition and subtraction math functions to be performed on the data in the system to give some additional data manipulation capability.

Up to five user defined printed reports are allowed in Profile II. A report can consist of any combination of the fields of information in the system based on the data required in the report. Scripts can also be used with Profile II for additional flexibility in generating reports.



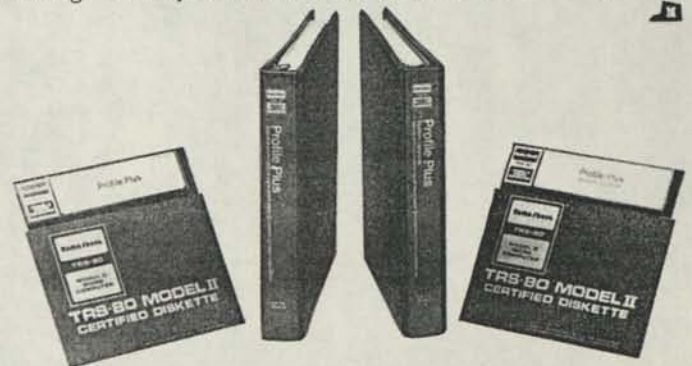
Profile Plus—Catalog No. 26-4517

Profile Plus for Model II is an extension of Profile II with all of the features mentioned above but with enhancements to make the user's data base even more manageable. Records can be defined by the user with five input screens for a total of 99 fields of information. One of the advantages of Profile Plus is the ability to define segments shorter than 256 characters whereas in Profile II the second, third, and fourth segments were 256 characters long with no option to allow for less space to be used. This means that if only 200 characters are required in a data base, the first segment would contain 85 characters and the second segment would have 115 characters instead of 256. This would save a great deal of storage space allowing for additional records to be stored. Profile Plus also has additional field indicators and the ability to associate fields in segment one into groups to allow for specialized data entry screens and data manipulation within the data base. Profile Plus also allows the user to define mathematical formulas that are to be used with certain pieces of information within the data base, making the system extremely flexible.

Profile Plus can be accessed in the same manner as Profile II. The added features of Profile Plus allow for specialized report generation and more versatile data manipulation. The user can design up to five report screens and using the advanced functions of Profile Plus can set up the specific reports required from the data base.

Conclusion

There are many reasons for a company to use a data base management system. Concise, accurate, well organized, and easily accessed information are the goals of a data base management system. Radio Shack's Profile data base management systems can help you achieve these goals.



Computer Customer Services Address and Phone Number

8AM to 5PM Central Time
Computer Customer Services
400 Atrium, One Tandy Center
Fort Worth, Texas 76102

These are New Numbers and Were Effective Aug. 27

Model I/III Business Group	(817) 390-3939
Model II/16 Business Group	(817) 390-3935
Languages and Compilers Group	(817) 390-3946
Color and Pocket Computer Group	(817) 390-3944
Hardware and Communications Group	(817) 390-2140
Educational Software	(817) 390-3302
Games, Books, and New Product Info	(817) 390-2133

Welcome To Someday

Editor's Note: The CompuServe Information Service is one of the largest information and entertainment services available to owners of personal computers and computer terminals. With each issue of TRS-80 Microcomputer News, various features of CompuServe will be discussed. The CompuServe Information Service is sold at Radio Shack stores nationwide and in Canada.

"Someday we'll leave them speechless by coming up with a way to shop, choose and purchase merchandise from our own home."

"Sure."

"Someday we'll develop a way to order and load a computer program from our system into a home computer all by using a couple of commands and then what'll they say?"

"Right."

"Someday we'll develop a way to do all our banking from our living rooms and see what they say to that."

"Whatever you say."

"Then what if I say 'Welcome to Someday,' today?"

Someday is here and we'd like you to get to know us. We're the CompuServe Information Service (CIS), and we'd like to welcome you to a tour of someday today.

We at CompuServe feel we're riding on the crest of a technological tidal wave as the largest electronic information and entertainment service available to owners of personal computers and computer terminals. We're taking entrepreneurs up on their ideas of computerized home management and coming up with a few of our own.

CompuServe is a computer services company headquartered in Columbus, Ohio, and is a subsidiary of H&R Block. We serve corporate and government customers who share time on our computers and often depend on our application products to manage their information and data processing needs.

We've been serving some of the nation's largest industries in major metropolitan areas through coast-to-coast sales and service facilities since 1969. By the late 1970s, CompuServe had an established reputation for reliability in providing software solutions, operations assistance and research and development capabilities to its business customers. Having successfully introduced electronic information services to the business world, we then turned our efforts toward personal computing services and opened up a world of information to the hobbyist, the professional and the novice.

CompuServe Information Service is available in Radio Shack outlets from Maine to Modesto and Tallahassee to Toronto. As a new CIS customer you can purchase a starter kit containing a special access number, a secret password, a CIS User's Guide and a free hour of time on us. In addition to access to all of our services you'll receive a free subscription to Today magazine, a CompuServe publication which keeps you up to date on industry trends, plus our monthly newslet-

ter, Update, which lists the latest innovations and additions to CIS.

Our primary and supplemental networks provide access to the information service through a local telephone call in more than 300 U.S. cities and Canada. At a cost of \$5 per hour, (in the U.S.) customers can access the service anytime before 5 a.m. and after 6 p.m. and all day Saturday, Sunday and holidays, thus utilizing CompuServe's off peak-hours computer power.

CompuServe provides you with the ability to communicate with other CIS users either across the street or across the country by using electronic mail. CIS combines menu-choice and word-search technology to display information arranged in an easy-to-find, "Go" anywhere format. As well as providing you with information, our service enables you to enhance and dramatically increase the capabilities of your TRS-80 computer or Videotex terminal.

CompuServe categorizes its offerings in menu form and enables you to access topics of interest by making selections from more specific menus. We have designed a subject index which alphabetically lists our subject areas and provides a page number enabling you to go directly to your subject without searching through the separate menus. In addition to our alphabetized index, we provide a built in key word search capability which enables you to enter the key word at the appropriate prompt. The computer will then respond with your topic and its electronic page number.

The CIS main menu contains the following listings: home services, business and financial services, personal computing services, user information and the subject index.

EXTRA, EXTRA! ELECTRONIC NEWSPAPERS HIT THE SCREENS

The CompuServe Information Service offers a number of major daily newspapers and The Associated Press newswire service under the Home Services menu category.

All of the electronic papers offer basic news as well as special features created specifically for CompuServe customers including on-line updating of current news, video games and advertisements.

The electronic editions feature the whole gamut of world, U.S. and local news, sports, theater and movie reviews, books, financial and art news and many other standard newspaper features.

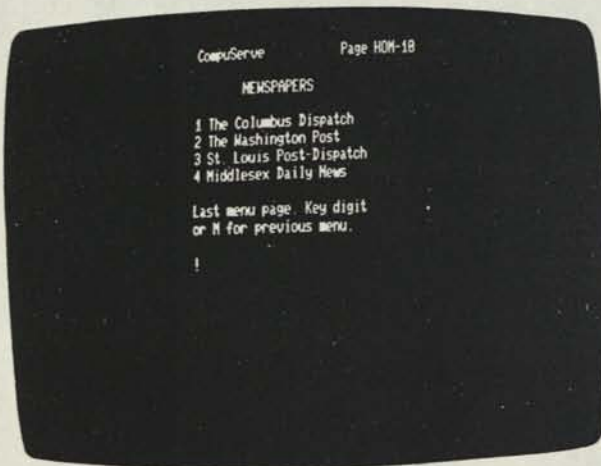
The Washington Post, for example, can help you find a new job in the Washington D.C. area when you input the job title desired, salary and geographic preference. The system provides you with a list of acceptable job titles and state abbreviations required for the search. The computer does the search for you using this information while you sit back. No walking, driving or pounding on doors.

The St. Louis Post-Dispatch offers a classified marketplace with almost 300 categories of daily updated advertise-

ments. News events are updated hourly through the system and comments and suggestions are welcomed through the paper's own electronic mail "Feedback" service.

The Columbus Dispatch offers a wide variety of on-line entertainment including entertainment news, movies and other reviews, plus the local, state and national news. The Dispatch also offers two video games: Starforce Marauder and Remove: a logic game. Starforce Marauder, an interactive space mission could raise even the eyebrows of the skeptical Spock. To complete your mission, you must move through quadrants on a patrol route provided with an arsenal of weapons. You must complete your patrol route before your battle armor runs out of power.

Remove is based on a Chinese game of intelligence and is played with a certain number of objects stacked in a column. The key to the game is to remove at least one object from only one column per turn. But if you're the player who picks up the last remaining object, you lose.



TAKE A "BYTE" OUT OF HOUSEHOLD CHORES

Many other services are available under our Home Services menu, such as weather information, a reference library, containing miscellaneous information such as video program reviews, movie reviews, and the Popular Science and Better Homes and Gardens magazines.

Also, under our Home Services menu is the communications section which offers the interactive services of Email, or electronic mail, "CB," the User's Directory, and the National Bulletin Board. This selection also includes Shop/Bank at Home, Groups and Clubs, Games and Entertainment, Education, which offers programs such as mind and memory builders, and Home Management programs which help you accomplish the more complex tasks of balancing your checkbook or amortizing a loan.

MIND YOUR OWN BUSINESS

Another main menu feature is Business and Financial Services providing news and reports from the business world. Information providers here include the Archer Commodities Report, the Small Business Report and the Business Information Wire, to name a few.

YOU SAY YOU WANT TO GET PERSONAL . . .

The Personal Computing Services listed on the main menu offer several on-line newsletters from companies like

Tandy, Commodore, Atari and RCA, as well as the Micro Advisor and Popular Electronics magazine. There is also a reference section featuring information on languages, word processing and mathematics programs. The Communications section of PCS offers the interactive services of CB Email and a shop-at-home section, which points you to the Software Exchange program and the Art Gallery where you can order or review the line printer art offerings. The Groups and Clubs section takes you into the Special Interest Groups (SIGs), and the Programming and Computing section enters you into the Command Mode allowing you to expand the programming capabilities of your personal computer by tapping into CompuServe's main frame computing power.

HELLO? INFORMATION . . .

The next main menu area, User Information, tells you immediately at log on "What's New" for the week in the way of our products, new telephone access numbers and new access locations. CompuServe's free Feedback section allows you to tell us what's on your mind. There are also instructions on how to change your terminal defaults or password and how to review your charges for the month. There is always a listing of telephone access numbers and the current rates which accompany them.

Also under User Information, CIS provides CompuServe Viewpoint, a series of articles which help explain our information service and Videotex. It offers answers to your billing questions, a short course in FILGE, (our text editing program), results of our user surveys plus an explanation of CompuServe and Videotex and career opportunities with CompuServe.

LET YOUR FINGERS DO THE KEYING THROUGH THE USER'S INDEX.

The last main menu item is the CompuServe Information Services Index. This is a complete alphabetical listing of our subjects. The user index appears on the screen in the following form:

Access	Go PCS-30
Access Phone Number	Go CIS-4
Adult Education	Go HOM-70
Adventure Game	Go HOM-60

The "Go" command in these examples refers to the method used in CIS to go directly to a page rather than running through all the separate menus.

"Go n . . ." refers to Go directly to page "n". N stands for a number alone or an information provider/number combination such as CIS-4.

CompuServe is proud of its over 450 subject offerings and we'd like to highlight some of our more popular services from the multitude.

BREAKER BREAKER ONE-NINE. WHAT'S YOUR ID NUMBER?

CompuServe offers "CB". This is our simulation of the Citizens' Band radio which allows you to talk with other CompuServe users around the city, state, country or Canada.

Our CBers have "handles" just like the regular CB users for the purpose of identification to other CBers once on the system. To gain access to CB, users enter the system with their user ID number and password and select "Communi-

tions". CB users can send, receive, and monitor messages on any of the 40 channels offered over the system.

CBers can talk to as many people as they wish at one time or just one person in a private conversational mode. Everything typed onto the screen is seen by other CompuServe CB users logged on to the same channel. Becoming a skilled CB user is accomplished by following a few basic commands.

/TUNE changes the channel so that you can send and receive messages on a different channel.

/STATUS tells you how many people are on the channel at the time you issued the command.

/MONITOR allows you to follow conversations on one or more channels.

/SCRAMBLE allows you to have a private conversation with another CBER by issuing a special code which only you and that other person know. By doing this, you and the other person are reading your messages while others are just seeing the scrambled code.

CIS customers can find the CB Simulator under the main menu item 2, Communications.

SIGs: OUR MEMBERS ARE "SPECIALLY INTERESTED IN GROUPS."

CompuServe offers another area where you can share your interests with others on-line. These are the Special Interest Groups or SIGs. SIGs are electronic clubs for those of you who like to converse about a particular subject.

We offer many different SIGs. One such SIG is MNET80, a TRS-80 Users Group providing information and the latest news on software for TRS-80 microcomputers. Its members include professional programmers and companies using TRS-80 hardware and software. A membership fee of \$20 allows access to TRS-80 experts and to other TRS-80 enthusiasts around the country.

Offerings in the MNET80 SIG include complete cassette based smart terminal programs, scores of original pieces of software written by its members and a section for the Color Computer, and TRS-80 Models I, II and III.

The MCONN SIG is sponsored by CompuServe and the Microperipheral Corporation, a manufacturer of modems for popular microcomputers. This SIG is a communication tool between Microperipheral Corporation and its dealers, customers and potential customers. The only requirement for

joining this SIG is that your company has purchased either hardware or software from the Microperipheral Corporation. Membership can also be requested upon completion of an entry in the User Interest Log.

A non-SIG informational product close to the heart of TRS-80 users is the Tandy Newsletter. This selection offers information on product availability, hints and tips on equipment use, product descriptions and its own Feedback section where our customers can communicate directly with Tandy headquarters for the answers to their questions.

RAINDROPS KEEP FALLING ON MY HEAD . . .

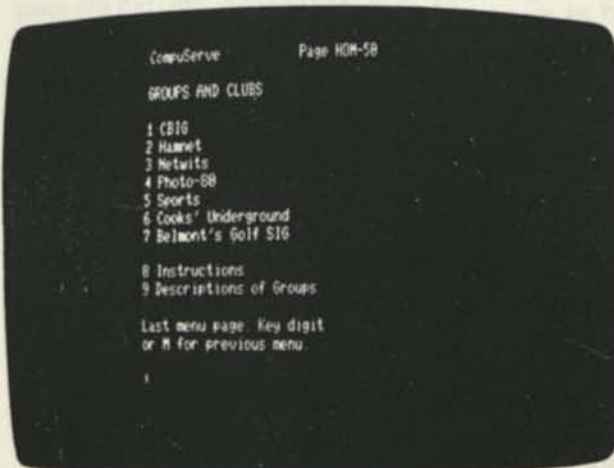
If you're going to discuss something with someone you don't know, it may as well be the weather. And CompuServe can have you discussing anything from whether it's going to rain in Topeka to the location of the nearest VORTAC station for the quick landing of your single engine aircraft.

CompuServe offers "Aviation and General Weather" on the Home Services menu. Public weather consists of Public and Marine weather reports written in plain English. The Aviation Weather is a service for pilots consisting of information taken from the National Oceanographic and Atmospheric Administration (NOAA) high-speed wire. There is help available for those of you who find you aren't able to read the listings with "flying" colors.

"Aviation" offers the hourly weather for different areas, terminal forecasts which contain data for specific airports and are issued three times daily; previous hourly weather, reports of weather conditions given by pilots in flight, and area forecasts, prepared twice a day for an area that may encompass several states.

PARLEZ-VOUS FORTRAN?

If weather is a little too broad a category for you, we offer a list of 11 computer languages for you to tinker with. CompuServe offers languages for the advanced computer hobbyist to the inexperienced novice including BASIC and XF4 FORTRAN, an extended FORTRAN language developed by CompuServe. Programming languages can be found by accessing Personal Computing Services.



IS IT SECURITIES OR BONDS? WE'VE GOT YOUR SERVICE IN "STOCK"

CompuServe has a vast array of financial services. The MicroQuote service is an historic securities database for people who follow the stock market or are exploring the world of finance. It's easy to use and is updated each day to provide fast access to a variety of information on securities traded on the exchanges and over-the-counter. MicroQuote offers historical data dating back to Dec. 31, 1973 on dividends, pricing and volumes.

There is also information on bonds including yields, maturity dates and Moody's ratings. Options information features prices, expiration dates and underlying stock prices. Depending on the information accessed, MicroQuote users must pay a small charge above the \$5 per hour CompuServe charge.

Another follower of the stock market is Quick Quote. This service provides high, low, closing, volume and net change figures on over 9,000 securities traded on the New York and American Stock Exchange and over-the-counter.

To use Quick Quote, you insert the ticker symbol or CUSIP (Committee on Uniform Security Identification Procedures) number, and the program gives you the latest information as reported from Wall Street.

Quick Quote also provides you with an easy way to find the name of a stock or whether or not it's being traded and through which exchange.

Another of our financial features is Value Line Database II. This product allows you to analyze the performance of more than 1,600 major industrial, transportation, utility, retail, banking, and insurance companies and includes detailed data from historical annual income statements, balance sheets, sources and uses of funds, as well as reported quarterly results.

Standard and Poor's is another financial product available under the Business and Financial Services section or by typing Go FIN-20. It's an easy way to obtain financial statistics on whatever company interests you. Included in this datafile are the separate features of the Business Summary, Important Business Developments, Product/Service Line and General Information.

WE'RE GAME IF YOU ARE

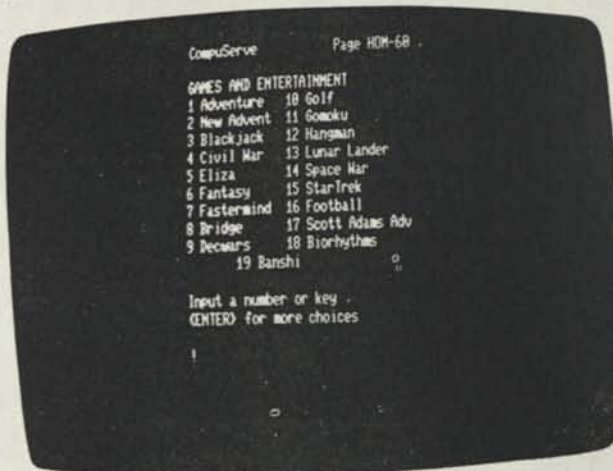
One of our more popular services with the young and old alike are the over 30 offerings of all types of games. Some of the selections include games of chance such as blackjack, craps or roulette, athletic games such as golf and football, and analytical or strategic games like Star Trek, backgammon and chess.

Other entertainment features are biorhythms, the cube solver and Eliza, an interactive program which features the computer in the role of the psychiatrist carrying on simple conversations with you, the "patient."

Two of the most popular games are Adventure and Megawars. Adventure uses your computer as your eyes and hands to battle snakes and dwarves in Colossal Cave in search of the lost treasure. Adventure has caused quite a stir with CIS customers and even has its own T-shirt, puzzle and two different Adventure maps to guide adventurers through the game (one is the advanced version), and a full-color poster depicting a confrontation between explorers and a large green snake in Colossal Cave.

Megawars is another popular game as a real-time space battle game where up to 10 players can play at one time. The object of this game is to choose either to captain a star ship or an enemy ship and thus proceed to attack your opponents' space ships and capture planets. Megawars also has its own accessories including T-shirts, brochures, and a full-color poster depicting a confrontation between Colonists and the Kryon Empire.

Adventure, Megawars and all the other CompuServe games can be found under Home Services, or by typing Go HOM-60.



HOW DO YOU SPELL CONVENIENCE?

Two features which welcome you to someday are the banking and shopping at home features. Customers may shop from the comfort of their own home through Comp-U-Store, our interactive electronic shopping service developed in conjunction with Comp-U-Card, a Connecticut-based company which allows customers to order merchandise electronically to be delivered to their homes. Comp-U-Store offers consumer products in a format which allows customers to either browse through or review items by description.

A customer can check price, features and receive information about a specific item all through Comp-U-Store. Comp-U-Store allows customers to search and compare shop electronically.

CompuServe, Radio Shack and United American Bank Service Corporation have joined forces to offer Bank-At-Home service. Bank-At-Home service permits customers the United American Bank to pay most of their bills, receive current bank information and view their accounts displayed on their screens. Bank-At-Home will also offer computerized bookkeeping and tax record services, and the ability to apply for loans on-line.

AND THERE'S STILL MORE, MORE, MORE, MORE, MORE . . .

The CompuServe Information Service offers beneficial subscriber services. These services include two publications which we provide in addition to CIS.

Update is our monthly newsletter which informs you about new products, programs, telephone number additions and corrections, plus other timely information. Update provides you periodically with the complete CIS subject index as an on-hand reference.

Today is a full-color magazine which offers a more in-depth review of new services, information providers to CIS, customers and how they use the service and special announcements. Today is free to CIS subscribers and is available to noncustomers through subscription.

CompuServe's free on-line Feedback service allows you to communicate with CompuServe's Customer Service department. Any problems which you are experiencing can be quickly attended to through Feedback found under User Information.

In addition to our Feedback section, new customers to our system receive a free hour of CIS time with the starter kit purchased at any Radio Shack outlet. This free hour is of course accompanied by your own User ID, secret password and user's guide.

For those of you who have questions regarding microcomputers and personal computing, CompuServe offers the Micro Advisor. The Micro Advisor is a clearinghouse for information to aid the experienced or novice user. Micro Advisor discusses equipment usage, selection, peripheral devices and software. Micro Advisor utilizes its own user Feedback service to answer questions or to comment on groups and the latest developments in the microcomputing industry.

CompuServe also offers Access. This service is for those of you who want to share programs which you have authored with other CIS users. Access contains a wide selection of user-contributed programs, games and text files.

OUR CIS NETWORK . . . AND WELCOME TO IT.

CompuServe Information Service is known as one of the largest information and entertainment services available to home computer and computer terminal owners across the country. And CompuServe Incorporated goes even further by offering the benefits of its high quality communications services through its value-added network in 130 cities.

Our customers using this network are connected to a system of communications lines which we've leased in order to offer them the most efficient service we can develop. Customers with personal computers and terminals can dial a local phone number and hook up to our network. Because we control the growth of our network, we have the ability to offer a smoothly-running, error-free and always operational network. If there's a problem, we're able to detect it and either repair it or reroute our customers' data to ensure minimal interruption of service.

U.S. customers in cities we don't reach access CIS through the TYMNET system, a common carrier network. Customers accessing through TYMNET are charged an additional \$2 per hour communication surcharge making the use of the service \$7 an hour rather than the usual \$5 an hour.

CompuServe plans to expand to 300 cities by 1983 to provide our customers with good quality communications in their city. It probably won't be long before we're in your neighborhood, too.

THINK YOU MIGHT BE CONVINCED?

All of the above information is just our way of introducing you to the CompuServe Information Service. We think you'd like having all this convenient information available from your living room, accessible through a local phone call for just \$5 an hour.

CompuServe is proud of its comprehensive service, but we don't plan to stop here. We're researching and discovering new services and methods of operation every day in order to offer you even more than we've ever had before. And when we find these discoveries, we let you know through our system announcements, *Update*, and *Today* magazine. After all, we are an information, communications service. And we're bullish on keeping you informed at all times.

Go ahead and take advantage of us. Take advantage of our convenient information, the convenience of using us in your own home and accessing our ever-growing databases. Purchase CompuServe Information Service. We're found at any Radio Shack locations around the country and in Canada.

Questions and comments about the CompuServe Information Service can be sent to Richard A. Baker, Editorial Director, CompuServe Information Service, 5000 Arlington Centre Blvd., P.O. Box 20212, Columbus, Ohio 43220 or through Feedback, main menu item 5, CompuServe User Information.

Computer Clubs

East Coast Computer Club
% James Law Conway
198 Old River Road
Wilkes-Barre, PA 18702

Rhode Island Tandy Users Group (RITUG)
% McDonald & Ferdinandi
1441 Park Avenue
Cranston, RI 02920
(401) 943-1801

Western Massachusetts Computer Club
% Phillip C. Jachem
2275 Westfield Street
West Springfield, MASS 01089
(413) 736-3521 or (413) 592-2023



The Academic American Encyclopedia—A to Z Electronically

Editor's Note: With over 45,000 subscribers, Dow Jones News/Retrieval is the leading provider of online business and financial information. We will keep you up-to-date on new software, new data bases and new prices in upcoming issues of TRS-80 Microcomputer News. Dow Jones News/Retrieval is sold at Radio Shack stores nationwide and in Canada.

There is a new kind of encyclopedia these days—an encyclopedia that is never out of date and is always easy to handle. It is both original and widely praised. It is an electronic encyclopedia, and News/Retrieval has it.

With the new electronic edition of the highly respected Academic American Encyclopedia already online for subscribers, News/Retrieval has become the first electronic information service to offer a highly praised encyclopedia to the consumer in a form he or she can quickly grasp and easily use.

Because electronic publishing permits limitless page expansion and constant re-editing, the new encyclopedia can be completely updated whenever necessary—not with separate, printed handouts, but with actual changes in the original text to provide the most current information available. New articles will be added and others revised to supply adults as well as students with a current, comprehensive, authoritative reference tool for the home or office.

In addition to these revisions, the capabilities of electronic publishing will allow News/Retrieval journalists to cross-reference other data bases with the encyclopedia, directing users to the Academic American for background on breaking news and sports developments.

The electronic currency of the encyclopedia data base is blended with scholarly research of more than 2,500 distinguished contributors and editors. Produced by Arct Publishing Company, the original 20-volume Academic American contains more than nine million words in 28,000 carefully researched and concisely written articles on a vast array of subjects.

The first new multi-volume encyclopedia in more than a decade, this reference work received excellent reviews when it was introduced in 1980. The Encyclopedia Buying Guide praised the Academic American as "an innovative, forward-looking encyclopedia." The set was found to be "meticulously researched . . . both comprehensive and well organized," by Booklist, an American Library Association journal. Adds Library Journal: "The Academic American succeeds admirably . . . It is up-to-date, concise, authoritative, and well documented . . . Its most outstanding attribute is its timeliness."

Reviewers have been particularly impressed by the Academic American's coverage of volatile subjects—political changes around the world, scientific backgrounds, sports and industry trends—all of which are accompanied by the historical, philosophical or scientific background needed to put these developments in perspective. "In its treatment of bias-prone areas, the editors are successful in presenting balanced, open statements, free from editorializing," said Booklist, citing as evidence the entries covering the Ku Klux Klan, Khomeini, Sun Myung Moon, and Terrorism.

In the encyclopedia's regular revisions—which initially will be made twice a year—many new articles, especially biographies, will be created. And original articles will be revised to reflect new facts and information. Grolier Incorporated, the world's largest publisher of multi-volume reference sets, will produce these updates, with the first planned for early autumn. Grolier recently acquired the long-term right to publish the Academic American in both its printed and electronic forms.

In addition to these continuing updates the encyclopedia will be made relevant to daily current events by the News/Retrieval staff. Newsroom writers and editors, keeping abreast of both the news and relevant encyclopedia information, will cross-refer news, sports and business readers to entries in the Academic American. An example was this summer's Wimbledon tennis coverage. On the day that saw Billie Jean King reach the semifinals, a sentence below the story in News/Retrieval Sports read: "For a closer look at Billie Jean King's tennis career, see King, Billie Jean in //ENCYC. Such cross-referencing is one way News/Retrieval interre-



lates all its data bases to provide the in-depth coverage a complicated world requires.

The building of an information service capable of that coverage was the first consideration of News/Retrieval editors when they selected the Academic American as the encyclopedia they would put online. The Academic American is the first encyclopedia specifically conceived for electronic publication, with complete but relatively concise articles, and fact boxes and abundant statistical information designed for easy and accurate updating. The prose style, as described in a review by *Choice* magazine, "is commendable, is well written, concise, and informative without being condescending or obscure." Also, many articles conclude with bibliographies which have won special praise. *Booklist* called them "appropriate, current, and skillfully selected".

To make the electronic encyclopedia easy-to-use, News/Retrieval editors and computer specialists spent the past year preparing the Academic American for electronic distribution. For example, the chemical notation for water (H₂O) had to be written out completely since most computers cannot handle below-the-line notation. The methods for getting into the encyclopedia were painstakingly devised for exceptional ease of use.

Data base organization, search strategies, commands and help messages all have been tested on secondary-level students, many of whom had no experience with computers or online information services. This "child-proofing" of the electronic encyclopedia has made it easier for everyone to use, from junior high school level on up.

The electronic encyclopedia has been designed to resemble the familiar "subject outline," leading students from the broad to the specific. In effect, it helps teach children how to begin and develop a research project. The online bibliographies and various types of cross-references make it easy for users to find additional information.

Adults and students will find the electronic encyclopedia invaluable for quick fact checks. For example, detailed tables on major athletic events will help sports fans settle arguments. And special fact boxes about countries and states highlight the area's population, principal products, languages, climate, levels of education, and infrastructure.

The encyclopedia is part of the General News and Information Services group of data bases and is accessible by typing //ENCYC. It costs standard subscribers 60 cents per minute in prime time and 30 cents per minute in non-prime time.

//ENCYC How-To

There are two ways to find information in the electronic encyclopedia. After typing //ENCYC and pressing the <ENTER> key to access the encyclopedia you can "search by partial title of a subject heading" or "search by the complete title of a subject heading."

SEARCHING BY PARTIAL TITLE

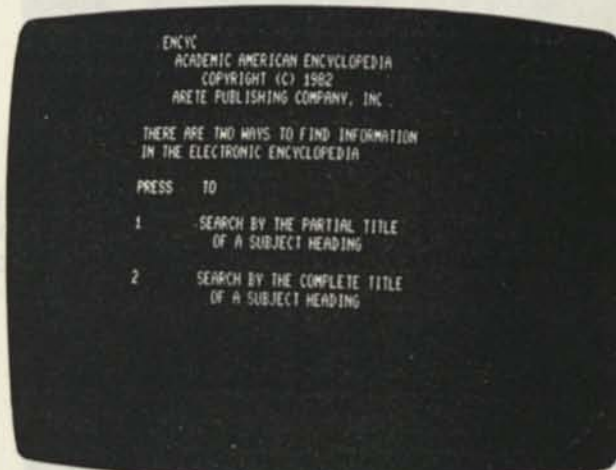
Searching by partial title, the method called for in most cases (exceptions will be discussed below), is little different from turning to a subject in a familiar, printed encyclopedia.

If you were in a library reference section searching for information about automobile drag racing you would pick up the "A" volume and flip through the pages until you found the

area of words whose beginning was AUTO, then AUTOM, then AUTOMO. Having narrowed your search area, you would look through headings beginning with AUTOMO until you found your chosen topic, automobile racing. Likewise with the electronic encyclopedia.

A SAMPLE SEARCH USING THE PARTIAL TITLE METHOD

After accessing the electronic encyclopedia you will be offered a choice of two search methods. Type 1 and press <ENTER> to search by the partial title.



The display screen will show ENTER QUERY. If you type AUTOMOBILE DRAG RACING and press <ENTER>, the display will say there is no title to match your query. This means your query either was too specific or doesn't have a main heading in the encyclopedia.

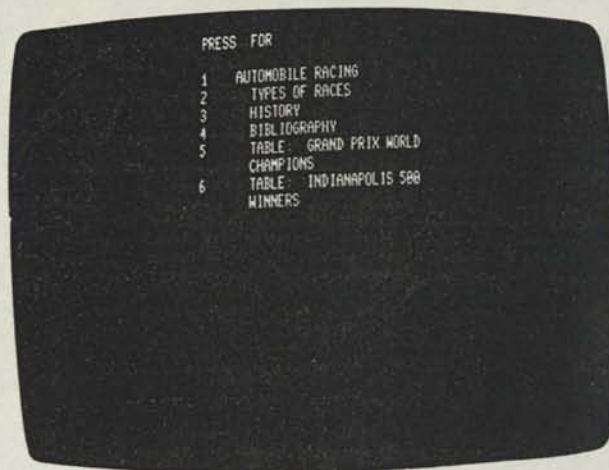


If you type only AUT (the beginning letters of the subject sought) and <ENTER>, the display will request a more specific query because the range of titles beginning with AUT is too broad.

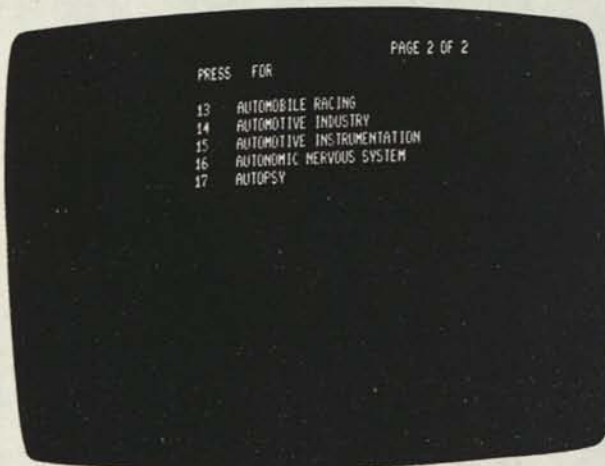
Add another letter in a new query—type AUTO and press <ENTER>. The display then will show a numbered list of titles beginning with those letters, titles from AUTOWORKERS and AUTOBIOGRAPHY to AUTONOMIC NERVOUS SYSTEM and AUTOPSY. Altogether there are 17 titles in the list. AUTOMOBILE RACING, number 13, is the title



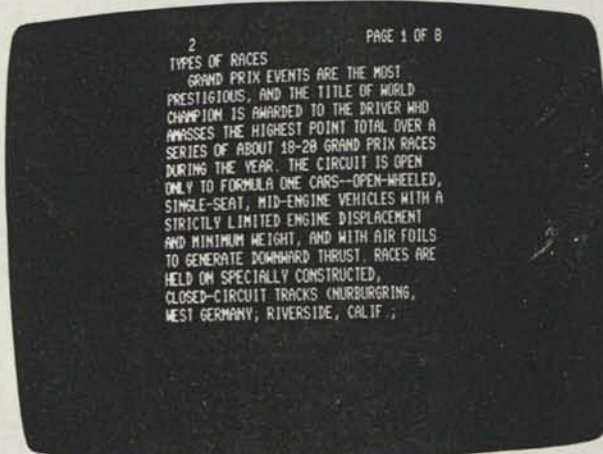
most likely to include information on automobile drag racing, the subject sought.



Type 2 and <ENTER>. The display will show page one of an eight-page article written under TYPES OF RACES. Press <ENTER> to proceed to the following page. Information on drag racing starts on page five.



Type 13 and <ENTER>. The display will show a list of six subheadings, number two of which is TYPES OF RACES.

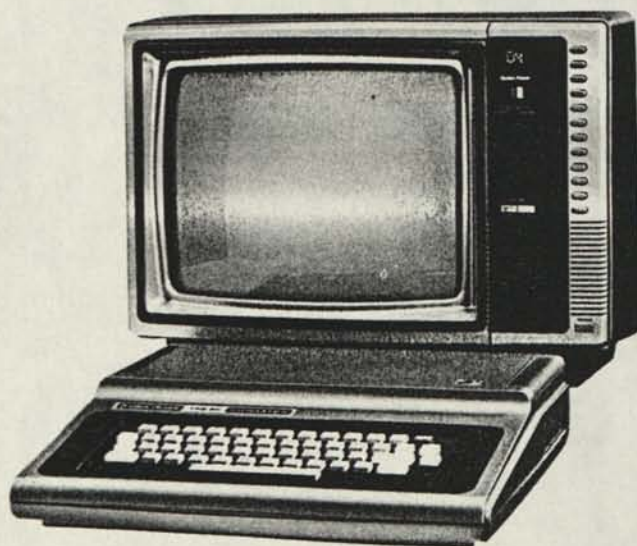


SEARCHING BY COMPLETE TITLE

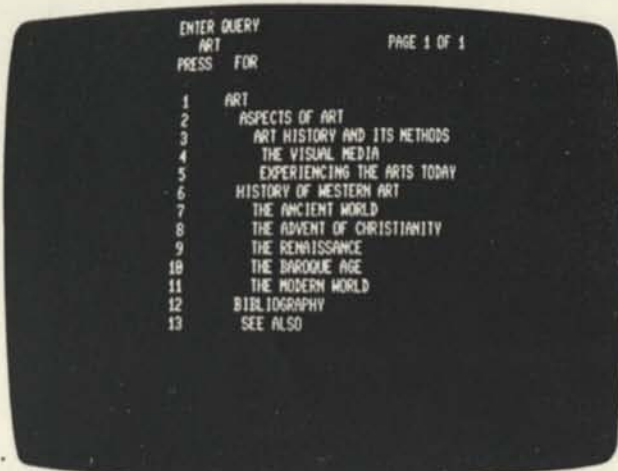
Sometimes the commonly used partial-title search method won't work. For instance, if you type ART and press <ENTER> while using the partial title search method, the display screen will say PLEASE MAKE YOUR QUERY MORE SPECIFIC. It will go on to suggest that you enter additional letters or words or that you try searching by complete title, if the title of the subject you seek is very short.

Since you can be reasonably certain that most encyclopedias contain articles about ART, you wouldn't want to add any more letters or words to your search query. So in this case—trying to find a subject with a short title—you should try a complete title search, which is the second way to search for information in the electronic encyclopedia.

To look up "art" by using the complete title search method, you would have begun your search using the search by partial title search. Type T and press <ENTER> to get to the top encyclopedia menu to change your search method. When the display offers the two search methods, type 2 and press <ENTER> to search by the complete title. When the display shows ENTER QUERY, type ART and press <ENTER>.



Under ART, the display will show a menu listing 13 different categories. Notice that two categories, ASPECTS OF ART and HISTORY OF WESTERN ART, are indented slightly. These are broad topics, corresponding to the upper-case letters in a conventional subject outline, which include information on the subcategories indented below them. If you want to read everything in the encyclopedia on art, including all the categories and subcategories listed on the menu as numbers 2 through 13, type 1 and press <ENTER>.



If you want only the information specified by a certain category or subcategory, type the corresponding number and press <ENTER>. If you were to type 6 and <ENTER>, for HISTORY OF WESTERN ART, you would get information on all the subcategories indented below it: THE ANCIENT WORLD, THE ADVENT OF CHRISTIANITY, THE RENAISSANCE, THE BAROQUE AGE, and THE MODERN WORLD.

SOME BASIC COMMANDS

Press <ENTER> to go forward. This will always take you to the next page if there is more than one in a search.


Type R and press <ENTER> to go backward (reverse) to every page except the menu.

Type T and press <ENTER> to get to the top encyclopedia menu to change your search method.

Type Q and press <ENTER> to start a new search or enter a new query.

Type M and press <ENTER> to go to the previous menu.

Type P (NUMBER) and press <ENTER> to go to a specific page in an article. For example: type P12 and press <ENTER> to go to page 12.

Type //ENCYC HELP if you find yourself in trouble. It will display information that will help you use the electronic encyclopedia. 

If you are not a Dow Jones News/Retrieval service subscriber but are interested in this or other data bases, call the Dow Jones Customer Service hotline number: 254-5114. It's toll free.



Coded Message

Peter L. Vogel
Notre Dame Regional Secondary
2855 Parker St.
Vancouver, B.C.
Canada
V5K 2T8

I have found David Snyder's Hex/ASCII problem, in the March 1982 Newsletter, to be an excellent exercise for students. The short program which I have written is a follow-up to Mr. Snyder's program.

This short program takes the concept one step further by automating the decoding process. A problem of this type would make an excellent short programming assignment.

The language used is Level II Microsoft BASIC.

```

100 REM HEX DECODER
110 DIM HEX$(15)
120 FOR I=0 TO 15
130 READ HEX$(I)
140 NEXT I
150 DATA 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
160 READ EX$
   : REM EX$=HEX EXPRESSION
170 FOR I=1 TO LEN(EX$) STEP 2
180 X1$=MID$(EX$,I,1)
   : X2$=MID$(EX$,I+1,1)
190 FOR J=0 TO 15
200 IF HEX$(J)=X1$ THEN 220
210 NEXT J
220 X1=J
230 FOR J=0 TO 15
240 IF HEX$(J)=X2$ THEN 260
250 NEXT J
260 X2=J
270 AC=X1*16+X2
   : REM AC=ASCII CODE
280 PRINT CHR$(AC);
290 NEXT I
300 DATA 434F4E47524154554C4154494F4E5321


```

SCRIPTSIT Ideas

William Ashworth
201 Gresham
Ashland, OR 97520

The method you present, in the April 1982 Newsletter, of creating hanging indents for Model I/III is unnecessarily complicated and time-consuming. A better way is as follows:

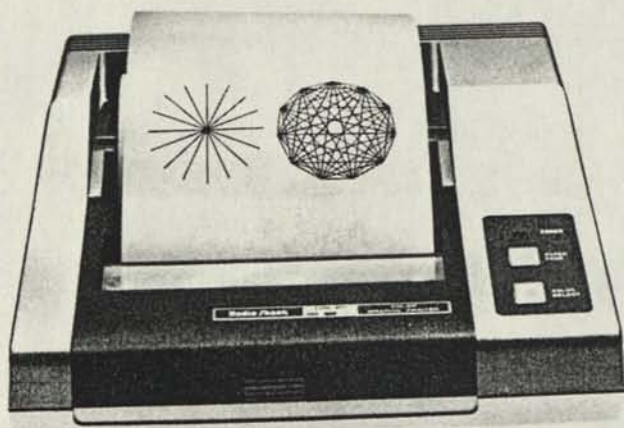
1. Set the cursor on the line BEFORE the line you wish to indent.
2. Use the TAB function to tab all the way to the right-hand end of the line. This prints spaces following the text right to the end of the 60-space line (or whatever your machine is set for).
3. Hit the space bar once. The space will print at the beginning of the next line, opening a new line and moving the text material beyond that point down one line.
4. TAB over to the spot where you want the indented material to begin, and punch CLEAR.

This process creates hanging indents very rapidly. It also avoids having to move the cursor from line to line manually to begin each new indent; after you have CLEARed the line into its indented location, the cursor is already positioned on the spot you want to begin Step One for working the next line. 

Color Graphic Printer

When the Pocket Computer 2 was introduced in January of this year, one of the things people spent the most time looking at was the printer. The printer was small, compact and featured X-Y plotting and four colors. You could tell that many of these people were wondering how to interface the printer to their larger computers.

With the introduction of the Color Graphic Printer (CGP-115 26-1192 Suggested Retail Price \$249.95) Radio Shack is providing the same technology with a wide 4 1/2" paper and dual interfaces (serial or parallel). The CGP-115 uses standard LPRINT commands on Models I/III, II, 16, and uses PRINT#-2 for Color Computers. The CGP-115 can also be used with the DT-1 Data Terminal.



OPERATION MODES

The CGP-115 has two operating modes, Text and Graphics as well as a self-test mode. The self-test mode is entered holding down the Paper Feed switch while turning on the printer.

TEXT MODE

This is the mode the printer powers up in. In the text mode, the CGP-115 print density is determined by a DIP switch which is read by the printer when the power is turned on.

Using the DIP switch (with power off), you can select either 40 or 80 characters per 4" line.

The text mode is used for note writing, program listings and most program output. In this mode, you can use the following commands to control the printer:

- CHR\$(8) - Backspace one character
- CHR\$(11) - Reverse Line Feed
- CHR\$(18) - Select Graphic Mode
- CHR\$(29) - Change Pen color

GRAPHIC MODE

In the Graphic mode, the following commands are available:

- CHR\$(17) - Select Text Mode
- A - Reset Pen. This moves the pen back to the left margin without drawing a line. The "A" command also selects the Text mode.
- Cn - Change Pen Color to the one specified by 'n'. In Text mode you must keep track of the current pen color since your only color command is to rotate the pen to the next color. In the Graphic mode, you specify which of the four colors (0-3) you want.
- Dx,y - Draw a line from the current pen location to the X,Y coordinate location specified by 'x,y'. Multiple points can be specified, resulting in connected lines.
- H - Moves the pen to the current origin without drawing a line.
- I - Sets the current Pen location as the new origin.
- Jx,y - Draws a line from the current Pen position to a point 'x' horizontal steps and 'y' vertical steps away. If 'x' is positive, the Pen moves to the right and if 'x' is negative, the Pen moves left. A positive 'y' moves up while a negative 'y' moves down the page. Multiple 'x,y' destinations can be specified with a single 'J' command.
- Ln - Change the line type to the type specified by 'n'. 'n' can be a value from 0 (solid) to 15.

Line Type Chart

Line Type 0	_____
Line Type 1	_____
Line Type 2
Line Type 3
Line Type 4
Line Type 5
Line Type 6
Line Type 7
Line Type 8
Line Type 9
Line Type 10
Line Type 11
Line Type 12
Line Type 13
Line Type 14
Line Type 15

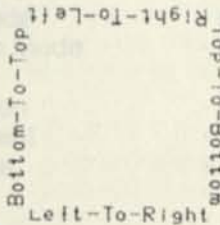
```

20000 LPRINT "          Line Type Chart"
20010 LPRINT
20020 FOR I=0 TO 15
20025 LPRINT CHR$(18)
20030 LPRINT "PLine Type";I
20050 LPRINT "I"
20060 LPRINT "L";I
20070 LPRINT "D300,0"
20090 LPRINT CHR$(17)
20110 NEXT

```

- Mx,y** - Move (without drawing a line) to a position 'x' horizontal steps and 'y' vertical steps from the current origin. Note that this move to an absolute position based on current origin, not on the current pen position.
- Ptext** - Print the text which follows the 'P' starting from the current pen location, using current character size and direction.
- Qn** - Change the print direction. 'n' can be a value from 0 to 3:
 0 - Normal left-to-right
 1 - Top-to-bottom
 2 - Upside down, right-to-left
 3 - Bottom-to-top

Print Direction Chart



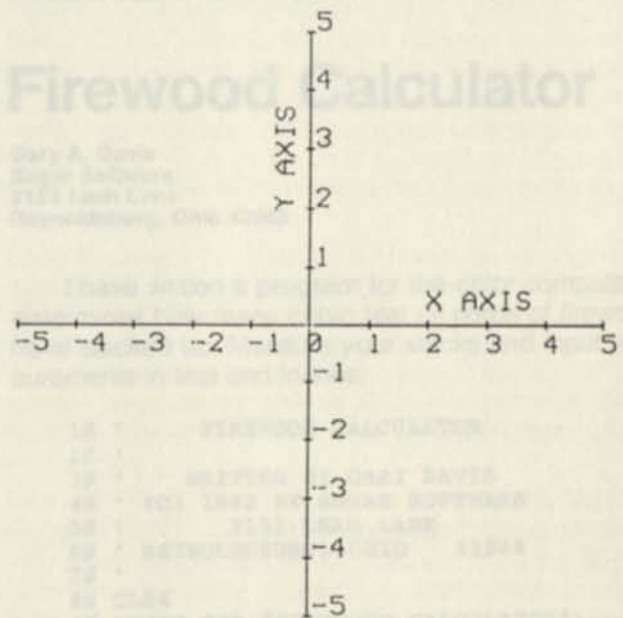
```

30000 LPRINT "          Print Direction Chart"
30010 LPRINT STRING$(3,13)
30020 LPRINT CHR$(18)
30030 LPRINT "M200,-100"
30040 LPRINT "I"
30050 LPRINT "M-50,-50"
30060 LPRINT "Q0"
30070 LPRINT "PLeft-To-Right"
30080 LPRINT "Q1"
30090 LPRINT "M100,120"
30100 LPRINT "PTop-To-Bottom"
30110 LPRINT "Q2"
30120 LPRINT "M95,135"
30130 LPRINT "PRight-To-Left"
30140 LPRINT "Q3"
30150 LPRINT "M-55,-35"
30160 LPRINT "PBottom-To-Top"
30170 LPRINT "Q0"
30180 LPRINT "A"
30185 LPRINT STRING$(10,13)
30190 LLIST 30000-

```

- Rx,y** - Move (without drawing a line) to a position 'x' horizontal steps and 'y' vertical steps from the current Pen position. Note that this is a move relative to the current position, not the currently defined origin.
- Sn** - Sets the current character size for use with the 'P' command. 'n' can be a value from 0 (80 CPL) to 15 (5 CPL). To determine how many characters per line a given value of 'n' yields, use the following formula:

$$CPL = \text{INT}(80/(n+1))$$
 where CPL stands for Characters Per Line.
- Xa,s,i** - The 'X' command is used to draw an X-Y axis from the current Pen position. 'a' specifies which axis (0=Y, 1=X) is to be drawn. 's' specifies the number of steps between interval marks, and can have a value between -999 and 999. 'i' specifies how many intervals there will be (how many times 'step' is to be repeated), and can have a value between 1 and 255. If 's' is negative, an X axis would be drawn to the left, or a Y axis would be drawn down from the current Pen position.



Here is the program that produced this chart:

```

10 REM X-Y AXIS TO BE DRAWN
20 LPRINT: LPRINT CHR$(18): REM GRAPHIC C
30 LPRINT "M240,-240" : REM MOVE
40 LPRINT "I" : REM SET NEW ORIGIN
50 LPRINT "X1,40,5" : REM RT HALF
60 LPRINT "H" : REM RETURN TO ORIGIN
70 LPRINT "X1,-40,5" : REM LF HALF
80 LPRINT "H"
90 LPRINT "X0,40,5" : REM TOP HALF
100 LPRINT "H"
110 LPRINT "X0,-40,5" : REM BTM HALF
120 FOR I=-200 TO 200 STEP 40

```

```

130 LPRINT "M"; I; ",-20" : REM LABEL X
140 READ A$ : REM READ LABEL
150 LPRINT "P" A$ : REM PRINT LABEL
153 IF A$="0" THEN 160 : REM 1 ZERO
155 LPRINT "M5,"; I : REM LABEL Y AXIS
157 LPRINT "P" A$ : REM PRINT LABEL
160 NEXT
170 LPRINT "M80,10" : REM MOVE
180 LPRINT "PX AXIS" : REM PRINT
190 LPRINT "M-10,80" : REM MOVE
200 LPRINT "Q3" : REM ROTATE PRINTING
210 LPRINT "PY AXIS" : REM PRINT
220 LPRINT "M0,-250"
999 LPRINT "A" :REM RESET, MOVE LEFT
1000 DATA -5, -4, -3, -2, -1, 0
1010 DATA 1, 2, 3, 4, 5

```

SPECIFICATIONS

Plotting/Printing System	- 4 Ball Point Pens
Plotting Speed	
Horizontal	- 52mm/sec (2.05 ips)
Vertical	- 73mm/sec (3.8 ips)
Printing Speed	- 12 Characters/sec
Resolution	- 0.2 mm/step
Effective plotting range	- X axis - 96mm (480 steps)
	- Y axis - Length of the paper
Characters Per Line	
Text Mode	- 40 or 80
Graphic Mode	- 5 to 80
Accuracy	
Repetition	- 0.2mm max
Movement	- 0.3mm max
Distance	- X axis 0.5% max
	Y axis 1% max
Dimensions	
Width	- 210mm (8.4 in)
Depth	- 216mm (8.64 in)
Height	- 75mm (3 in)
Pens	
3 Black Pens	- Radio Shack 26-1480
1 ea Blue, Green, Red	- Radio Shack 26-1481
Pen Life	- 250 meters (825 ft)
Paper	- Radio Shack 26-1428
Parallel Interface	- 8-bit parallel
	BUSY handshaking,
	STROBE, and
	ACKNOWLEDGE
Serial Interface	- RS-232C Using DATA
	and BUSY. 600 Baud,
	7-bit, no parity, two
	stop bits


Heart Health Programming Contest

In keeping with its reputation as a leader in promoting heart health education in the young, the American Heart

Association, Greater Boston Division, is taking a new approach to education—an electronic approach. The American Heart Association in conjunction with **Classroom Computer News** has announced a "Heart Health Computer Programming Contest" to solicit the development of programs on health education for eventual national distribution to schools.

Programs should be aimed at an elementary, junior high or high school audience, and should deal with ways in which heart disease can be prevented. The Surgeon General of the United States has recommended prevention of heart disease as a national priority and the Heart Association has started its campaign to educate youngsters about the importance of good habits early in life.

The best program entry has the potential to receive \$1500.00; eleven other entries may receive \$500.00. Programs are to be written in BASIC language to run on the Apple, Atari, TRS-80, Texas Instruments or the Commodore PET microcomputers. All entries must be received by January 31, 1983. A panel of computer and education experts will act as judges and reach a decision by Spring 1983.

For further information, write to "Heart Health Computer Programming Contest", American Heart Association Greater Boston Division, 33 Fourth Avenue, Needham Massachusetts, 02194, before December 1, 1982. 

Firewood Calculator

Gary A. Davis
Sugar Software
2153 Leah Lane
Reynoldsburg, Ohio 43063

I have written a program for the color computer which determines how many cubic feet or cords of firewood you have stacked up. Measure your stacks and input the measurements in feet and inches.

```

100 ' FIREWOOD CALCULATOR
200 '
300 ' WRITTEN BY GARY DAVIS
400 ' (C) 1982 BY SUGAR SOFTWARE
500 ' 2153 LEAH LANE
600 ' REYNOLDSBURG, OHIO 43068
700 '
800 CLS4
900 PRINT @38,"FIREWOOD CALCULATOR";
1000 PRINT @96,"YOU MUST ENTER BOTH FEET AND
1100 PRINT "INCHES (SEPARATED BY A COMMA)"
1200 PRINT "FOR EACH PROMPT."
1300 PRINT @224,"ENTER WIDTH"
1400 PRINT "ENTER LENGTH"
1500 PRINT "ENTER HEIGHT"
1600 PRINT @236,;
:INPUT WF,WI
1700 PRINT @268,;
:INPUT LF,LI
1800 PRINT @300,;
:INPUT HF,HI
1900 CF=(WF+WI/12)*(LF+LI/12)*(HF+HI/12)+CF
2000 PRINT @384,;
2100 PRINT USING "####.# CUBIC FEET, ##.#
CORDS";CF,CF/128
2200 GOTO 1300

```

Profile III+ : Introducing The Flexible Data Base Management System

The small Computer Company
P.O. Box 2910
Fort Worth, TX 76113-2910
By Ivan Sygoda, Director, Pentacle

Pentacle is a New York City-based non-profit service organization specializing in administrative services such as bookkeeping, grant-writing, tour management, public relations and booking representation for performing arts groups. We keep our member companies going while their artists are on stage, on tour, or in the rehearsal studio.

We've been running PROFILE III+ on our two-disk Model III (48K) for more than six months. So far, the program has made itself indispensable in two ways: 1) By enabling us to do certain jobs more efficiently, our small staff has been able to accomplish more for each member and at the same time offer our services to new members. 2) We have been able to create new services that we simply couldn't handle before.

We've used PROFILE III+ to create a half-dozen completely different files to serve different needs. Along the way we've learned a few tricks which should prove useful in other applications. I'll describe them as I go along.

WHY WE USE PROFILE III+

Mailings: One of our jobs is to send out press releases about the companies' performances and activities to a press-list of over 400 critics and publications. We used to use copier labels, which are expensive and look sloppy, especially after a few months of accumulated additions, deletions and corrections. PROFILE III+ lets us keep the list "clean" in both senses of the word. Saturation mailings which can be sent bulk-rate come out in zip code order, ready to go to the Post Office.

And we can use the selection feature to target sub-sets of the complete list for first-class mailings, saving money because there are no unused labels.

Trick: We use a four-character key field for each press-list record. The first character of the four identifies the entry as dance, theater or music press. The second identifies the entry as a monthly, weekly, daily and so forth—important because these different types receive releases according to different rhythms. The third character identifies the addressee as being on the "A" or "B" list or as accepting photos. The last character distinguishes between three- and four-line addresses so that the labels look their best. We do two runs—first we use the selection operation to pull out all the four-line addresses, then we use it to pull out the three-line addresses.

ACCESSING PROFILE DATA FROM AN INDEX SEQUENTIALLY

In this part of our column, we'll show you how to retrieve data from Model II PROFILE into BASIC programs. We've written a program that will let you access PROFILE data sequentially from a PROFILE index file or with BASIC programs. You can also use the program with BASIC programs. BASIC programs using the program...

We can select with incredible flexibility using the "wild card" (=) character: To send information to the entire "A" list, we simply type =A= or ==A (in this case, the computer treats a blank and the wild-card symbol following the selection character the same way).

PROFILE III+ works so efficiently as a mailing list generator that it is now possible for us to manage the member companies' individual mailing lists for them, tailoring the lists to their particular needs—people who take dance classes, VIPs who might donate money, people who helped in a special project. The password protection offered by PROFILE III+ makes the companies confident in the security of this precious resource.

Fund Raising: We've made a special file for our own fund-raising solicitations, and that's password-protected to the hilt! PROFILE III+'s math capability lets us tally an individual's contributions during the year and then total them easily for end-of-year financial statements. People who donate certain amounts are entitled to various perks, and the PROFILE sort feature makes it easy to pluck them out for special mailings.

Trick: PROFILE III+ can interface with Radio Shack's new SuperSCRIPSIT to create personalized letters to these very important people. We anticipated this from the beginning, devoting a field to a salutation appropriate to each individual. It's silly to "individualize" a letter to Mrs. Smith when it's Aunt Millie or an old acquaintance you're writing to.

Scheduling: Another important part of our work is to keep track of our companies' performing activity and schedules, both past and future. As we deal with over 30 groups, B.P. (Before Profile) it was very tedious to collect and arrange this information every time we had to do a grant application or final report. Each public funding agency requires a different assortment of these statistics. We designed a PROFILE III+ file which tells us the dates of each tour, the name of the theater, the fee paid, the type of performances and workshops involved and the attendance figures. PROFILE's flexible report formats let us tailor print-outs for each grant application.

Trick: By using the YY/MM/DD format, the tours come out in chronological order, and it's simple to summarize the information for a particular period or fiscal year.

Booking: Our first application of PROFILE III + , and the reason we haunted our local dealer until the program was available, is the most important of all. We try to get bookings for our touring groups, which means keeping track of hundreds of potential presenters across the country and abroad. Calls go in and out all the time, interruptions interrupt other interruptions, crucial information (leads, contacts, phone numbers possible dates and fees) gets scribbled on pieces of paper. Even a mild breeze could spell wasted effort or disaster.

PROFILE III + lets us keep track of these bits of information, and nothing except a direct hit by a tornado could scatter them again. The information is all in one place, legible and accurate; plus we always know we're looking at the latest data. Not only does this system insure that I know what I'm talking about when I'm dealing with presenters and company heads, but it makes it possible to use assistants more effectively because the information is ordered and legible.

Here's an application which really flexes PROFILE's muscles—keeping on top of the hundreds of presenters interested in one or more of over 20 performing groups. A trick we learned from the airlines is to reduce each company's name to a three-letter airport-style code. For instance, James Cunningham's Acme Company becomes ACM. Then we set up associated fields to hold the codes of the company or companies in which the presenter is interested. So long as ACM is in any one of these associated fields, the record will surface when needed.

DESIGNING YOUR FILES

It's important to think through your file design carefully—the key fields especially, since later you will use them to sort and select records. In planning our presenter file, we first listed all the types of information we wanted to have on file and made preliminary assessments of the field lengths required for each.

Use Worksheets: This is also the time to anticipate screen, label and especially report formats. A good idea is to block out the formats on the video/program worksheets available from Radio Shack (cat. no. 26-2105). The program side of the worksheets has exactly 80 spaces across, counting the one between the listing and the variable sections—perfect for laying out report formats for your 80-column printer. Using the format sheets, you might find that by shortening a name field one or two characters, you'll have room for both address and phone number on the same line. (Don't forget to allow a space between fields.)

Choose Key Fields Carefully: We made the area code a key field because it's easier to use for geographic targeting than zip codes. The phone number itself is relegated to another segment. As for zip code, we allocated 10 characters—the new nine-digit codes have a dash. Also, since we deal with foreign presenters, we can use the zip field for country names (in this case, use the alphanumeric instead of the numeric indicator for the zip code field on the screen format). This works perfectly for both labels and sorted reports.

Our key segment has 34 fields! This means that I can walk up to my Model III and say, "Spit out mailing labels for all directors of student activities in Ohio and Illinois who booked the Acme Company in the last five years, and then send them personalized letters offering special rates on a return engage-

ment." As long as I'm typing in the instructions at the same time as I say them, the computer will do exactly that.

This power and flexibility gets turned into increased opportunities for artists to perform—which is what it's all about. PROFILE III + can do the same for your business.

PROFILE Editor's Note: Mr. Sygoda is writing a series of 'how-to' PROFILE III + articles, which will be published over the next few issues in this column. We hope that you enjoy this new feature, and we look forward to your comments and questions on PROFILE III + .

ACCESSING PROFILE DATA FROM AN INDEX SEQUENTIALLY

In this part of our column, we'll show you how to move data from Model II PROFILE into BASIC programs.

We've written a program that will let you access PROFILE data sequentially from a PROFILE index for use with BASIC programs. You can also use the program as a module, building a BASIC program around it. The program shown is a demonstration program only—you should adapt it for your own purposes. Some possible uses:

- * You have an inventory analysis program written in BASIC and you want to pull out data from your PROFILE stock file.
- * You have a drug side-effects BASIC program and you want to access PROFILE patient information.
- * Before you started your business, you analyzed your expected customer base; now you want to analyze your actual customer base with the same program to check the accuracy of your projections.

A few notes are in order. This demonstration program accesses data only from the key segment, which is the most difficult to access because it must be unblocked. If you want to access non-key data, include statements between lines 800 and 900 reading as follows:

```
OPEN "D", 3, FS + "/DAT", nnn
```

(where "nnn" is the length of the segment and "/DAT" is the second data segment; DA2 and DA3 are the other data segments)

```
FIELD 3, n AS XX$, m AS YY$...
```

where "n", "m", etc. are the lengths of your PROFILE fields, and "XX\$", "YY\$" etc. are field names)

If you want an entire record, you can then access it by typing (between lines 1600 and 1700; make sure you have a third file):

```
GET 3, LR (where "LR" is the logical record)
```

To open: BASIC FIND/BAS-F:2 (two files are needed, one for the key and one for the index)

```
10 FIND/BAS
20 The small Computer Company, Inc.
30
40 Index demo program
50 Sequential access of PROFILE+ records from
   BASIC using a PROFILE+ index
60
70
100 CLS : CLEAR 5000 : DEFINT A-Z
200 INPUT " ENTER FILE NAME"; FS
300 FS = LEFT$(FS+"00000000",8) 'Pad with zeros to
   make 8-byte file name
400 INPUT " ENTER KEY LENGTH"; KY
500 OPEN "D", 1, FS + "/IX1", KY+2 'Open index file
600 FIELD 1, (KY) AS KY$, 2 AS LR$ 'Key field and
   two-byte pointer
```

```

700 OPEN "D", 2, F$ + "/KEY" 'Open /KEY segment
800 '
900 '
1000 FOR R= 4 TO LOF(1) 'Start of index is record #4
1100 GET 1, R 'Read the index
1200 LR = CVI(LR$) 'PROFILE+ record number from index
      pointer
1300 PR = INT((LR-1)/3) 'Deblock the KEY segment
      record:
1400 SR = LR - PR*3 - 1 'SR = sub-record number (0, 1
      or 2)
1500 PR = PR+1 'PR = physical record number
1600 GET 2, PR 'Get the /KEY record
1700 FIELD 2, (85*SR) AS DD$, 85 AS A$ 'A$ = /KEY
      segment data fields
1800 PRINT : PRINT LR; " "; A$ 'Print record number
      and data
1900 NEXT R 'Advance pointer to next index record
2000 CLOSE

```


```

47 REM NINE WHOLE NUMBERS IN THE SQUARE
48 FOR I=1 TO T
49 L$=MID$(Z$,I,1)
50 IF VAL(L$)=0 THEN S=S+1
52 NEXT I
53 IF T-S<9 THEN GOTO 120
55 IF K=T THEN K=1
      : GO TO 65
56 REM IF THERE ARE NINE WHOLE NUMBERS THEN THIS
57 REM PART CHECKS TO SEE IF THEY ARE THE NUMBERS
      1-9
60 IF K=T THEN K=1
      : J=J+1
65 IF J=10 THEN GOTO 120
70 K=K+1
80 L$=MID$(Z$,K,1)
90 IF VAL(L$)=J THEN R=R+1
      : J=J+1
      : K=1
      : GOTO 55
95 GOTO 60
110 IF THE NINE WHOLE NUMBERS ARE 1-9 THE NUMBER
      IS PRINTED
120 IF R=9 THEN PRINT X,Z$,
125 IF R=9 THEN LPRINT X,Z$
130 X=X+1
140 GOTO 30

```

LETTERS

A few of our readers have written in with a similar problem: They've added a segment or segments to an existing file, rewritten their screens to accept the new fields, then found that the records wouldn't accept data in those fields when they tried to update records.

The answer is to expand the file to accept the new segment. However, you don't have to add any records. When the prompt "This File Is Currently Allocated nnnn Records. How Many Additional Records . . ." comes up, simply press ENTER. 

microRESEARCH Problems

Richard V. Andree
microRESEARCH
P.O. Box 2910
Fort Worth, Texas 76113-2910

Before we go on to a further discussion of Problem 2, here is another program which solves Problem 1. It was written by M. Harris, a high school student.

```

5 CLS
10 DEFDBL I,X,Y
14 PRINT "THIS PROGRAM FINDS SQUARES THAT CONTAIN
      EACH OF THE
15 PRINT "NINE NON-ZERO DIGITS ONCE AND ONLY ONCE
16 LPRINT "THIS PROGRAM FINDS SQUARES THAT
      CONTAIN EACH OF THE
17 LPRINT "NINE NON-ZERO DIGITS ONCE AND ONLY
      ONCE
18 FOR J=1 TO 1000
      : NEXT
      : CLS
19 LPRINT
20 X=11810
25 PRINT " NUMBER"," SQUARE"," NUMBER"," SQUARE"
26 LPRINT " NUMBER"," SQUARE"
27 PRINT
      : LPRINT
29 REM BEGINNING OF MAIN PROGRAM
30 Y=X*X
34 J=1
35 R=0
36 K=1
37 S=0
40 Z$=STR$(Y)
45 T=LEN(Z$)
46 REM THIS PART CHECKS TO SEE IF THERE ARE JUST

```


Last month's Problem 2 suggested a hunt for palindromic squares having an even number of digits. Mathematically inclined readers would cut their computer time by first proving that if $S = N^2$ is a palindrome containing an even number of digits, then S , and hence N , must be divisible by 11. One proof uses a test for divisibility by eleven which forms the sum SE of all digits in even positions of S and the sum SO of all digits in odd positions of S . Then, if the absolute value of $SE - SO$ is zero or is divisible by 11, the original number S is divisible by eleven. In a palindromic number having an even number of digits this sum is zero or divisible by 11. Hence 11 divides S . Since $S = N^2$ and 11 is prime, 11 also divides N . This observation can save 90% of your computer time.

$(836)^2 = 698896$ has six digits
 $(798644)^2 = 637832238736$ has twelve digits
 $()^2 = 14$ digits

microRESEARCH Problem #3:

The number 371 has the property that the sum of the cubes of its digits equals 371: $3^3 + 7^3 + 1^3 = 27 + 343 + 1 = 371$. Find other whole numbers N such that the sum of the cubes of the digits of N is equal to N .

There are four such numbers among the 3-digit numbers. Mathematically inclined readers may wish to show that there cannot be such an N with five (5) or more digits and then find all integers N such that the sum of the cubes of the digits of N is equal to N .

Possible extensions: for "cubes" read "squares" or "fourthpowers" or "k-th powers" or "factorials". 



Four types of program files are defined for DD:

The Executive Program, which is resident in memory when TRSDOS is operating.

The Auxiliary System Files, which are the routines and commands which are needed to execute your commands and programs. These files load into an "overlay" area, and are present only when needed. These files will not affect memory above 51FFH.

The Library Command Files, which execute most of the operator commands. These files load into memory addresses from 5200H to 6FFFH. Because of the Library Files, machine language programs should generally be located above 6FFFH. Three library commands use ALL available RAM - BACKUP, FORMAT, and single drive COPY.

The Disk BASIC extensions to Level II BASIC. This file loads into memory beginning at 5200H.

In addition to low RAM, TRSDOS 2.7DD also uses some high memory locations. The keyboard driver routine resides above BD00H in a 32K system, and above FD00H in a 48K system. Also, DO uses memory above BADFH (32K) or FADFH (48K).

DOUBLE DENSITY EXTENSIONS TO TRSDOS

The following DD TRSDOS commands have been added beyond those commands which exist in SD Model I TRSDOS:


- BLINK — Turn blinking cursor ON or OFF.
- BUILD — Create an automatic command input file.
- CLEAR — Clear user memory between start and end addresses, as well as setting the memory protect address.
- CLS — Clear screen from TRSDOS.
- CONFIG — Temporarily change the drive stepping rate and specify the number of tracks for a drive (if they can be changed for that drive.)
- CREATE — Create a Pre-allocated disk file.
- DO — Begin Auto command input from a BUILD file.
- DUAL — Duplicate output to Video and Printer.
- ERASE — Erase a file from a diskette by writing zeros in the data area.
- ERROR — Display error message.
- FILFIX — Load and modify the contents of a disk file using a full screen display.
- FREE — Displays a disk allocation map.
- HELP — Explanations of TRSDOS commands.
- LPC — Line Printer Control program.
- MASTER — Set Master read/write drive.
- MEMTEST — Test ROM and RAM memory.
- PATCH — Change the contents of a disk file.
- PAUSE — Pause execution during execution of a BUILD file.
- PURGE — Delete disk files.
- RELO — Change where a machine language program loads into memory.
- SETCOM — Initializes the RS-232C communications channel (if the RS-232 board is installed.)
- SPOOL — Create an area in memory where output for the printer is dumped. Allows operation of the printer and some other operation.

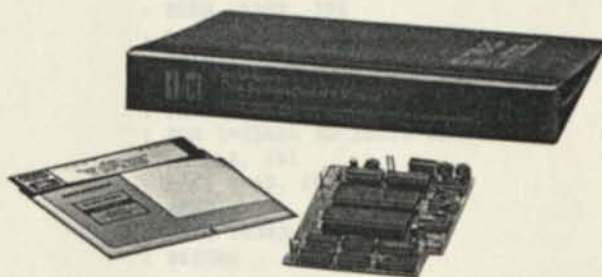
- TAPE — Transfer machine language programs between tape, disk and RAM.
- ULC — DD TRSDOS will automatically load the Upper/Lower case driver if the lower case modification is available.
- UNKILL — Recover a KILLED file, if the directory and disk space have not been written over since the KILL command. You can not UNKILL an ERASEd file.
- USER — Allows you to add a single machine language routine to the TRSDOS library.
- WP — Write protect a disk drive via software.

DOUBLE DENSITY ENHANCEMENTS TO DISK BASIC

The following enhancements have been made to Disk BASIC:

BASIC filename -F: -M: When loading Disk BASIC, you can now specify a program to be executed (filename), you can specify the number of files that may be OPEN at one time (-F:) and you can specify the highest address available for use by BASIC (-M:).

- CMD"A" — Return to TRSDOS with an error message.
- CMD"B" — Enable or Disable the <BREAK> key.
- CMD"C" — Compress the BASIC program in memory by removing spaces and remarks.
- CMD"D" — Display the directory for the specified drive.
- CMD"E" — Display previous TRSDOS error.
- CMD"J" — Convert calendar dates.
- CMD"K" — Turn CLOCK display on or off.
- CMD"L" — Load a Z-80 subroutine or program file into RAM.
- CMD"O" — Alphabetize (sort) a string array.
- CMD"P" — Check printer status.
- CMD"X" — Cross-reference of reserved words, string variables or strings in a program.
- CMD"Z" — Duplicate output to video and printer. 



A Baker's Dozen of Helpful Hints

John F. Rogers
600 Seventh Street
Morgan City, LA 70380

I own two 48K multi-disk Model I systems with printers. Since Spring 1978 I have been using my systems in various ways, read many magazines, talked to many other users, and learned quite a lot. (I'm just a beginner with assembly language, however.)

Although many tips have been published in *Micro-computer News* and in other magazines, perhaps some of the enclosed ideas have been missed along the way by enough of your readers that publishing them again would be helpful.

1. When using INKEY\$ in a loop to read the keyboard, be sure to compare the string variable to a length of zero—i.e., use A\$="" , not A\$=0 (where 0 represents a space.) Example:

```
100 A$=INKEY$
   : IF A$="" THEN 100
```

will loop at line 100 until a key is pressed.

2. The opposite of the above is true for using LPRINT to create a blank line — i.e., use LPRINT "0", not LPRINT"" to advance the printer one line. Example:

```
100 LPRINT"LINE ONE"
   : LPRINT" "
   : LPRINT"LINE THREE"
```

Editors note: This is especially true in early Radio Shack printers. The newer printers will respond to LPRINT by printing the contents of the buffer (if any) and advancing one line. Thus if the print buffer is empty, sending just an LPRINT will result in a blank line.

3. In 32-character mode, PRINT@x must use an even number for 'x' — i.e., PRINTCHR\$(23): PRINT@21,"DOUBLE WIDE" will not print anything that is visible on the screen. Example:

```
100 PRINTCHR$(23)
   : PRINT@74, "DOUBLE WIDE SCREEN PRINT"
```

4. When printing several items on the same line, it is not necessary to put the semicolon (;) between them — i.e., PRINT "PART NUMBER"; PN does not need the semicolon. Example: If X = 2, then

```
PRINT"SECTION ONE"X"SECTION THREE"
```

will print SECTION ONE 2 SECTION THREE on the same line. (The semicolon is still needed after the last element in a print statement if you want to suppress the cursor—dropdown.)

5. PRINT@p,STRING\$(n,c) is much faster than POKE for horizontal graphics. Example:

```
PRINT@64,STRING$(64,131)
```

will print a horizontal line in the blink of an eye, while

```
FOR I=15424 TO 15487
  : POKE I,131
  : NEXT I
```

allows one to see the line being formed, even with DEFINT I to speed up the loop.

6. The Reference Manual gives the impression the PRINT@p, allows only integer values for 'p', but variable may also be used for location 'p'. Example:

```
FOR J=192 TO 560 STEP 16
  : PRINT@J, "LOCATION" J
  : NEXT J
```

will print LOCATION 192, etc., four across on six lines starting with the fourth line.

7. LPRINTTAB(x) can go past TAB(63) on 132 column printers by using STRING\$(n,c). Example:

```
LPRINTTAB(60)"62"STRING$(48,32)"TAB 110"
```

will print TAB 110 beginning at x = 110 (the 111th print position on the line.)

8. To print a heading that shows the LPRINTTAB() positions on a 132-column printer, use:

```
LPRINT"0 5";
  : FOR T=10 TO 95 STEP 5
  : LPRINT" " RIGHT$(STR$(T),2);
  : NEXT T
  : FOR T=100 TO 130 STEP 5
  : LPRINT" " RIGHT$(STR$(T),3);
  : NEXT T
  : LPRINT"1"
```

9. To get quotation marks inside a string, use CHR\$(34). Example:

```
PRINT"HERE ARE "CHR$(34)"QUOTES"CHR$(34)" WITHIN
QUOTES."
```

will print HERE ARE "QUOTES" WITHIN QUOTES.

```
A$="HERE ARE "+CHR$(34)+"QUOTES"+CHR$(34)+
" WITHIN QUOTES."
: PRINT A$
```

will do the same. (The quotation mark at the end of a string not needed if the quotation mark would be the last character on the line.)

10. This subroutine will print a solid border of graphic characters around the edge of the screen:

```
10 CLEAR 124
  : CLS
  : POKE 15360, 191
  : PRINT@1, STRING$(62,131);
  : POKE 15423, 191
  : POKE 16320, 191
  : PRINT@961, STRING$(62,176);
  : FOR I=15424 TO 16256 STEP 64
  : POKE I, 191
  : POKE I+63, 191
  : NEXT I
  : POKE 16383, 191
  : RETURN
```

When the routine is finished, the cursor will be at screen position 1023.

11. This subroutine creates a blinking prompt on the bottom line of the screen; the line is erased when a key is pressed. (The cursor will be on the last line; you should do CLS or another PRINT@ whenever the program resumes.)

```
1 Z0$=INKEY$
  : PRINT@978,"PRESS ANY KEY TO CONTINUE";
  : FOR I=1 TO 40
  : NEXT I
  : IF Z0$<>"" THEN PRINT@976,STRING$(31,32);
  : RETURN
2 PRINT@976,STRING$(31,143);
  : FOR I=1 TO 2
  : NEXT I
  : GOTO 1
```

12. This subroutine moves a rocket from bottom-right to top-right of the screen:

```
100 DIM RK(12)
: RK(1)=16120
: RK(2)=16121
: RK(3)=16183
: RK(4)=16185
: RK(5)=16247
: RK(6)=16250
: RK(7)=16311
: RK(8)=16312
: RK(9)=16313
: RK(10)=16314
: RK(11)=16373
: RK(12)=16377
110 FOR J=1 TO 12
: POKE RK(1), 150
: POKE RK(2), 148
: POKE RK(3), 170
: POKE RK(4), 170
: POKE RK(5), 149
: POKE RK(6), 149
: POKE RK(7), 181
: POKE RK(8), 176
: POKE RK(9), 176
: POKE RK(10), 149
: POKE RK(11), 191
: POKE RK(12), 149
: FOR I=1 TO 12
: RK(I)=RK(I)-64
: NEXT I
120 NEXT J
: PRINT@55, "B O O M !";
: RETURN
```

13. Hey, reader, why not make your "trick of the trade" number thirteen, and send it to [TRS-80 Microcomputer News?](#)



Model I/III Bugs, Errors and Fixes

Note to Users:

The following changes and corrections are provided for your information. If you have an applications program which is working correctly, you should probably NOT make any changes to it. If you feel that changes should be made, but you do not feel qualified to make the change yourself, please contact your local Radio Shack Computer Center or Expanded Computer Department for assistance. If you do not have access to one of these stores, then you may want to call Computer Customer Service in Fort Worth for assistance.

DISK PAYROLL (26-1556 Version 3.0 and Prior)

Users of Disk Payroll, version 3.0 or prior, in Oklahoma or Montana should contact their local Computer Center, Expanded Computer Department or Dealer for information on changes needed to Disk Payroll:

1. Oklahoma—Changes are needed in the State Tax Table
2. Montana—Changes are needed to allow multiple "Standard Deductions as Percentage of Gross Pay"

MODEL III DISK SYSTEM MANUAL (26-2111)

On pages 78-79 of the Model III Disk System Manual, please note the following additional information about the \$INIT I/O Call:

If the filespec name is found, \$INIT simply opens the file for use and resets the pointers. \$INIT is a destructive open of a file. The following pointers are reset: NRN, ERN, ODECR, EOFOLDLPR. If no writes are made, the disk copy is not changed. If a write is made, the directory is updated with the new information.

On page 80 please change the first paragraph under \$READ to read:

If LRL is not equal to zero, then \$READ transfers the logical record whose number was placed in the DCB by \$POSN into the RAM area addressed as UREC. The value of LRL is defined at open time. The record comes from "BUFFER" defined at open time. If TRSDOS must read a new physical record to satisfy the request, it will do so. "Spanned" logical records will be re-assembled as necessary. \$READ will automatically increment in the DCB the offset to delimiter at end of current record (ODECR) by the value of LRL for each logical record and NRN by one for each physical record after each transfer is completed. \$INIT/\$OPEN will set NRN = X'0000' and ODECR = X'00' in order to read the first record with the first \$READ.

Also on page 80, please change the first paragraph under \$WRITE to read:

If LRL is not equal to zero, then \$WRITE transfers the one logical record from the RAM area addressed by UREC with the length LRL as defined at open time. The record goes into the 'BUFFER' which was defined at open time. If TRSDOS must write a physical record in order to satisfy the request, it will do so. "Spanning" will be handled by TRSDOS as necessary. At \$INIT/\$OPEN time the DCB value of NRN = X'0000' and the offset to delimiter at end of current record (ODECR) = X'00' so the first record can be written. After each logical record is transferred, the ODECR value in the DCB will be incremented by the value of LRL. After each physical record is transferred, the NRN value in the DCB will be incremented by one.

COBOL DEVELOPMENT SYSTEM (26-2203)

Page 23 in the user's guide (in the third paragraph of section 2.5.1) the manual states that a record can be any length from 2 to 255. Change 255 to 254, which is the maximum record length for variable records. If you try to create a file with a record length of 255, an ERROR 94 occurs.

Microcomputer-Based Data Management for Small Libraries

The Department of Library Science at Clarion State College in Clarion, Pennsylvania, is concerned about the data management needs of small libraries. Professor Bernard Vavrck at Clarion State has been experimenting with ways in which the TRS-80 Model III and off-the-shelf software can be used in school, rural, and small college libraries. He has developed some interesting applications for Radio Shack's MICROFILES program and has used the TRS-80 Model III in giving workshops and demonstrations at library conferences.

Microfiles is a software package for use with a TRS-80 Model I or Model III disk system with at least 32K RAM. It allows you to design a custom filing system based on your individual needs. You can build files in any format, make particular items in a file optional or required, index information and locate particular entries, update items in a file easily, and print out information in any format.


Using Microfiles with a 48K Model III disk system and a Daisy Wheel printer, Vavrck has worked out applications both for library circulation (check-outs and returns) and for inventory. A typical file for circulation might have a listing for each book held by the library, using the title, book number, and/or other information. "Name" and "date" fields would be "optional" items in the file, meaning that they need not be filled in all of the time for each book. When a book is checked out, the borrower's name and the date due can then be entered as part of the book's listing. Or, each listing could be entered as a book is checked out. When the book is returned, the entire listing can be deleted until the book is borrowed again. To find listings of overdue books, the librarian can index the file by date and/or use the Microfiles "FIND" command to locate listings for all books that were due on a certain date. Vavrck estimates that a data management system like this would work well for a small library with up to 1000 circulations per week.

A Microfiles-based filing system for recording the library's acquisitions (books on order or just received) would work similarly. Instead of searching through paperwork to find out about books ordered by the library, the librarian could use Microfiles to set up a quick, easily updated reference source. A file for acquisitions would have one listing per book. Information fields in the file would include the book's author, title, publisher, Library of Congress number or ISBN number, the date ordered, the supplier the book was ordered from, and the price. This would give the librarian a valuable record system that would allow him or her to locate information on ordered books quickly—by publisher, date ordered, or any of the other fields.

Similar filing systems could be used for small inventory needs, such as the library's list of community resources (other libraries and collections, charitable and educational institu-

tions, and so on). For a small library, Microfiles could keep inventory records by author, title, and subject—similar to the card catalog. Vavrck noted that the time required to key catalog information would be a disadvantage, but added that "once that is done, a tremendous facility is available." Other library departments that could use such a filing system include the audio/visual and serials/periodicals departments.

Vavrck mentioned the flexibility of Microfiles and its low cost as special advantages. He found the Microfiles user manual very clear, and he remarked that "help is nearby" that Radio Shack store personnel in Erie, Pa. had been helpful in answering his specific questions. He remarked that in the field of library science there has been a general "dearth of attention paid to small libraries," but added that he has lately seen microcomputer programs emerging that can help small libraries—some of these programs specifically designed for library use and written to run on the TRS-80 microcomputer. Some of these programs, Vavrck notes, can even calculate numerical totals and library discounts. The microcomputer may well become an important tool for small libraries.

Note: Microfiles (Catalog Number 26-1565) is available from Radio Shack stores and dealers. The listed price is \$99.95, though prices may vary at individual stores and dealers. Minimum system is a 32K Model III or Model I with at least one disk drive. 

Introducing "Number Theory"

Radio Shack's new "Number Theory" program is an exciting addition to the growing library of TRS-80 courseware designed to aid students in learning math. With this program, educators can now use either the TRS-80 Model I or Model III to present one of the oldest and most elegant branches of mathematics to students at the junior high school level and up.

The developers of the "Number Theory" program, Liang and E. B. Saff, have spent many years working with high school students gifted in math. Their program has been classroom-tested at the University of South Florida—Hillsborough County and Pinellas County summer programs in mathematics and science.

Besides supplementing secondary and college courses, Radio Shack's "Number Theory" can also be used outside the classroom—as a self-study course by anyone interested in learning about the modern number system. The lesson

concentrate almost exclusively on the computational or constructive areas of number theory, making the program accessible to users with a minimal mathematics background. In fact, Part I of the program requires only junior high school-level math skills; knowledge of high school algebra is sufficient for the rest of the program.

The field of number theory presents many purely mathematical challenges and is filled with unresolved or "open" problems of current interest to mathematicians. However, number theory also has many practical applications, such as coding theory and cryptography, and contains a wealth of examples and ideas which are useful in studying other branches of mathematics. Radio Shack's "Number Theory" program covers the fundamental topics in number theory. The lessons are designed to provide a clear overview of the essentials of number theory, especially the basic properties of integers.

The program is divided into six major parts. Each part begins with a lesson menu listing the names of each lesson. The topics covered in these lessons are as follows:

Part I: Prime Factorization and Greatest Common Divisors

- Lesson 1: Divisibility of Integers
- Lesson 2: Perfect Numbers
- Lesson 3: Prime Numbers
- Lesson 4: Prime Factorization
- Lesson 5: Greatest Common Divisor

Part II: Solutions of Linear Diophantine Equations

- Lesson 1: Division Algorithm
- Lesson 2: Euclidean Algorithm
- Lesson 3: Diophantine Equations

Part III: Congruence MOD N

- Lesson 1: Congruence MOD N
- Lesson 2: Complete Residue System MOD N
- Lesson 3: Reducing an Integer MOD N

Part IV: Arithmetic MOD N

- Lesson 1: Addition, Multiplication, and Inverse MOD N
- Lesson 2: Evaluating a Power MOD N

Part V: Linear Congruence & Primitive Roots

- Lesson 1: Linear Congruence
- Lesson 2: Chinese Remainder Theorem
- Lesson 3: Primitive Root MOD a Prime

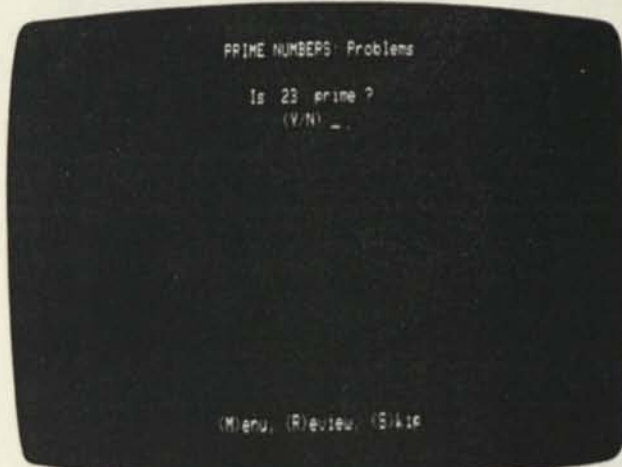
Part VI: Quadratic Residue & the Euler Phi-Function

- Lesson 1: Quadratic Residue
- Lesson 2: Continued Fraction Expansion
- Lesson 3: Euler Phi-Function

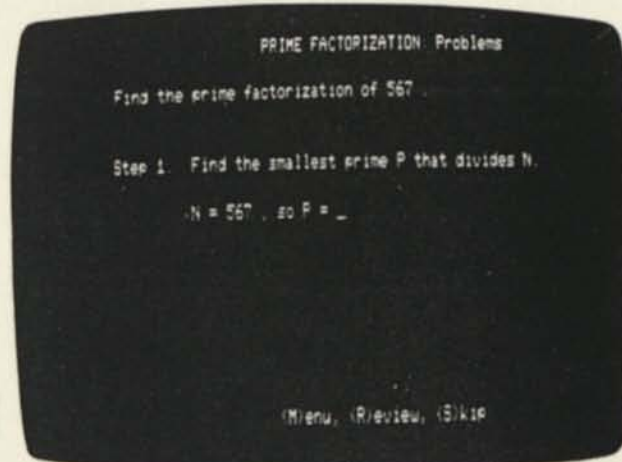
The "Number Theory" program has many features to aid and encourage the student. The lessons are progressive—arranged to build on the concepts and skills covered in the previous lessons; students are encouraged to approach the subject systematically, working from one lesson to the next. Each lesson provides definitions, explanations, examples, and problems over a specific area of number theory, to insure that the student fully understands each topic before moving on to the next lesson.

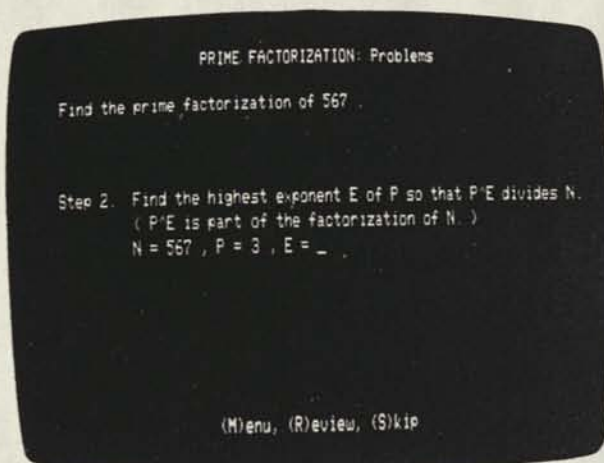
In the problem sections of each lesson, the program keeps track of the number of problems the student has successfully completed. The program provides reinforcement when the student's score is high, and advice for review when the score is low. Feedback messages appear after individual problems. The student sees a positive message for a correct response. For an incorrect response, the student is given another chance or the correct answer is displayed. The

problems and many of the examples are randomly generated, so the program can be used again and again without repetition in the interactive sections. The type of problem used varies throughout the program, according to the topics covered. Some lessons use simple YES/NO questions to check student understanding of a basic concept. The problem screen below is from Part I, Lesson 3 on Prime Numbers:



Other lessons use long problems with several steps to check student ability to successfully complete a procedure. For instance, the following screen is from the problem section of Part I, Lesson 4, "Prime Factorization":





Another important aspect of this program is the availability of special keys. Throughout the program, the student is allowed to control the pace of the lessons, using four options: (M)enu, (R)evue, (S)kip and (C)ontinue. Pressing the "M" key takes the student back to the previous menu; that is, from a screen in the first lesson of Part I, "Divisibility of Integers," the "M" key gives the student the menu for Part I:



In those lessons divided into sections, the "M" key from within a section takes the student back to the menu for that lesson.

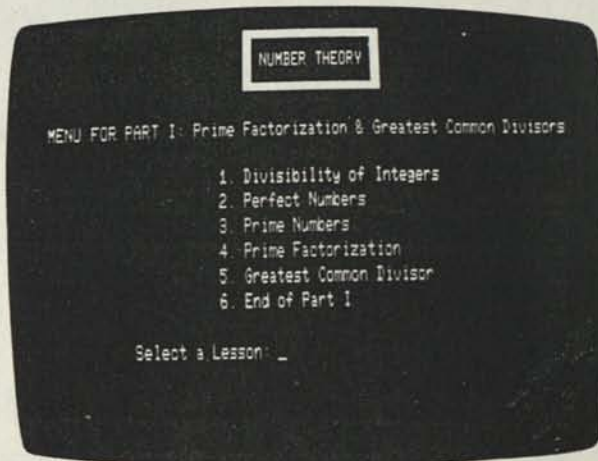
The "R" key returns the student to the first screen of a lesson; this "(R)eturn" function makes it possible for the student who wishes to review to return to the beginning of the lesson without going back through the menu. The student will thus feel free to progress through the lessons at his or her own pace, returning to the definitions and examples as often as is needed.

For those students who may not need to see all the explanations and examples for each lesson, the "S" key provides a way to skip from the initial definition screen of each lesson to the set of problems at the end of the lesson. Students can use this "(S)kip" option to scan through lessons they have already worked once, or to return to a set of problems after a break.

The "C" key allows the student to continue through the lessons one screen at a time, taking time to read each explanation and example as it appears. (M)enu, (R)evue, (S)kip,

and (C)ontinue appear as prompts at the bottom of the screen whenever the options are available to the student, making the program self-explanatory and easy to get around in.

The manual included in the "Number Theory" package provides step-by-step instructions on how to use the program, along with a reference section with summaries of each lesson, sample problems, and instructional objectives. The manual also provides suggestions for additional projects in number theory which are designed to stimulate the advanced student to write his or her own algorithms and computer programs involving the techniques covered in the "Number Theory" program.



The "Number Theory" program is designed to provide a variety of applications. For instance, the program could be used by an advanced student for self-instruction. Or an entire algebra class could be structured around the material in the program, so that the teacher could introduce each concept to the class, present the initial examples and answer student questions, and then allow the "Number Theory" program to provide students with the necessary review and skill-building practice. "Number Theory" can also be used with Radio Shack Network System.

The "Number Theory" program (catalog no. 26-2613) is available—on 16K tape and 32K disk—at Radio Shack Stores and Computer Centers. \$69.95 is the listed price. Prices may vary at individual stores and dealers.



PATCHWRT

Richard A. Burke CSR
Radio Shack Computer Center 11-7554
Carmichael, CA 95608

I would like to take this opportunity to introduce to you a program which I have written. The program's name is PATCHWRT (short for PATCHWRITER). As the name implies, the program is used to write, edit, save, and retrieve various Model II DO files.

The program ends the nasty frustrations of incorrect PATCHes by allowing the user to correct any portion of any line. With BUILD files, the user must replace the whole line even if only one character is incorrect. With PATCHWRT, he can not only correct lines, he may add lines, delete lines, and insert new lines into the file. The program is written in standard BASIC and from the TRSDOS READY level can be entered by typing the following:

```
BASIC PATCHWRT -F:1
```

Once BASIC has loaded the program the screen will ask what function you wish to perform. You may choose from the following options

(A)dd a Patch (Create a new BUILD file)

(E)dit an existing file

or you may

(Q)uit the program

When you choose (A) for add, the screen begins to input one set of commands at a time. To end each line, simply hit <ENTER> and begin inputting the next line. PATCHWRT will keep track of up to 20 command lines and will list which line the user is currently on

For example, enter the (A)dd section and type the following PATCH:

```
PATCH SYSTEM/SYR A=DEFF F=AB C=00 <ENTER>
```

When you have finished entering the line, simply hit the <ENTER> key at the beginning of the next line. The program will then immediately go into the edit phase.

The editor is quite simple to use. It allows you to select from the following options:

(Y) - make changes to a particular line

(N) - exit editing as no more changes are necessary

(I) - insert a line at a particular place (even at the end of the patch)

(D) - delete any line in the file

Each of the options will then ask you which line number you wish to work on. Give the line number and then hit <ENTER>. If you were correcting part of a given line (Y), the screen will display that line only and ask for the "SEGMENT TO REPLACE." To insure proper editing, you should give at least a three character "section." For example:

The screen shows:

```
PATCH SYSTEM/SYR A=DEFF F=AB C=00
SEGMENT TO REPLACE:
```

The user should respond - SYR <ENTER>

The screen then asks:

REPLACE WITH:

and the user should answer with the phrase - SYS <ENTER>

The machine makes the corrections and takes the user back to edit the menu and re-displays all the lines. For our example we would see:

```
1 - PATCH SYSTEM/SYS A=DEFF F=AB C=00
```

If you need to insert a new line, type (I) and then answer the line number prompt. If we need a blank line ahead of our current line we would type (I) and respond <1> <ENTER>. The program then asks the user for what is to appear on the newly created line. For example we type:

```
PAUSE THIS PATCH DOES XXXXX XXXXX TO YYYYY FOR ZZZZ
```

and hit <ENTER>

The display then reads:

```
1 - PAUSE THIS PATCH DOES XXXXX XXXXX TO YYYYY FOR
ZZZZ
```

```
2 - PATCH SYSTEM/SYS A=DEFF F=AB C=0
```

In order to delete a whole line, we simply type (D) and enter which line is to be deleted. Suppose we typed (D) and then <1> <ENTER>. The computer would then ask if we are sure we wish to delete this line. If we answer (N) for no we return to the edit menu. If we type (Y) for yes, the screen then shows

```
1 - PATCH SYSTEM/SYS A=DEFF F=AB C=0
```

and the line has been deleted

When there are no more corrections to be made, we type (N) at the edit menu and are taken to the output section of the program. The computer asks if we are sure the program is to be saved (answer Y or N as appropriate). If we do not save the file it is removed from memory and we are returned to the (A)dd (E)dit (Q)uit menu

If we are going to save the program we are then asked for a file name. This file name should observe all Model II filename protocols. After hitting <ENTER>, the BUILD file is then saved under the given name and the user is returned to the top menu. If you wish to execute the file simply type (Q) to quit and then use the BUILD file as you would use any DO file (i.e. type DO filename)

If you have a file on the disk already, you can (E)dit it by typing (E) and entering the file's name. The program then goes immediately into edit phase and the user may proceed as listed above.

```
1 'Program to write PATCHes and DO files
```

```
2 'Written by Richard Burke -C.S.R.
```

```

3 '      Radio Shack Computer Center 11-7554
4 '      6305 Fair Oaks Blvd.
5 '      Carmichael, Ca. 95608
6 '      (916)-484-6815
7 'Completed May 27,1982
8 '
9 'Clear memory for 20 strings
10 CLEAR 3000
   : DIM A$(20)
20 CLS
   : PRINT@(9,27),"PATCH WRITER - VERSION 3.3"
25 PRINT@(12,32),"RICHARD A. BURKE"
27 'Find out what user wishes to do
30 PRINT@(15,23),"(E)DIT PATCH (A)DD PATCH OR
   (Q)UIT";
40 Z$=INKEY$
   : IF Z$="" THEN 40
50 IF Z$="E" THEN 560 ELSE IF Z$="A" THEN 55 ELSE IF
   Z$="Q" THEN SYSTEM ELSE 30
55 FOR I=1 TO 20
   : A$(I)=" "
   : NEXT
57 'Create new PATCH or DO file
58 'Begin inputting strings
60 CLS
   : B=1
65 PRINT "ENTER STRING -";B;
70 INPUT A$(B)
73 'If A$(C)="" then no more strings to be added -
   goto edit section
75 IF A$(B)="" THEN A=B-1
   : GOTO 100
80 B=B+1
   : GOTO 65
100 CLS
104 'Edit file section
105 'List out the file
110 FOR B=1 TO A
120 PRINT B;"- ";A$(B)
130 NEXT
135 'Find out if there are any changes
136 'Y = change a given line
137 'N = no more changes - goto output section
138 'I = insert a line
139 'D = delete a line
140 PRINT@(22,10),"ANY CHANGES (Y/N/I/D)";
145 Z$=INKEY$
   : IF Z$="" THEN 145
150 IF Z$="Y" THEN 155 ELSE IF Z$="N" THEN 330 ELSE IF
   Z$="I" THEN 205 ELSE IF Z$="D" THEN 230 ELSE 140
153 'Determine which line to be edited and make the
   correction
154 'If not valid line number then try again
155 PRINT@(22,10),"EDIT WHICH LINE NUMBER ";
   : INPUT C
   : IF C<1 OR C>A THEN 155
157 'Print line to be corrected
160 CLS
   : PRINT@(5,0),A$(C)
163 'Get segment to be replaced
165 PRINT@(9,0),"SEGMENT TO REPLACE"
   : INPUT " " ;D$
   : D=LEN(D$)
167 'Get correction to line
170 PRINT"REPLACE WITH"
   : INPUT " " ;E$
173 'Search to see if segment to be replaced is in
   the string
174 'If not then exit else make replacement
175 FOR B=1 TO (LEN(A$(C))-D+1)
180 IF MIDS(A$(C),B,D)=D$ THEN 190 ELSE NEXT
185 PRINT "STRING NOT FOUND"
   : FOR X=1 TO 1000
   : NEXT
   : GOTO 100
190 E=B-1+D
195 A$(C)=LEFT$(A$(C), (B-1)) + E$ + RIGHT$(A$(C),
   (LEN(A$(C))-E))
200 GOTO 100
203 'Insert line section
204 'Find out where to put new line
205 PRINT@(22,10),"INSERT A STRING WHERE";
   : INPUT C
207 'Get the new line
210 INPUT"ENTER NEW STRING";IN$
213 'Insert new line into the file
214 'Then exit
215 FOR B=A TO C STEP(-1)
220 A$(B+1)=A$(B)
   : NEXT
   : A=A+1
221 A$(C)=IN$
   : GOTO 100
225 'Delete line section
226 'Find out which line is to be deleted
227 'If user is sure, then delete line else exit
230 PRINT@(22,10),"DELETE WHICH LINE #";
235 INPUT C
236 PRINT@(23,10),"ARE YOU SURE YOU WISH TO DELETE
   LINE #";C;"(Y OR N)";
237 Z$=INKEY$
   : IF Z$="" THEN 237
238 IF Z$="Y" THEN 240 ELSE IF Z$="N" THEN 100 ELSE
   236
240 FOR B=C TO A
245 A$(B)=A$(B+1)
   : NEXT
250 A=A-1
260 GOTO 100
330 'Output section
335 'Add all necessary info to the front & back of
   file
340 A$(1)=CHR$(8) + A$(1)
350 FOR F=1 TO A
360 A$(F)=A$(F) + CHR$(13)
370 NEXT
380 CLS
381 'If user wishes to save then continue else exit
   to beginning
383 PRINT@(22,10),"DO YOU WISH TO SAVE";
384 Z$=INKEY$
   : IF Z$="" THEN 384
385 IF Z$="Y" THEN 386 ELSE IF Z$="N" THEN 30 ELSE
   383
386 PRINT@(22,9), " ";
387 'Get filename for saving
388 'If filename exists then kill the file
389 'Then save file to disk
390 INPUT" SAVE AS WHAT FILENAME";F$
400 ON ERROR GOTO 730
410 KILL F$
420 OPEN "R",#1,F$,1
430 FIELD #1, 1 AS Z$
440 FOR F=1 TO A
450 FOR G=1 TO LEN(A$(F))
460 LSET Z$ = MIDS(A$(F),G,1)
470 PUT #1,(LOF(1)+1)
480 NEXT
490 NEXT
500 CLOSE
510 CLS
520 GOTO 30
550 END
555 'Get file for editing
556 'If file doesn't exist then exit
560 CLS
   : INPUT "EDIT WHICH FILENAME";F$
565 FOR X=1 TO 20
   : A$(X)=" "
   : NEXT
570 OPEN "R",#1,F$,1
580 FIELD #1.1 AS Z$

```

```

590 E=1
      : C=1
595 ON ERROR GOTO 730
596 'Begin input of valid file
600 GET #1,E
610 A$(C)=A$(C)+ZZ$
615 'Increment string count if end is detected
620 IF ZZ$=CHR$(13) THEN C=C+1
625 'Exit this section if end of file
630 IF E=LOF(1) THEN A=C-1
      : GOTO 650
640 E=E+1
      : GOTO 600
650 CLOSE
655 'Begin stripping away file markers from the
      strings
656 '(Output replaces them)
660 A$(1)=MID$(A$(1),2,LEN(A$(1))-2)
670 IF A=1 THEN 100
680 FOR X=2 TO A
690 Y=LEN(A$(X))-1
700 A$(X)=LEFT$(A$(X),Y)
710 NEXT
720 GOTO 100
725 'Handle all expected errors
730 IF ERL = 410 THEN RESUME 420
740 IF ERL = 600 THEN PRINT "NO SUCH FILE"
      : FOR M=1 TO 1000
      : NEXT
      : CLOSE
      : KILL F$
      : RESUME 30

```

To prevent this from happening, apply the following patches. The date at the bottom of the STARTUP screen must be 02/23/1982 before you apply these patches; if it is not, obtain and apply all previous patches.

```

PATCH SCRIPSIT/SYS R=59 B=142 F=20BBF1FD361400
C=FD36140020B7F1
PATCH SCRIPSIT/SYS R=61 B=241 F=01 C=02

```

Sometimes user defined print codes are split by the "wraparound" feature of the system. This causes confusion in recognizing them as print codes and/or translating them properly. The following patches correct the problem:

```

PATCH SCRIPSIT/SYS R=157 B=220 F=D64147 C=CD58D6
PATCH SCRIPSIT A=9F67 F=E1FE07 C=C360D6
PATCH SCRIPSIT A=D658 F=00000000000000000000
C=E67FD64147C9E17EFE07
PATCH SCRIPSIT A=D664 F=000000 C=C36A9F

```

The Merge routine was failing to set up the proper data for alignment tabs. This sometimes caused the aligned text to print in the wrong column when merging. The following patches will correct the problem:

```

PATCH SCRIPSIT/SYS R=139 B=175 F=32FDDA C=CD4ED6
PATCH SCRIPSIT A=D64E F=00000000000000000000
C=32FDDAFD7E163223FFC9

```

When using the SCRIPSIT backup utility, there is no prompt when the destination diskette contains data. Also, when doing a backup to the same drive other than drive 0, the source diskette would be formatted. The following patches will correct these problems:

```

PATCH SCRIPSIT/SYS R=37 B=147 F=32EDE2 C=C36BD6
PATCH SCRIPSIT A=D66B F=00000000000000000000
C=32EDE26F3A10E38D200D
PATCH SCRIPSIT A=D675 F=00000000000000000000
C=FE30280921202022F0E2
PATCH SCRIPSIT A=D67F F=000000000000 C=22F2E2C340E4

```

After applying these patches, apply the following to change the date in the STARTUP screen:

```

PATCH STARTUP A=E40E F=B0B2AFB2B3 C=B0B5AFB2B0

```

Correction 2 - When repaginating a document with multiple headers or footers, repaginate sometimes uncenters centered-lines and reformats paragraphs even when no reformat was requested and when the paragraphs were locked.

Before applying the following patches to correct this problem, check to make certain that the date at the bottom of the STARTUP screen is 05/20/1982. If it is not, obtain and apply all previous patches.

```

PATCH SCRIPSIT A=D635 F=00000000000000000000
C=E52A4CF47EE1FE8FC823
PATCH SCRIPSIT A=D63F F=0000000000 C=232323C979
PATCH SCRIPSIT A=D644 F=00000000000000000000
C=FE03D8CD79E4C4CFE4C9
PATCH SCRIPSIT/SYS R=59 B=209 F=23232323 C=CD35D600
PATCH SCRIPSIT/SYS R=59 B=240 F=CD79E4C4CFE4
C=CC43D6000000
PATCH SCRIPSIT/SYS R=60 B=51 F=79FE0338 C=CC43D618

```

After making the above patches, apply the following to change the date in the STARTUP screen:

```

PATCH STARTUP A=E40E F=B0B5AFB2B0 C=B0B5AFB2B4

```

Correction 3 - When using a user defined print code to define a character or characters such as [c] or [f], the character(s) would not adhere to the current print mode (i.e. bold, underline, double underline, etc.).

Model II Bugs, Errors, and Fixes

NOTE TO USERS:

The following program changes and corrections are provided for your information. If you have an applications program which is working correctly, you should probably NOT make any changes to it. If you feel that the changes should be made, but you do not feel qualified to make the changes yourself, please contact your local Radio Shack Computer Center or Expanded Computer Department for assistance. If you do not have access to one of these stores, then you may want to call Computer Customer Services in Fort Worth for assistance.

PATCHES

PATCHes are entered from TRSDOS READY and are used to make corrections to files stored on the disk.

1. Before making a PATCH, back up the diskette that requires modification and make the PATCHES to the backup copy of the diskette.
2. Apply PATCHES according to the information given in your TRSDOS manual.

SCRIPSIT 2.0 (26-4531)

Following are several series of patches to correct problems that have been found in Scripsit 2.0. They should be applied in the order that they appear.

Correction 1 - When repaginating a document and reformatting it, locked blocks are sometimes changed. The margins will not be adjusted, but tabs will be refigured, leading to strange results.

The following patches will correct the problem:
Be sure that the date at the bottom of the STARTUP screen is 05/24/82. If not, be sure to obtain and do all previous patches.

```
PATCH SCRIPSIT/SYS R=163 B=196 F=F7F7 C=19D6
PATCH SCRIPSIT A=D619 F=000000000000000000000000
C=78FE213004CDF7F7C9C53AFDDA
PATCH SCRIPSIT A=D626 F=000000000000000000000000
C=0E01CB5F28043AFDD4FCD35F9
PATCH SCRIPSIT A=D633 F=0000 C=C1C9
```

After applying these patches, apply the following to change the date in the STARTUP screen:

```
PATCH STARTUP A=E40E F=B0B5AFB2B4 C=B0B5AFB2B7
```

Correction 4 - Scripsit will hang up if you try to edit a word that will not fit in the margins. The following patch corrects this problem. First be sure that the date at the bottom of the STARTUP screen is 05/27/1982. If not, obtain and make all previous patches.

```
PATCH SCRIPSIT A=9F68 F=60 C=5E
```

If a user defined print code contains a space, the space will not be underlined via the normal SCRIPSIT underline mode. The following patch corrects this problem:

```
PATCH SCRIPSIT A=D61B F=21 C=20
```

In order to print more than one copy of a document when doing a merge printout, apply the following patches:

```
PATCH SCRIPSIT A=D5E5 F=000000000000000000000000
C=C2EBD53204DB3A50DC
PATCH SCRIPSIT A=D5EE F=000000000000000000000000
C=3290E7C3BAE0
PATCH SCRIPSIT A=D5F5 F=000000000000000000000000
C=CFC25CE13A9AE5FE43
PATCH SCRIPSIT A=D5FE F=000000000000000000000000
C=CA48E12190E735
PATCH SCRIPSIT A=D605 F=000000000000000000000000
C=C2BAE0C348E1
PATCH SCRIPSIT/SYS R=135 B=186 F=200332 C=C3E5D5
PATCH SCRIPSIT/SYS R=136 B=78 F=CF2014 C=C3F5D5
```

After applying these patches, apply the following to change the date in the STARTUP screen to 06/02/1982

```
PATCH STARTUP A=E40E F=B0B5AFB2B7 C=B0B6AFB0B2
```

Correction 5 - Scripsit locks up when the user attempts to use a serial printer with Merge. The following patches correct this problem:

The date at the bottom of the STARTUP screen must be 06/02/1982 before these patches are applied. If not, obtain and apply all previous patches.

```
PATCH SCRIPSIT/SYS R=176 B=20 F=90DA C=0BD6
PATCH SCRIPSIT A=D60B F=000000000000000000000000
C=FDCB7F7EC8C390DA
```

Under some circumstances, when the user recalls text into a word, causing the text to exceed the margins, the cursor may be left off the text. Scripsit fails to flag the error and unpredictable results may occur. The following patches will correct the problem:

```
PATCH SCRIPSIT A=B228 F=2CA4 C=DFD5
PATCH SCRIPSIT A=D5DF F=000000000000000000000000
C=CD2CA4C35CA9
PATCH SCRIPSIT A=B21C F=FE8D20 C=C3B8D5
PATCH SCRIPSIT A=D5B8 F=000000000000000000000000
C=FE8DCA20B2CD5CA9AFC9
```

After the preceding patches have been applied, make the following patch to change the date at the bottom of the STARTUP screen to 06/09/1982.

```
PATCH STARTUP A=E412 F=B2 C=B9
```

The Firmament

Waldo T. Boyd
P.O. Box 86
Geyerville, GA 95441

The Model II is becoming the workhorse of small business proprietors, professionals, and specialized departments within larger companies. As such it is rarely tested anything like its ability to perform in what might be called non-business ways. Despite its long workday, there are times when the operator's attention is required elsewhere for short or medium "break" periods, and very few of these happen when people turn off the CPU and its peripherals for such relatively brief interludes in their busy day.

Chances are, the life of the cathode-ray tube filament is prolonged by this practice since initial current surge at power on is avoided. On the other hand, the screen endures repetitive burning of a fixed pattern into the phosphor, a situation passersby sometimes interpret as an open invitation to start touching the various keycaps.

Here's a simple, neat little program that uses only one 256-byte record, which can be loaded into working memory in a few seconds just before taking that coffee break. While running, it looks like space at night, with the stars twinkling the distance. Somehow a dynamic screen looks much more business like and less inviting to itchy fingers, making the program more practical than it might at first appear. Certainly it is anything but frivolous.

Four "star" types are depicted with the plus sign, the caret, the period, and the asterisk. A "wink-out" blank follows. PRINT@ statements assign print positions based upon random selection of row and column numbers. The cursor is inhibited in line 10, and the program loops back upon itself at line 20. The field never becomes full because again at random, the line feed is activated and the field moves up a notch, and stars disappear. The constant exchange of asterisk for period, cross for caret—48 combinations happening sequentially so rapidly as to appear almost simultaneous—gives the effect of twinkling stars on a moonless night.

But be careful! The result is so fascinating to watch that operators have been known to forget the coffee break entirely. . . .

```
10 CLS
: PRINT CHR$(02)
20 X=RND(23)
: Y=RND(79)
30 PRINT@ (X,Y), " + ";
40 X=RND(23)
: Y=RND(79)
50 PRINT@ (X,Y), " ~ ";
60 X=RND(23)
: Y=RND(79)
70 PRINT@ (X,Y), " . ";
80 X=RND(23)
: Y=RND(79)
90 PRINT@ (X,Y), " * ";
92 X=RND(23)
: Y=RND(79)
94 PRINT@ (X,Y), " ";
98 GOTO 20
```

DATA FILE RECORDING PROGRAM

Allen G. Barclay
6402 Tamalpais Avenue
San Jose, CA 95120

Although this program was developed for use on a PC-1 pocket computer without a printer, it could be easily adapted to any other computer with or without a printer.

This program provides for memorizing, reviewing, and correcting data before recording it on magnetic tape. It also provides for recalling previously recorded data from tape for making additions or corrections before re-recording it on tape.

Other programs would be used to recall the data from the tape and to perform the desired analysis.

DATA FORMAT

Memory	X (Sequence No.)	Y (The Data)
A(10)	1	73.12
A(11)	2	8.03
A(12)	3	127.44
.	.	.
.	.	.
A(110)	101	11.723
A(111)	102	E99

The X values, or sequence numbers, are generated by the program. The Y values are the input data that you want to store and recall for future use. The last value is always E99, so it can be utilized in the using programs as a trailer to indicate the end of the data.

OPERATING SEQUENCE

1. Set to DEF mode, <SHFT> <Z>
2. "DATA RECORD FILE" <ENTER>
3. "1 = YES, 0 = NO" <ENTER>
4. "NEW DATA FILE?"

<1> <ENTER> denotes that a new file is to be compiled for recording onto magnetic tape. After each "Y=" enter the next item of data. If any error is entered, ignore it and continue making entries. After the last data has been entered, enter E99.

<0> <ENTER> denotes that an existing file on magnetic tape is to be loaded into the computer memory for expansion or correction. The next message is, "GET DATA TAPE READY." <ENTER> will start the loading from the magnetic tape into the computer memory.

5. "LAST X = , Y = " is printed after all the data has been entered into memory either manually or from a tape. <ENTER>.

6. "ADD MORE DATA?"

<1> <ENTER> will provide for entering more data, exactly as described above for compiling a new file. Again, enter E99 after all the new data has been entered.

<0> <ENTER> moves to the next step if no new data is to be added.

7. "REVIEW DATA?"

<1> <ENTER> initiates a print of the X and Y values for each data entry. If a larger number is entered, the review will start with that sequence number. If a number, decimal point, and two-digit numbers are entered, the review will be limited to the range of data starting with the number to the left of the decimal point and ending with the sequence number of the number to the right of the decimal. (i.e., 4.09 would limit the review of the data to the X and Y starting with X = 4 and ending with X = 9.)

<0> <ENTER> denotes that no review is required and that the data is ready for recording. The next message is, "GET TAPE READY." <ENTER> will start recording of the data on magnetic tape.

8. "ALL VALUES CORRECT?" This question always follows the completion of the review of data.

<1> <ENTER> denotes that all the data is correct and is ready for recording. The next message is, "GET TAPE READY." <ENTER> will start the recording of the data on magnetic tape.

<0> <ENTER> will lead to three methods of correcting the data, as described below. Each of these three methods may be skipped, or ended, by entering 0 for the sequence number.

9. "SEQ # OF ERROR?" In response to this question, enter the sequence number of the data that is in error, or 0 if there is no data that needs to be corrected. When "CORRECT Y =" appears, enter the correct value for the data.

10. "DELETE SEQ #" Enter the sequence number of any data that is to be deleted, such as a data value that was entered twice by mistake. The data will be deleted and all the data values that follow will be moved up to close up the gap in the sequence. (A caution: If you are going to correct more than one item of data, remember that the sequence numbers have now been changed.) Enter <0> to move to the next step.

11. "INSERT NEW SEQ #" Enter the sequence number for the data to be inserted. Then enter the new data in response to "NEW Y = ". All the data that follows will be moved to the next higher sequence number to make room for this new data in its correct location. When 0 is entered for the sequence number, the program returns to "REVIEW DATA".

Note: In this article the 'E' in 'E99' is the <Exp> key found on the PC-1 next to the <+> key.

DATA RECORD FILE PROGRAM ("DATAFIL")

```

1 "Z" CLEAR
  : D=9
  : PRINT "DATA RECORD FILE"
  : PRINT "1=YES, 0=NO"
  : INPUT "NEW DATA FILE?";B
  : IF B=1 THEN 20
5 PRINT "GET DATA TAPE READY"
  : INPUT # "MI";A(10)
10 FOR F=10 TO 100
  : IF A(F)=E99 THEN 30
15 NEXT F
20 FOR F=D+1 TO 110
  : INPUT "Y= ";A(F)
  : L=F-9 IF A(F)=E99 THEN 30
25 PAUSE C, A(F)
  : NEXT F
  : A(111)=E99
  : PRINT "MAX DATA"
30 A=F-10
  : D=F-1
  : PRINT "LAST X=";A;"Y=";A(D)
  : INPUT "ADD MORE DATA?";B
  : IF B=1 THEN 20
35 INPUT "REVIEW DATA?";B
  : IF B>0 THEN 45
40 PRINT "GET TAPE READY"
  : PRINT# "MI";A(10)
  : PRINT "END"
  : END
45 E=INT B
  : H=(B-E)*100
  : IF H>0 THEN 55
50 H=A
55 FOR F=9+E TO 9+H
  : C=F-9
  : PRINT C,A(F)
  : NEXT F
60 INPUT "ALL VALUES CORRECT?";B
  : IF B=1 THEN 40
65 INPUT "SEQ # OF ERROR=";B
  : IF B=0 THEN 75
70 F=B+9
  : INPUT "CORRECT Y= ";A(F)
  : GOTO 65
75 INPUT "DELETE SEQ # ";B
  : IF B=0 THEN 85
80 FOR F=B+9 TO A+10
  : A(F)=A(F+1)
  : NEXT F
  : A=A-1
  : GOTO 75
85 INPUT "INSERT NEW SEQ #";B
  : IF B=0 THEN 35
90 FOR F=A+10 TO B+9 STEP -1
  : A(F+1) = A(F)
  : NEXT F
  : F=B+9
  : A=A+1
  : INPUT "NEW Y = ";A(F)
  : GOTO 85

```

Reverse Polish Notation on a Pocket Computer

Joe H. Morrison
Radio Shack Computer Centre
1706 Dundas Street East
Mississauga, Ontario
CANADA

For all you Reverse Polish Notation lovers, here is a program for the PC-1 pocket computer (PC-2 changes follow the main article) that will replace the old Hewlett Packard you threw out! It is an almost perfect simulation of an RPN-calculator, and includes a register shift key, an "exchange X and Y register" key, keys to store and recall up to 38 numbers (in the data memories), and all of the functions you would expect to find in any self-respecting calculator. Oh, and there is a quadratic notation equation solution program, as well!

OPERATION

You will need to make an overlay like this:

A	S	D	F	G	H	J	K	L	=
f	ln	sin	cos	tan	$\frac{x^2}{\sqrt{x}}$	STO	USR	$\frac{1}{x}$	CHS FIX
Z	X	C	V	B	N	M	SPC	ENTER	
+	-	x	$\frac{-}{+}$	Y*	R↓	RCL	ENTER		
INT	FRAC	QUAD	CL REG	EXREG	DEG	RAD	GRAD		

Keyboard Overlay for PC-1
(Hope you can write small!)

To operate the program once it is in the PC-1, use the DEF mode. You should recall that in the DEF mode, the <SHFT> key tells the computer which of the defined keys you want. Note that some of the keys have two functions defined (Note: we will use <UPPER CASE> letters to designate actual keys on the Pocket Computers, and <lower case> to designate RPN functions. This is especially important because the RPN calculator has its own <enter> key.)

For the two function keys, the first function is called by keying <SHFT> and the function key, and the second function is called with <SHFT> <A>, then <SHFT> and the key you want to execute. As in an ordinary calculator, to execute single-operand functions (like SIN, COS, etc.), you must key in the operand, and then press the function key. (For those of you that are very familiar with RPN, all of the functions provided in this program (including CHS and 1/x) automatically enter the result into the stack. The stack, by the way, contained in variables A through D, A being the 'x' register and D being the 't' register.)

The arithmetic functions work exactly as in an RPN calculator with one minor difference that I will get to in a minute. That is, to add two and three, you type <2> <SHFT> <SPC> <3> <SHFT> <Z>. And, of course, the result is stored in the 'stack' enabling you to do calculations like: $(2+3) \times (5+1)$ by typing: <2> <enter> <3> <plus> <5> <enter> <6> <plus> <multiply>. Now, that minor difference I mentioned. It is simply that because of the way the stack works, YOU CANNOT PERFORM A 2-OPERAND FUNCTION USING THE SAME NUMBER FOR BOTH OPERANDS. YOU WILL GET

WRONG ANSWER IF YOU DO! (Therefore, you cannot add 2 and 2, multiply 3 by 3, subtract 1 from 1, etc.) However, do not panic! That problem is taken care of by an interesting design feature of the pocket computer. You can intermix the RPN program with the algebraic operating system of the pocket computer. This means you can key in things like:

$(2+2) \cdot 4$ <enter> $8 \cdot 16$ </ >.

Get it? Here is a more complicated example. Solve:

$$\frac{\left(\frac{3}{8}\right)^3 + \left(4 - \left(\frac{5}{3} \times (3)^3\right)\right)}{2 \times \left(\frac{4}{9} \text{SIN}46^\circ\right)}$$

The required sequence (using the combination of AOS and RPN) is:

3/8 <enter> 3 <Y* > 4 <enter> 5/3 <enter> 3^3 <multiply > <minus > <plus > 2 <enter> 4/9 <enter> 46 <sin > <multiply > <multiply > <divide >.

On the pocket computer, the actual keystrokes are:

3/8 <SHFT > <SPC > 3 <SHFT > 4 <SHFT > <SPC > 5/3 <SHFT > <SPC > 3^3 <SHFT > <C > <SHFT > <X > <SHFT > <Z > 2 <SHFT > <SPC > 4/9 <SHFT > <SPC > 46 <SHFT > <D > <SHFT > <C > <SHFT > <C > <SHFT > <V >

EXTRA FEATURES

User key: The RPN program ends at line 199, so if you want to enter an additional program, you merely have to go to PROgram mode, and start your program at line 200. To execute your program, just type <SHFT > <K > (the key labeled 'user').

Data memories: If you wish to store the number in the display, simply type <SHFT > <J > (labeled 'sto') and when the question mark appears, key in the number of the memory you wish to use. (The number of available memories depends on the length of any additional programs in memory. If there are no additional programs in memory, there are 38 memories available.) Similarly, you can recall STORed numbers by typing <SHFT > <M > (labeled 'rcl') and keying in the number of the memory you wish to recall.

Trigonometry: The second functions of the <N >, <M > and <SPC > keys switch the calculator into DEGREE, RADIAN and GRAD modes, respectively.

Formatting: The second function of the <= > key switches the display back and forth between floating point format and fixed format. (In fixed format the display is fixed to 5 decimal places.) In fixed format, you will get an error if the result of a calculation is greater than 999,999,999. However, the fixed format makes it easier to read the results of the trig functions, since they normally come out in exponential notation.

Quadratic equation solution: The second function of the <C > key will allow you to find the two roots of a quadratic equation (provided they are not imaginary). To use it, type <SHFT > <A > <SHFT > <C >, and then you get 'A=?' enter the X² coefficient. When you get 'B=?', enter the X coefficient, and when you get 'C=?', enter the constant. After a few seconds, both roots will appear simultaneously on the display.

Miscellaneous: The second functions of the <V > and keys are respectively, the 'clear register' and the 'ex-

change X and Y register' keys. When using the latter, the <X > number must be entered before you use the function (i.e., To calculate $6/(3+2)$, you type 3 <enter> 2 <plus > 6 <enter > <exch. reg. > <divide >). The same is true of the <N > key, which is the register shift key—you must enter the number in the display before shifting registers.

An interesting side effect of this system is that you can turn off the pocket computer in the middle of a calculation, and can later pick up where you left off!

All in all, though the system requires more keystrokes for a given calculation than a normal RPN calculator would, I have found that it is faster to use than the regular system of the pocket computer, because I no longer need to type out 'SIN' and 'LOG', etc. Have fun with the program!

In the program listing, SQR should be replaced by the square root symbol.

Note to PC-2 users: The RPN calculator will function properly on the PC-2 with three changes. First, The PC-1 uses single character variable names only. When ever you see two variables or a number and a variable together (e.g., YY or 4X in line 51) separate the items with an '*' (for multiplication.) Second, because of the added '*'s, line 51 will be too long. Split the line into two parts:

```
51 F=0
   : INPUT "A=";X, "B=";Y, "C=";Z
   : W=SQR(Y*Y-4*X*Z)
   : U=(-Y+W)/(2*X)
   : V=(-Y-W)/(2*X)
52 PRINT U,V
   : END
```

The third change is that where you see the word <SHFT > in the article, substitute <DEF >. To add two and three, you type <2 > <DEF > <SPC > <3 > <DEF > <Z >.

```
10 "A" AREAD X
   : F=ABS (F-1)
   : PRINT X
   : END
20 " " AREAD X
   : IF F=0 LET D=C
   : C=B
   : B=A
   : A=X
   : PRINT A
   : END
25 GRAD
   : F=0
   : PRINT X
   : END
30 "Z" AREAD X
   : IF F=1 LET D=C
   : F=0
   : C=B
   : B=A
   : A=INT X
   : PRINT A
   : END
33 IF X<>A LET A=A+X
   : PRINT A
   : END
35 A=B+A
   : B=C
   : C=D
   : PRINT A
   : END
40 "X" AREAD X
   : IF F=1 LET D=C
   : F=0
```



```

: C=B
: B=A
: A=X-INT X
: PRINT A
: END
43 IF X<>A LET A=A-X
: PRINT A
: END
45 A=B-A
: B=C
: C=D
: PRINT A
: END
50 "C" AREAD X
: IF F=0 GOTO 53
51 F=0
: INPUT "A=?"; X, "B=?"; Y, "C=?"; Z
: W=SQR(YY-4XZ)
: U=(-Y+W)/2X
: V=(-Y-W)/2X
: PRINT U, V
: END
53 IF X<>A LET A=AX
: PRINT A
: END
55 A=BA
: B=C
: C=D
: PRINT A
: END
60 "V" AREAD X
: IF F=1 LET D=0
: F=0
: C=0
: B=0
: A=0
: X=0
: PRINT A
: END
63 IF X<>A LET A=A/X
: PRINT A
: END
65 A=B/A
: B=C
: C=D
: PRINT A
: END
70 "B" AREAD X
: IF F=1 LET X=B
: B=A
: A=X
: F=0
: PRINT A
: END
73 IF X<>A LET A=A^X
: PRINT A
: END
75 A=B^A
: B=C
: C=D
: PRINT A
: END
80 "H" AREAD X
: IF F=1 LET D=C
: F=0
: C=B
: B=A
: A=SQR X
: PRINT A
: END
83 D=C
: F=0
: C=B
: B=A
: A=XX
: A=XX

```

```

: PRINT A
: END
90 "D" AREAD X
: IF F=1 LET D=C
: F=0
: C=B
: B=A
: A=ASN X
: PRINT A
: END
93 D=C
: F=0
: C=B
: B=A
: A=SIN X
: PRINT A
: END
100 "F" AREAD X
: IF F=1 LET D=C
: B=0
: C=B
: B=A
: A=ACS X
: PRINT A
: END
103 D=C
: F=0
: C=B
: B=A
: A=COS X
: PRINT A
: END
110 "G" AREAD X
: IF F=1 LET D=C
: F=0
: C=B
: B=A
: A=ATN X
: PRINT A
: END
113 D=C
: F=0
: C=B
: B=A
: A=TAN X
: PRINT A
: END
120 "S" AREAD X
: IF F=1 LET D=C
: F=0
: C=B
: B=A
: A=EXP X
: PRINT A
: END
123 D=C
: F=0
: C=B
: B=A
: A=LN X
: PRINT A
: END
130 "=" AREAD X
: IF F=0 LET D=C
: C=B
: B=A
: A=-X
: PRINT A
: END
132 F=0
: USING
: R=ABS (R-1)
: IF R=1 USING "#####.#####"
133 PRINT X
: END

```

```

140 "L" AREAD X
  : IF F=0 LET D=C
  : C=B
  : B=A
  : A=1/X
  : PRINT A
  : END
141 F=0
  : END
150 "J" AREAD X
  : IF F=0 INPUT Y
  : A(Y+26)=X
  : PRINT X
  : END
151 F=0
  : END
160 "M" AREAD X
  : IF F=0 INPUT Y
  : D=C
  : C=B
  : B=A
  : A=A(Y+26)
  : PRINT A
  : END
161 F=0
  : RADIAN
  : PRINT X
  : END
170 "N" AREAD X
  : IF F=1 LET F=0
  : DEGREE
  : PRINT X
  : END
172 X=A
  : A=B
  : B=C
  : C=D
  : D=X
  : PRINT A
  : END
199 "K"

```

Pocket Computer Bugs, Errors and Fixes

Calendar (26-3529)

Please note the following correction to page 7, example 3 in the Calendar program manual:
 <SHFT> <K> should read <SHFT> <X>

Ode to Micro Idiom

Joe Carr
 3950 Cromart Ct. N.
 Ft. Worth, TX 76133

When speaking the lore of computer technology,
 I offer the layman no mean apology.
 Handshaking, polling, and flag waving appear,
 Even if it's not an election year.
 Addressing, bus drivers, and integration,
 Have nothing to do with desegregation.
 Booting the terminal hurts not a Bit,
 But the CMOS Ram might throw a fit.

Wares, soft, firm, and hard, you'll find by the bunch,
 But the dreaded hidden bug could still eat your lunch.
 So grab your joysticks and hold onto your hair,
 'Cause interchangeable programs are still pretty rare.
 Memory mapped might be the key,
 But what's it all mean?
 Don't ASCII* me.
 *Pronounced ASKY

Pearson <R> Correlation

Galen Currah
 11023 N.E. Glisan St.
 Portland, OR 97220

The Pocket Computer has gone with me to West Africa on demographic and other survey research. This program figures amongst several I have written for statistical analysis.

The Pearson Product-Moment Coefficient of Correlation (r) is calculated, showing the degree of association between up to 50 pairs of numbers. 616 steps and 103 memories are utilized.

The formula followed reads:

$$r = \frac{n \sum xy - \sum x \sum y}{\sqrt{[n \sum x^2 - (\sum x)^2][n \sum y^2 - (\sum y)^2]}}$$

```

10: CLEAR
  : PRINT "PROD MOMT CORRL COEF (R)"
  : INPUT " NO. OF PAIRS: "; A(101)
12: FOR N = 1 TO A(101)
  : M = N + 50
  : PAUSE "PAIR ";N;":""
  : INPUT A(N),A(M)
  : NEXT N
14: FOR N = 1 TO A(101)
  : M = N + 50
  : A(102) = A(102) + A(N) * A(M)
  : A(103) = A(103) + A(N) * A(N)
  : PAUSE "WAIT"
16: A(104) = A(104) + A(N)
  : A(105) = A(105) + A(M) * A(M)
  : A(106) = A(106) + A(M)
  : NEXT N
18: A(107) = A(102) * A(101)
  : A(108) = A(103) * A(101)
  : A(109) = A(104) * A(104)
20: A(110) = A(105) * A(101)
  : A(115) = A(106) * A(106)
  : A(116) = A(106) * A(104)
22: A(113) = A(107) - A(116)
  : A(114) = A(108) - A(109)
  : A(117) = A(110) - A(115)
24: A(118) = A(114) * A(117)
  : LET A(110) = |A(118)|
  : A(118) = A(113) / A(110)
26: A(118) = INT(A(118) * 100) / 100
  : BEEP 1
  : PRINT " R= "; A(118)

```

LOADCOPY and LOADDATA

George Fraser
3403 Bear Creek Drive
Newbury Park, CA 91320

LOADCOPY and LOADDATA are two programs which will load a stand alone machine language utility program which I have written and assembled. I have tested the programs only with Extended Color BASIC 1.0, but I believe they should also work for the 4K computer if they are relocated in memory.

LOADCOPY will create a stand-alone machine language program. Everyone, whether a casual hobbyist or professional programmer, should find this program extremely useful. This program will change a BASIC program in memory by copying any requested program line to a new line. This is useful when you want to move a line to somewhere else in the program. Just copy the old line to the new location and delete the old line. You don't have to retype the whole thing. Or if you need a series of BASIC lines which are almost the same, you can use the copy repeatedly and then use "edit" to put the lines into final form. It is a very valuable tool for program development and maintenance.

I keep my "COPY" program on a cassette tape in machine language format. Then whenever I need it, I do the following:

```
CLEAR 100,16085      'THIS RESERVES 290 BYTES FOR COPY
CLOADM "COPY"
EXEC                 'OR EXEC 16086
```

To get your own copy of the program, type in my BASIC program "LOADCOPY" carefully. After it will run O.K. and you have double checked it for accuracy, CSAVE it to tape. Then put in a cassette for recording and save the machine language program with:

```
CSAVEM "COPY", 16086, 16383, 16086
```

Make sure you have program "LOADCOPY" CSAVED on tape before you try to "EXEC 16086" for the first time. Any mistake in the machine language code can easily wipe out everything in memory. Owners of the 4K Color Computer should change lines 30 and 40 to:

```
30 CLEAR 100, 3797
40 S = 3798
```

and then "EXEC 3798." The machine language program can be relocated to any protected area of memory by changing lines 30 and 40.

It overlays the "EXEC" line on the screen with "COPY" followed by a cursor character. Type in the line number to be copied (the source line), a comma, and the new line number (the target line number). Press <ENTER>. If you omit the target line number, it defaults to one greater than the source.

```
COPY 30, 200      'WILL COPY LINE 30 TO LINE 200
COPY 120          'WILL COPY LINE 120 TO LINE 121.
```

If COPY generates one of the following error messages, your BASIC program was not changed.

- ?OV Overflow. Source of target line number is greater than 63999.
- ?UL Undefined line. The source line number doesn't exist.
- ?TG Target error. The target line number already exists. Delete it first, and then do the copy.
- ?OM Out of memory. Try to free up more memory by clearing less string space or deleting some program comments.

The backspace key doesn't work in COPY, but you can interrupt the program with the <BREAK> key which causes an "?OV ERROR."

The machine language program is the shortest and most refined version of "COPY" that I have developed. It uses the hardware stack to store intermediate results, which guarantees a complex assembly language program.

The first half of the program is occupied with keying in the line numbers and converting them to binary. It scans the BASIC program to find the source line number; then it scans the BASIC program again to find where the target line is to be inserted and to make sure that the target line number does not already exist.

It has to make sure there is enough free memory left before it starts to do the copy; then it opens up enough room in your BASIC program at the target address for the copy. At long last, it copies the source line to the target area and slaps in your target line number. Finally, it has to update BASIC's next line pointer in each BASIC line from the target line to the end of the program.

All in all, this little program has a lot of very complex things to do just to copy one line!

```
10 ' "LOADCOPY" FOR THE COLOR COMPUTER
20 ' BY G.FRASER MARCH 1982
30 CLEAR 100, 16085
40 S=16086 'START ADDRESS
50 E=S+297 'END ADDRESS
60 FOR A=S TO E
70 READ B: POKE A, B
   : C=C+B
80 NEXT
90 IF C=34462 THEN PRINT "CSAVEM";S;E;S
   : END ELSE PRINT C
   : STOP
100 DATA 222, 136, 51, 200, 224, 204, 67, 79, 237, 19
110 DATA 204, 80, 89, 237, 193, 204, 96, 141, 237, 19
120 DATA 198, 26, 167, 192, 90, 38, 251, 51, 200, 229
130 DATA 50, 120, 141, 22, 175, 100, 129, 13, 39, 8
140 DATA 141, 14, 109, 97, 42, 6, 174, 100, 48, 1
150 DATA 141, 71, 175, 102, 32, 80, 204, 0, 255, 237
160 DATA 98, 95, 237, 100, 189, 161, 193, 129, 3, 39
170 DATA 39, 129, 13, 39, 46, 129, 44, 39, 40, 129
180 DATA 48, 37, 237, 129, 57, 34, 233, 141, 40, 128
190 DATA 112, 167, 99, 236, 100, 142, 0, 9, 227, 100
200 DATA 37, 8, 48, 31, 38, 248, 227, 98, 36, 208
210 DATA 134, 96, 167, 196, 204, 79, 86, 32, 76, 141
220 DATA 8, 174, 100, 140, 249, 255, 34, 238, 57, 136
```

```

230 DATA 64, 198, 141, 237, 192, 57, 134, 96, 167, 196
240 DATA 16, 158, 25, 31, 35, 16, 174, 196, 39, 37
250 DATA 174, 66, 172, 100, 38, 243, 239, 228, 236, 196
260 DATA 163, 228, 237, 98, 16, 158, 25, 31, 35, 16
270 DATA 174, 196, 39, 46, 174, 66, 172, 102, 34, 40
280 DATA 38, 241, 204, 84, 71, 32, 10, 204, 85, 76
290 DATA 32, 5, 204, 79, 77, 50, 98, 48, 140, 10
300 DATA 237, 1, 198, 10, 189, 185, 166, 50, 104, 57
310 DATA 63, 88, 88, 32, 69, 82, 82, 79, 82, 13
320 DATA 17, 163, 228, 34, 6, 236, 228, 227, 98, 237
330 DATA 228, 220, 27, 31, 2, 227, 98, 52, 70, 48
340 DATA 232, 206, 172, 225, 37, 202, 31, 1, 159, 27
350 DATA 159, 29, 159, 31, 16, 172, 228, 39, 6, 166
360 DATA 162, 167, 130, 32, 245, 52, 16, 174, 100, 16
370 DATA 174, 102, 166, 128, 167, 192, 49, 63, 38, 248
380 DATA 53, 70, 174, 102, 175, 66, 237, 196, 31, 3
390 DATA 236, 196, 39, 169, 227, 98, 32, 244

```

"LOADDATA" will create a stand-alone machine language program I call "DATA". I use this program whenever I want to include a machine language subroutine in a BASIC program in the form of "DATA" statements which will be read and POKEd into memory to load the subroutine.

I usually keep "DATA" on a cassette tape in machine language format which will load starting at address 15360.

Suppose I have written a machine language subroutine which will be called by BASIC program "X", and I have the machine language code already in memory locations 16200-16383. (For example, I may have hand assembled it and POKEd it into memory.) Now I want to convert it into BASIC data statements to go on to the end of program "X".

Suppose I also have the calling program "X" in memory, complete except for the data statements. Here is where my program "DATA" is really a big help. I will now do the following:

```

CLEAR 100,15359 'THIS WILL PROTECT MEMORY FOR
                'DATA".
CLOADM "DATA"  'LOADS AT ADDRESSES 15360-15676
EXEC           'OR EXEC'D 15360

```

Of course, I have to make sure that "DATA" does not overlay my other machine language subroutine. As soon as I enter "EXEC", that line on the screen is overlaid with "DATA" followed by a cursor character. I now key in the desired range of memory that I want to capture so that the line now looks like:

```
DATA 16200,16383
```

The two numbers are the starting and ending address in decimal. When I press <ENTER>, "DATA" instantly creates the data line on the end of my BASIC program "X". "DATA" can generate three error messages:

- ?OV Overflow. You typed an address greater than 65535.
- ?SQ Sequence Error. The second address is less than the first.
- ?OM Out of Memory. Perhaps some, but not all, of the DATA statements were created.

The backspace key doesn't work in "DATA" but you can interrupt the program with the <BREAK> key, which causes a "?OV ERROR".

To get your own copy of the program, type in my BASIC program "LOADDATA" carefully. After it will run O.K. and you have double checked it for accuracy, CSAVE it to tape; then put in a cassette for recording and save the machine language program with:

```
CSAVEM "DATA", 15360, 15676, 15360
```

Make sure you have program "LOADDATA" CSAVED on tape before you try to "EXEC 15360" for the first time. Any mistake in the machine language code can easily wipe out everything in memory. Owners of the 4K Color Computer may change lines 30 and 40 to:

```
30 CLEAR 100,3778
40 S=3779
```

and then "EXEC 3779". The machine language program can be relocated to any protected area of memory by changing lines 30 and 40.

I created my first copy of "DATA" with my 6809 assembler. I then used the machine language program to create the DATA statements in the "LOADDATA" BASIC program.

You can customize "DATA" to some extent with the following patches. In each case, "S" is the start address you put in line 40 of "LOADDATA".

1. The line number increment for successive "DATA" line is set to 10. You can vary that value from 1 to 15 by POKing that value into S+218. You must not exceed 15.
2. Ten numbers are put on a DATA line. If you want more or fewer, POKE that number into S+228. If you exceed 15, it probably won't catch an "OM" error in time.
3. "DATA" generates a space after each comma for readability. To save memory, you can eliminate the space with POKE S+284,160.

The program LOADDATA.

```

10 ' "LOADDATA" FOR THE COLOR COMPUTER
20 ' BY G.FRASER MARCH 1982
30 CLEAR 100, 15359
40 S=15360 'START ADDRESS
50 E=S+316 'END ADDRESS
60 FOR A=S TO E
70 READ B
   : POKE A,B
   : C=C+B
80 NEXT
90 IF C=31862 THEN PRINT "CSAVEM";S;E;S
   : END ELSE PRINT C
   : STOP
100 DATA 222, 136, 51, 200, 224, 134, 2, 31, 139,
    204
110 DATA 68, 65, 237, 193, 134, 84, 237, 193,
    204, 96
120 DATA 141, 237, 193, 198, 26, 167, 192, 90,
    38, 251
130 DATA 51, 200, 229, 141, 49, 159, 4, 129, 13,
    39
140 DATA 24, 141, 41, 13, 1, 43, 16, 156, 4, 36
150 DATA 14, 141, 5, 204, 83, 81, 32, 89, 134, 96
160 DATA 167, 196, 57, 158, 4, 159, 6, 141, 245,
    190
170 DATA 0, 25, 31, 18, 174, 164, 39, 119, 236,
    34
180 DATA 221, 8, 32, 244, 204, 0, 255, 221, 0, 95
190 DATA 221, 8, 221, 2, 189, 161, 193, 129, 3,
    39
200 DATA 39, 129, 13, 39, 67, 129, 44, 39, 61,
    129
210 DATA 48, 37, 237, 129, 57, 34, 233, 141, 56,
    128
220 DATA 112, 151, 1, 220, 2, 142, 0, 9, 211, 2
230 DATA 37, 8, 48, 31, 38, 248, 211, 0, 36, 208
240 DATA 50, 98, 141, 170, 204, 79, 86, 48, 140,
    10
250 DATA 237, 1, 204, 0, 10, 31, 139, 126, 185,
    166
260 DATA 63, 88, 88, 32, 69, 82, 82, 79, 82, 13
270 DATA 141, 3, 158, 2, 57, 138, 64, 198, 141,
    237
280 DATA 192, 57, 79, 95, 237, 161, 31, 139, 31,
    33

```

```

290 DATA 159, 27, 159, 29, 159, 31, 57, 52, 32,
    48
300 DATA 232, 142, 172, 225, 34, 7, 141, 230,
    204, 79
310 DATA 77, 32, 190, 31, 35, 158, 8, 48, 10, 49
320 DATA 34, 175, 161, 159, 8, 158, 4, 198, 10,
    215
330 DATA 10, 204, 134, 32, 237, 161, 15, 11, 166,
    128
340 DATA 198, 48, 129, 100, 37, 5, 128, 100, 92,
    32
350 DATA 247, 141, 51, 198, 48, 129, 10, 37, 5,
    128
360 DATA 10, 92, 32, 247, 141, 38, 31, 137, 203,
    48
370 DATA 231, 160, 156, 6, 34, 17, 10, 10, 39, 7
380 DATA 204, 44, 32, 237, 161, 32, 205, 141, 8,
    159
390 DATA 4, 32, 160, 141, 2, 32, 141, 79, 167,
    160
400 DATA 16, 175, 196, 57, 13, 11, 38, 6, 193, 48
410 DATA 39, 4, 12, 11, 231, 160, 57

```

Mailing Labels on the CoCo

David L. Banaszak
168 Stratmore
New Carlisle, OH 45355

The program MAIL allows me to print mailing address labels for up to 105 names and addresses. Before loading or entering MAIL, type PCLEAR 1 or POKE 25, 6 NEW to ensure sufficient memory. Basically MAIL is an expansion of the "Inventory Shopping List" (ISL) program on page 295 of the Color BASIC Manual. Also, several finer features of MAIL were inspired by previous newsletters. I'm still refining MAIL, but I have used this version to print address labels for our Christmas card list and several non-profit club's mailing lists. The equipment and supplies I use to run MAIL are:

- 1). Line Printer VII
- 2). Color Computer with 16K Extended Color BASIC
- 3). Tape recorder and tape
- 4). Television
- 5). 9 1/2" fanfold paper (26-1433)
- 6). Dry gum labels (26-1456)

After pressing <ENTER>, the screen displays:

- 1). The memory left
- 2). My name, street, city, state, zip, and phone number
- 3). The 9 option main menu.

Option 1 on the main menu (lines 190-340) requests input items in the following format:
LAST NAME, FIRST NAME, STREET, CITY, STATE ZIP *
ACPH CODE.:

For each item, the four commas and * must be typed. Characters after the * are for additional information such as phone numbers and the : serves as a separator if one wants to include 2 or 3 addresses per input item. L is used as a current item string length counter and should not be allowed to exceed 255. LL is a coarse total string character counter which should be kept less than about 7000 to avoid running out of string space. INKEY\$ is used for input and PEEK(343) is used to detect left arrow backspace ←. Option 2 (lines 350-

460) replaces items, Option 3 (lines 220-340) adds items to the list and Option 4 (lines 470-590) deletes items all similar to program ISL. Option 5 (lines 600-780) allows the user to view the items on the screen as in ISL and/or on the printer. Option 6 (lines 790-950) saves items on a user provided file name and Option 7 (lines 960-1150) loads items from a user specified file name.

Options 8 and 9 are not included in ISAL. Option 8 (lines 1160-1300) separates each string item S\$(n) into 5 parts LN\$(1) through LN\$(5). The array LN\$ is then printed in address label format as directed by the choices selected in lines 1170, 1175 and 1177. Currently option 9 (lines 1330-4000) contains two working functions. Lines 1400-1590 will reformat N items containing three addresses each separated by colons (:), in 3N times containing 1 address per item. Lines 1600 to 1999 alphabetized the items by last names since the last name is first in each element of S\$. Expanded functions I hope to add to the option 9 section are the ability to search the entire list for some user specified character string and an ability to sort by user specified parameter such as street or city or phone or zip.

```

10 CLEAR 8000
   : DIM S$(105)
   : DIM LN$(5)
20 LL = 0
30 CLS
33 PRINT @7, "MAIL 13-30NOV81"
34 PRINT @39, "MEM LEFT="; MEM
35 PRINT @64, "BY DAVID BANASZAK, 168 STRATMORE,
    NEW CARLISLE, OH45344*513-8499140"
50 PRINT @134, "(1) INPUT ITEMS"
60 PRINT @166, "(2) REPLACE ITEMS"
70 PRINT @198, "(3) ADD TO THE LIST"
80 PRINT @230, "(4) DELETE ITEMS"
90 PRINT @262, "(5) PRINT ALL ITEMS"
100 PRINT @294, "(6) SAVE ITEMS ON TAPE"
110 PRINT @326, "(7) LOAD ITEMS FROM TAPE"
120 PRINT @358, "(8) ADDRESS LABELS"
130 PRINT @390, "(9) SEARCH&SORT"
140 PRINT @427, "(1-9)";
150 INPUT M
160 IF M < 0 OR M > 9 THEN 20
170 ON M GOSUB 190, 350, 220, 470, 600, 790, 960, 1160, 1330
180 GOTO 30
190 'INPUT/ADD
200 IF Y > 1 THEN RETURN
210 Y = 1
220 L = 0
   : CLS
   : PRINT @3, "LN, FN, ST, CY, STZP * ACPHCODE:"
230 PRINT @34, "PRESS <ENTER> WHEN DONE"
240 PRINT
   : PRINT "ITEM" Y;
250 S$(Y) = ""
260 AS = INKEY$
   : IF AS = "" THEN 260
274 IF PEEK(343) <> 247 THEN 280
275 L = L - 1
   : IF L < 0 THEN 220
276 S$(Y) = LEFT$(S$(Y), L)
279 GOTO 300
280 IF AS = CHR$(13) THEN 310
290 S$(Y) = S$(Y) * AS
300 L = LEN(S$(Y))
301 PRINT @129, "LL="; LL; "L="; L
   : PRINT @160, S$(Y)
302 IF L = 0 THEN 220
305 GOTO 260
310 IF S$(Y) = "" THEN RETURN
320 Y = Y + 1
330 LL = LL + L
   : GOTO 220
340 '
350 'REPLACE
360 N = 0
   : L = 0
370 CLS
   : PRINT @9, "REPLACE ITEMS"
380 PRINT @34, "PRESS <ENTER> WHEN FINISHED"
390 PRINT
   : INPUT "ITEM NO. TO REPLACE"; N
400 PRINT "OLD ITEM WAS: "; S$(N)
   : LL=LL-LEN(S$(N))
   : S$(N) = ""
405 IF N = 0 THEN RETURN
410 PRINT "REPLACEMENT";
420 AS = INKEY$
   : IF AS = "" THEN 420
430 IF AS = CHR$(13) THEN 460
431 IF PEEK(343) <> 247 THEN 440
432 L = L - 1
   : IF L < 0 THEN 360

```

```

433 SS(N) = LEFT$(SS(N), L)
434 GOTO 441
440 SS(N) = SS(N) + AS
441 L = LEN(SS(N))
444 PRINT @129, "LL="; LL; "L="; L
      : PRINT @160, SS(N)
445 IF L = 0 THEN 420
450 GOTO 420
460 LL = LL + L
      : GOTO 350
470 'DELETE
480 N = 0
490 CLS
      : PRINT @9, "DELETE ITEMS"
500 PRINT @34, "PRESS <ENTER> WHEN FINISHED"
510 PRINT
      : INPUT "ITEM TO DELETE"; M
520 IF M > Y - 1 THEN 510
530 IF M = 0 THEN RETURN
540 FOR X = M TO Y - 2
550 SS(X) = SS(X+1)
560 NEXT X
570 SS(X) = ""
580 Y = Y - 1
590 GOTO 470
600 ' PRINT
610 CLS
620 PRINT @134, "1 SCREEN ONLY"
630 PRINT @166, "2 PRINTER ONLY"
640 PRINT @198, "3 SCREEN & PRINTER"
650 INPUT M
      : IF M < 1 OR M > 3 THEN 620
660 ON M GOTO 670, 750, 670
670 FOR X = 1 TO Y - 1 STEP 15
680 FOR Z = X TO X + 14
690 PRINT Z; SS(Z)
700 NEXT Z
710 INPUT "PRESS <ENTER> TO CONTINUE"; C0
720 NEXT X
730 IF M < 3 THEN 780
750 FOR X = 1 TO 105
760 IF SS(X) <> "" THEN PRINT@-2, X; SS(X)
770 NEXT X
780 RETURN
790 'SAVE ITEMS
800 CLS
      : PRINT @135, "SAVE ITEMS ON TAPE"
810 INPUT "TAPE FILE NAME"; FF$
820 MOTOR ON
      : AUDIO ON
830 PRINT @224, "POSITION TAPE MOTOR & AUDIO IS ON PRESS <ENTER> WHEN READY"
      : INPUT R0
840 MOTOR OFF
      : AUDIO OFF
850 PRINT @356, "PRESS PLAY AND RECORD"
860 PRINT @388, "PRESS <ENTER> WHEN READY"
870 INPUT R0
880 OPEN "O", #-1, FF$
890 FOR X = 1 TO Y - 1
900 PRINT #-1, SS(X)
910 NEXT X
920 CLOSE #-1
930 CLS
      : PRINT "FILENAME IS "; FF$
940 PRINT "STOP RECORDER & <ENTER> WHEN READY"
      : INPUT R0
950 RETURN
960 ' LOAD ITEMS
970 CLS
      : PRINT @136, "LOAD ITEMS FROM TAPE"
980 INPUT "TAPE FILE NAME"; FF$
990 MOTOR ON
      : AUDIO ON
1000 PRINT @235, "REWIND TAPE MOTOR & AUDIO ON"
1010 PRINT "PRESS <ENTER> WHEN READY"
      : INPUT R0
1020 MOTOR OFF
      : AUDIO OFF
1030 PRINT @388, "PRESS PLAY THEN <ENTER> WHEN READY"
1040 INPUT R0
1050 OPEN "I", #-1, FF$
1060 Y = 1
1070 IF EOF(-1) THEN 1120
1080 INPUT #-1, SS(Y)
1090 PRINT SS(Y)
1095 LL = LL + LEN(SS(Y))
1100 Y = Y + 1
1110 GOTO 1070
1120 CLOSE #-1
1130 PRINT "FILE "; FF$; " HAS BEEN LOADED LL="; LL
1140 PRINT "STOP RECORDER & <ENTER> WHEN READY"
      : INPUT R0
1150 RETURN
1160 CLS
      : PRINT @2, "THIS IS THE AREA FOR ADDRESS LABELS"
1170 INPUT "FIRST RECORD, LAST RECORD"; I1, I2
1175 INPUT #0 FOR SCREEN 2 FOR PRINTER"; U
      : IF U = -1 OR U > 2 OR U < 0 THEN 1175
1177 CLS
      : PRINT @33, "INSERT LABEL PAPER HERE-# OF COLUMNS 1 OR 2 <ENTER>"
      : INPUT CC
      : IF CC < 1 OR CC > 2 THEN 1177
1180 FOR I = I1 TO I2
1190 RS = SS(I)
1200 FOR J = 1 TO 5
1210 P = INSTR(1, RS, ",")
1215 IF J = 5 THEN P = INSTR(1, RS, ",*")
1220 IF P = 0 THEN 1760

```

```

1230 L = P - 1
1240 LN$(J) = LEFT$(RS, L)
1250 RS = MID$(RS, P+1)
1260 NEXT J
1265 LN$(1) = LN$(2) + " " + LN$(1)
      : LN$(5) = LN$(4) + " " + LN$(5)
1268 ON CC GOTO 1272, 1280
1272 PRINT#-U, ""
1273 PRINT#-U, TAB(1) LN$(1)
1274 PRINT#-U, TAB(1) LN$(3)
1275 PRINT#-U, TAB(1) LN$(5)
1276 PRINT#-U, ""
      : PRINT#-U, ""
1277 GOTO 1290
1280 PRINT#-U, ""
1281 PRINT#-U, TAB(1) LN$(1)
      : TAB(41) LN$(1)
1282 PRINT#-U, TAB(1) LN$(3); TAB(41) LN$(3)
1283 PRINT#-U, TAB(1) LN$(5); TAB(41) LN$(5)
1284 PRINT#-U, ""
      : PRINT#-U, ""
1289 FOR J = 1 TO 999
      : NEXT J
1290 NEXT I
1300 RETURN
1330 CLS
      : PRINT@5, "SEARCH & SORT AREA"
1340 PRINT @134, "1-3 LINE TO 1"
1350 PRINT @166, "2-SORT BY LAST NAME"
1360 PRINT @198, "3-SEARCH"
1370 PRINT @230, "4-SORT BY ITEM#"
1380 PRINT @262, "5-DONE"
1390 INPUT M
      : IF M < 0 OR M > 5 THEN 1330
1400 ON M GOTO 1420, 1610, 1330, 1330, 4000
1410 '3 TO 1
1420 IF Y < 2 OR Y > 34 THEN 1330
1430 FOR X = Y TO 1 STEP-1
1440 XN = X * 3
1450 'DERIVE 3 PARTS HERE
1460 SS(XN) = SS(X)
      : SS(X) = ""
      : 'PRACTICE
1470 FOR I = 2 TO 1 STEP-1
1480 P = INSTR(1, SS(XN), ",")
1490 IF P = 0 THEN 1550
1500 L = P - 1
1510 SS(XN-1) = LEFT$(SS(XN), L)
1520 SS(XN) = MID$(SS(XN), P + 1)
1530 NEXT I
1570 NEXT X
1580 Y = Y + 3 - 2
1585 SOUND 128, 99
1590 GOTO 1330
1600 'SORT BY LAST NAME
1610 CLS
      : PRINT "SORTING-WAIT"
1620 FOR X = 1 TO Y - 2
1630 FOR I = X + 1 TO Y - 1
1640 IF SS(I) < SS(X) THEN RS = SS(X)
      : SS(X) = SS(I)
      : SS(I) = RS
1650 NEXT I
      : NEXT X
1999 GOTO 1330
4000 RETURN

```

MAIL 13-30NOV81
MEN LEFT= 10134
BY DAVID BARASZAK, 168 STRATHORE,
MEN CARLISLE, OH45314+513-8439140
(1) INPUT ITEMS
(2) REPLACE ITEMS
(3) ADD TO THE LIST
(4) DELETE ITEMS
(5) PRINT ALL ITEMS
(6) SAVE ITEMS ON TAPE
(7) LOAD ITEMS FROM TAPE
(8) ADDRESS LABELS
(9) SEARCH&SORT
(1-9)?

TRS-80®

Volume 4, Issue 10

November, 1982

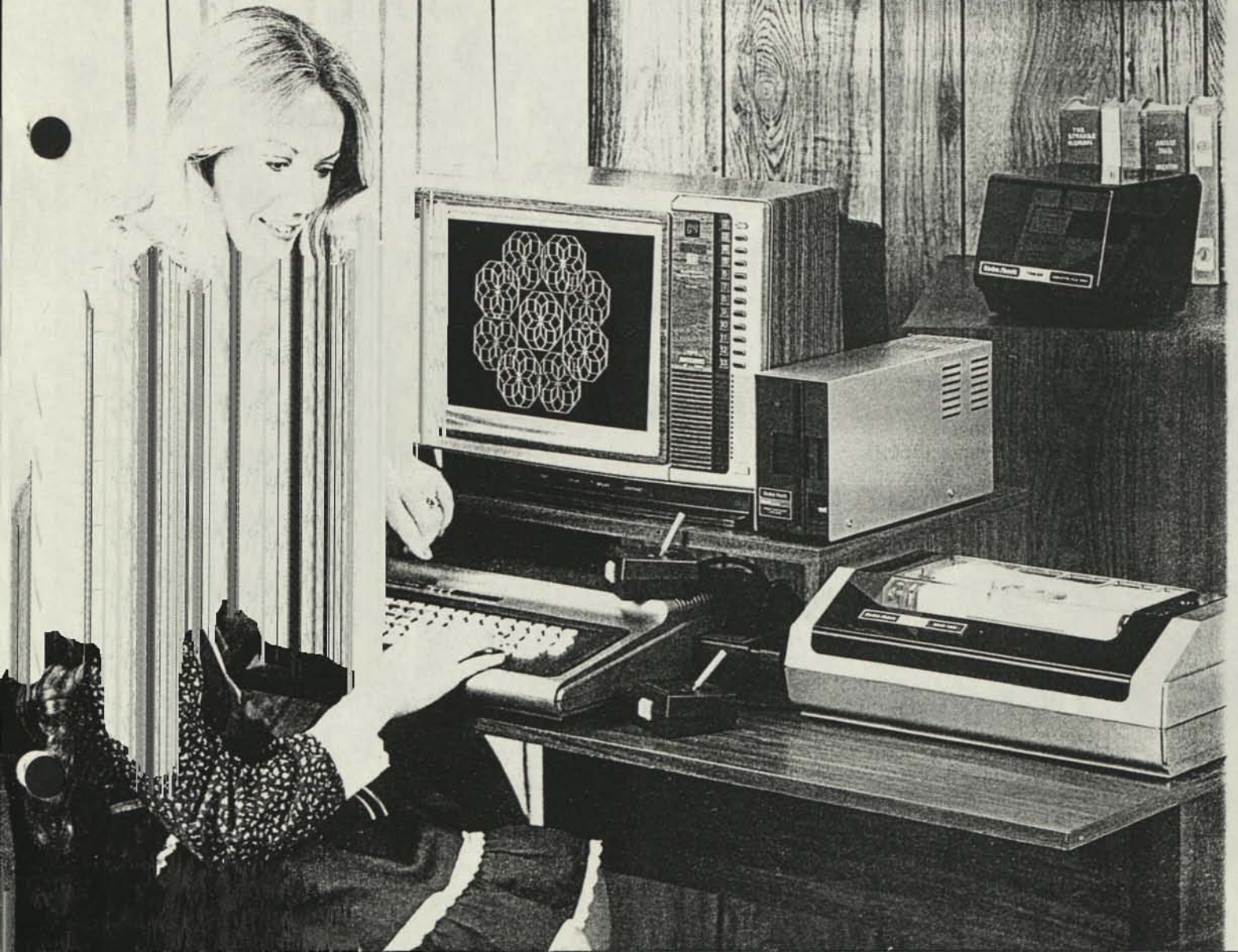
Price \$1.50

Microcomputer News

Printer Issue

Color LOGO

**Christmas Tree
Letterhead**



TRS-80[®] Microcomputer News

Reproduction or use, without express written permission from Tandy Corporation of any portion of the Microcomputer News is prohibited. Permission is specifically granted to individuals to use or reproduce material for their personal, non-commercial use. Reprint permission for all material (other than the SATN material), with notice of source, is also specifically granted to non-profit clubs, organizations, educational institutions, and newsletters.

TRS-80 Microcomputer News is published monthly by Radio Shack, a division of Tandy Corporation. A single six month subscription is available free to purchasers of new TRS-80 Microcomputer systems with addresses in the United States, Puerto Rico, Canada and APO or FPO addresses. Subscriptions to other addresses are not available.

The subscription rate for renewals and other interested persons with U.S., APO or FPO addresses is twelve dollars (\$12.00) per year, check or money order. Single copies of the Microcomputer News may be purchased from Radio Shack Computer Centers or Computer Departments for \$1.50 suggested retail each. The subscription rate for renewals and other interested persons with Canadian addresses is Fifteen dollars (\$15.00) per year, check or money order in U.S. funds. All correspondence related to subscriptions should be sent to: Microcomputer News, P.O. Box 2910, Fort Worth, Texas 76113-2910.

Retail Prices in this newsletter may vary at individual stores and dealers. The company cannot be liable for pictorial and typographical inaccuracies.

Back issues of Microcomputer News prior to January, 1981 are available through your local Radio Shack store as stock number 26-2115 (Suggested Retail Price \$4.95 for the set). Back issues of 1981 copies are available as stock number 26-2240 (Suggested Retail Price \$9.95 for the set).

The TRS-80 Newsletter welcomes the receipt of computer programs, or other material which you would like to make available to users of TRS-80 Microcomputer systems. In order for us to reprint your submission, you must specifically request that your material be considered for reprinting in the newsletter and provide no notice that you retain copyrights or other exclusive rights in the material. This assures that our readers may be permitted to photocopy and use your material without creating any legal hassles.

Material for publication should be submitted on magnetic media (tape, disk, or CompuServe). If you submit material on tape or disk, and it is accepted for publication, we will send you two cassettes or diskettes for each one you sent us. Cassettes will come from our box of mixed blank cassettes. If you submit material on CompuServe, and we think we may use the material, we will extend your Microcomputer News subscription by six months for each article accepted. If you are submitting material over CompuServe, please include your name and address or your subscription number so we can find you. If the material is very short, send it to us in E-Mail. If you have more than a few lines, you need to place the material in the ACCESS area of CompuServe and then let us know it is there by leaving a message on E-Mail.

Material may be submitted by mail to P.O. Box 2910, Fort Worth, Texas 76113-2910, or through CompuServe. The Microcomputer News' CompuServe user ID number is 70007.535.

Notes to Program Users:

Programs published in the Microcomputer News are provided as is, for your information. While we make reasonable efforts to ensure that the programs we publish here work as specified, Radio Shack can not assume any liability for the accuracy either of the programs themselves, or of the results provided by the programs.

Further, while Microcomputer News is a product of Radio Shack, the programs and much of the information published here are not Radio Shack products, and as such can not be supported by our Computer Customer Service group. If you have questions about a program in the Microcomputer News, your first option is to write directly to the author of the program. When possible, we are now including author's addresses to facilitate communications. If the address is not published, or if you are not happy with the response you get, please write us here at Microcomputer News. We will try (given the limited size of our staff) to find an answer to your question and, in many cases, will publish the answer in an upcoming issue of Microcomputer News.

Trademark Credits

CompuServe [™]	CompuServe, Inc.
Dow Jones [™]	
NEWS/RETRIEVAL	
Service [®]	Dow Jones & Co., Inc.
VisiCalc [®]	VisiCorp, Inc.
Program Pak [™]	Tandy Corporation
SCRIPSIT [™]	Tandy Corporation
TRSDOS [™]	Tandy Corporation
TRS-80 [®]	Tandy Corporation

Contents:

Color Computer

Programs	
Christmas Tree Letterhead	by Art Voisard 37
Football Program Pak Tip	by Matthew Staller 40
Navigation with a Color Computer	by L. McNeil 31

Computer Customer Service

What is an Operating System? 5
------------------------------	---------

Data Bases

CompuServe 7
Megawars	
Dow Jones 9
More Quotes and a Whole World of News	
Profile 11
Profile III+: Optimizing Storage Capacity	
VisiCalc/SATN 14
Introducing SATN, The Journal for VisiCalc Users	

Education

Color LOGO Now Available for the TRS-80 Color Computer 43
microResearch Problems 10
Using Your Printer to Make Math Worksheets for Grades K-8 41

Fort Worth Scene

 2
--	---------

Model I/III

Bugs, Errors, and Fixes	
Medical Office System (26-1568) 26
Model III Disk Course (26-2014) 26

Products

Business Checkwriter (26-1585) 15
LDOS—The New Operating Systems for Your TRS-80	by Logical Systems, Inc. 23
Note to SuperSCRIPSIT Users with the DMP-200 20

Programs

Budget Management and the LP VIII	by Ruth Shackelford 6
A Pascal Division Program	by J. Wood 26
Quick Printer Graph Routine	by G. Patek 36
Single Disk FORTRAN	by T. Daneliuk 43
Software Driver for Model I Voice Synthesizer	by S.D. Griffis 22
Tape Mailing Lists and Two Across Labels	by Randy Blinn 10

General Interest

Binding Fan Fold Paper	by E. Kooi 10
------------------------	---------------------

Model II

Model II Scripsit 2.0, LP VIII and the New Printers 30
---	----------

Programs

Changing Print Size on the Line Printers V and VI	by Robert Graham 28
---	---------------------------

Notes on Previous Newsletters

 20
--	----------

January, 1982

Cassette Telephone File	
-------------------------	--

February, 1982

Mailing List Files	
Random Files	
Typewriter	

March, 1982

Blinking Prompt and INKEY\$	
William Barden Jr.'s Series	

May, 1982

SuperSCRIPSIT—an Overview	
---------------------------	--

June, 1982

Disk Head Cleaner	
-------------------	--

Peripherals

Have We Got Printers for You	by Linda Miller 16
------------------------------	--------------------------

Programs

Descenders on the LP VII	by Joe Diel 29
Line Printer Set Up	by William Allen 27
More TAB and USING in a Total Inventory Program	by Robert Perkins 27
Printing Titles	by Bill Christ 26

Pocket Computer

Word Processing on the Pocket Computer	by Don Hergert 32
--	-------------------------

Programs

Circle Subroutine for PC-2 36
----------------------------	----------

View from the 7th Floor

	by Jon Shirley 4
--	------------------------

Prices shown in TRS-80 Microcomputer News are in U.S. Funds.

What is an Operating System?

Whenever we have a program which requires disk access, there are really two ways to do the disk access. We can write into the program all the steps required to perform the access, or we can call upon the Disk Operating System (DOS) to provide those services for us. If we are writing the routines into the program, we must be very consistent, and must set up a means to keep track of whatever is on the disk in our program. If, however, we choose to use an operating system approach, we have the consistency of the system, and can save considerable time and effort in programming by calling upon the system to do all of our routine "housekeeping" functions.

Let's look at the difference between a tape operating system and a disk operating system. Don't be silly, I hear someone cry. There's no such thing as a tape operating system. Who ever heard of TRSTOS? Well, you're right. However, there are many similarities between how you handle tape program and data files and how a disk operating system handles disk files. For example, if you want to load the best little program to do whatever it is you want to do, on a tape system you would first find the tape by looking through your shoe box full of tapes until you find the cassette on which you think this program resides. Up to this point, I must admit, it's pretty much like finding the right disk. However, with the disk, we just shove it into the drive and type "DIR" to see what's on it. With the tape, we have to look at the label and see if we can make out what WE wrote on it. I don't know about you, but my computer prints better than I do. Anyway, after locating the tape, we then must try to use the tape counter to locate the specific section of tape that we want. Of course, we could use the old tried and true method of listening to the recorder's output, but that's not much more fun. With the disk and its operating system, we just type in something like "BASIC PROG1 -F:3" at TRSDOS Ready. This tells the Computer (by way of the Operating System) that we wish to load Disk BASIC (which allows us to use many features of the operating system), load a program called PROG1 and allow buffer space for three files. The operating system then carries out our wishes. (Did we lose you on file buffers? If we want to have disk access from BASIC for data files, we must have BASIC reserve space in memory to accumulate what will be written to disk. This way we can write a "chunk" to disk at one time. This area of memory is known as a buffer.)

The only hardware requirement is that the disk drive be able to find and read a "boot track" which will then tell it where to find the rest of the required information. Disk Operating Systems are usually on the Disk where they will be used. This makes it relatively easy to change the way the operating system works, if that should become necessary.

A notable exception to the DOS being on the disk is our own Color Computer. In this case, the DOS is in a ROM-PACK. Some advantages to this are that you only need one copy of the operating system, not a copy on each disk that will be used in Drive 0 as is required on the Models I, II,

III and 16. Also, the space normally occupied by the operating system is now available for you to store programs and data on the disk. Also, it cannot be damaged as easily as the information on a disk can be.

Now we will try to answer some of your questions about Operating Systems, particularly about TRSDOS 4.1 for the Model II/16.

Question: How can I get a directory of a 'saved' diskette?

Answer: <RESTORE 0 [DIR]>

Question: How can I get a directory of a 2.0 format diskette?

Answer: <FCOPY 0 [DIR]>

Question: How do you use wild cards?

Answer: Wild cards are a means of defining a group of files. The TRSDOS-II commands which allow use of wild cards are DIR, FCOPY, SAVE, RESTORE, FILES, MOVE and KILL. For example, all 3 disk General Ledger files start with GL as the first two characters. To see a directory of just the General Ledger files, type <DIR GL*/*>. The asterisk means 'any sequence of characters here'. You could be even more specific by using <DIR GL*/DTA> which will give you a directory of just the data files.

Question: Explain what the IND option for the SAVE is.

Answer: The IND option allows you to specify a list of files to be saved. For example, say you have several Profile databases on hard disk along with some 3 drive accounting packages. Rather than saving the entire contents of the hard disk to floppy, it would be much faster and more convenient to save just the data files. The following example illustrates the use of the IND option: (User input inside < >)

Step 1. TRSDOS-II READY

<BUILD TRF> (Note:TRF is an example filename.)

ENTER COMMAND LINE (1-80)

<*/DTA>

Store Line? (cr/esc)

<ENTER>

*** LINE STORED IN FILE ***

ENTER COMMAND LINE (1-80)

<PRO1*/*> (Note: PRO1 is a Profile database name.)

Store Line? (cr/esc)

<ENTER>

*** LINE STORED IN FILE ***

ENTER COMMAND LINE (1-80)

<PRO2*/*> (Note: PRO2 is another Profile database name.)

Store Line? (cr/esc)

<ENTER>

*** LINE STORED IN FILE***

ENTER COMMAND LINE (1-80)

<ENTER>

*****EDITCOMPLETE*****

TRSDOS-II READY

<SAVE TRF:4 TO 0 [IND]>

This save command tells TRSDOS that TRF is not actually the file to be 'saved' but a file which contains a list of files to be 'saved'. (Notice the use of wild cards.)

Question: I have a Model II and a Daisy Wheel printer. Sometimes I need to type a quick envelope. How do I do that?

Answer: Place a blank sheet of paper in the printer. At TRSDOS READY, type <DUAL ON>. At TRSDOS READY, type <ECHO>. Now remove the sheet of paper, insert the envelope and start typing. When you are through, re-insert the paper and press <BREAK>. At TRSDOS READY, type <DUAL OFF>.

Question: How can I run a program without getting print-outs?

Answer: Use the command <FORMS D>. To return to the normal mode, use <FORMS N>.

Question: Using the HOST command, can I run Profile and Visicalc from remote locations?

Answer: No. These programs use 'cursor positioning' to write information at selected locations on the screen. Currently HOST is not capable of transmitting the codes necessary for that positioning to a remote terminal.

Question: What are the 'Supervisor Calls' used for?

Answer: They are used by Assembly Language programmers as programming aids to shorten development time. SVCs provide standard entry points to TRSDOS which tend not to change from version to version.

Question: What is the purpose of VERIFY?

Answer: Verify has 2 purposes on TRSDOS 2.0. VERIFY <ON/OFF>: (Defaults to ON) causes the computer to verify each write. We recommend that you leave this option on at all times.

VERIFY DETECT <ON/OFF>: (Defaults to ON) causes the computer to check for disk swap before each read or write. For applications like PROFILE, turning off verify detect will speed up the program tremendously.

On TRSDOS 4.1, VERIFY DETECT is no longer an option.

Question: How can I have an ASCII text file list to the screen in text format?

Answer: At TRSDOS READY, type:

```
<FORMS D>
<PRINT filename, V>
<FORMS N>
```



Budget Management and the LP VIII

Ruth Shackelford
Route 1, Box 261
Yale, OK 74085

Here are some line changes that I have made to Budget Management Program in order to have the program work with my Line Printer VIII.

Thank you for the many wonderful articles that you have been publishing in the Microcomputer Newsletter. I have found them to be of great assistance in my learning program.

This works with the Model III, 16K.

```
3110 T$=" NO. NAME CURRENT
      BUDGET PERCENT REMAINING"
3120 F$="##### % % #####.##
      #####.## ##.## #####.##-"
      : RETURN
3130 T$="DATE DESCRIPTION AMOUNT"
      : F$="##/## % % #####.##-"
3140 IF TY > 1 AND TY < 4 THEN T$=
      " NO. " + T$
      : F$="##### " + F$
      : ELSE T$=STRING$(10," ") + T$
      : F$=STRING$(10," ") + F$
3150 IF TY > 1 AND TY < 5 T$=T$+" EXP. ACCT"
      : F$=F$ + " #####"
3160 IF TY=3 THEN T$=" NO. WRN CAN
      DESCRIPTION AMOUNT EXP. ACCT"
      : F$="##### ##/## ##/## %
      #####.## #####"
      : RETURN ELSE RETURN
3530 GOSUB 3900
      : LPRINT TAB(25); "BUDGET MANAGEMENT REPORT"
      : STRING$(15," ");
      : LPRINT USING "PAGE ##"; PG
3540 LPRINT TAB(25) "*** HS" REPORT**"
      : LPRINT TAB(30)"DATE: ";
      : D=DT
      : GOSUB 500
      : LPRINT USING "##/##/##"; D1; D2; D3
3900 LPRINT STRING$(80,"*")
```

Computer Customer Services Address and Phone Number

8AM to 5PM Central Time
Computer Customer Services
400 Atrium, One Tandy Center
Fort Worth, Texas 76102

Model I/III Business Group (817)390-3939
Model II/16 Business Group (817)390-3935
Languages and Compilers Group (817)390-3946
Color and Pocket Computer Group (817)390-3944
Hardware and Communications Group ... (817)390-2140
Educational Software (817)390-3302
Games, Books, and News Product Info ... (817)390-2133
Newsletter Subscription Problems (817)870-0407

Megawars

Editor's Note: The CompuServe Information Service is one of the largest information and entertainment services available to owners of personal computers and computer terminals. With each issue of TRS-80 Microcomputer News, various features of CompuServe will be discussed. The CompuServe Information Service is sold at Radio Shack stores nationwide and in Canada.

NOW YOU CAN BE IN CHARGE OF YOUR OWN KRYON OR COLONIST SPACE ODYSSEY

By day you can be a mild-mannered newspaper writer, receptionist or company president, but by twilight you are ZELTAR, Kryon king. You are the dictator of the brave Colonists who have been enslaved under your reign for 700 years. You fight the Colonists and they fight you. As Kryon king you have immense power but you might lose that power if you are not careful.

Megawars is CompuServe's intergalactic battle game located on the games menu page. It's a noble game for one to 10 players at a time. Choose either to side with the Colonists and champion the Coalition or become a Kryon whose allegiance is to the Empire. The game can be played "interactively" by people throughout the country.

Whether Colonist or Kryon, both warriors' objective is to destroy all enemy bases and ships and capture all enemy planets before your enemy beats you to it.

As an intergalactic fighter you are in control of your destiny by commandeering your own spaceship from your computer keyboard. You are free to enter and leave battles whenever your objectives are complete. Of course you will have a rank which will be decided when you choose your battlefield.

New recruits enter the clash as Cadets who are assigned a fighter class ship with sufficient powers until they learn the ways of destroying the enemy.

Upon accumulating a battle score of 10,000 units, you will be eligible for promotion to Lieutenant where you will be given the opportunity to command a Scout or Fighter ship. A Scout ship is a reconnaissance model and a Fighter is a workhorse of sorts. Thirty thousand points promotes you to Captain and the option of using a Scout, Fighter or a Miner class spaceship which is a converted merchant ship used for retrieving Photon Mines. For those inherent leaders who score 100,000 points, the rank of Admiral becomes yours and you are then permitted to command any class of ship you desire including a Flagship which is a movable Starbase.

As you begin your intrepid journey in your Flagship, Fighter or Miner, you must always remember that your enemy and his ship are equal to you and yours and your survival depends on your skill as a commander. You must be aware of the locations of certain space inhabitants. An immobile starbase will surely blow you to bits if you stray too near, but is harmless if left to itself.

Enemy planets are weaker but still not to be ignored. You must be careful of neutral planets which will fire at you unless you win them over to your side. Another fearful enemy to watch is the dreaded Acheron. This devious enemy is concealed by a cloaking device which hides it from all wanderers and star fighters alike, and whose existence it seeks to terminate.



In your perilous voyage you are armed with various devices which keep you in flight. In case of attack, your crew is able to repair any damages while still in flight. You also are equipped with back-up engines, for when warp engines are down, as well as a deflector shield, life support system, sub-space radio, a tractor beam for the towing of fellow ships in trouble and a ship computer used for firing, computation of movement and phaser control.

You are now ready to travel out into the galaxy of battle. Go and be well and may the forces be with you.

SHOPPING AT HOME WITH DICK AND JANE

Dick and Jane are sitting in their living room when Jane turns to Dick and says:

More Quotes and a Whole World of News

Editor's Note: With more than 45,000 subscribers, Dow Jones News/Retrieval is the leading provider of on-line business and financial information. We will keep you up-to-date on new software, new data bases and new prices in upcoming issues of TRS-80 Microcomputer News. Dow Jones News/Retrieval is sold at Radio Shack stores nationwide and in Canada.

News/Retrieval users are benefitting from two more recent additions to their service—vastly expanded general news coverage and an enhanced historical quotes data base that will extend back an entire year.

THE WORLD REPORT

News/Retrieval now provides national and foreign news in a new service called the News/Retrieval World Report. The data base is compiled from the wires of United Press International, which for years has been recognized as a leader in the field of gathering and distributing news to newspapers, radio and television.

Throughout the day, the News/Retrieval editorial staff monitors and rewrites the news wire to produce a concise, coherent, continuously updated picture of the world and national scene in an easy-to-use electronic package designed specifically for busy subscribers.

Instead of putting all UPI material directly on-line, stories are selected and ranked according to importance by journalists in News/Retrieval's Princeton, N.J. newsroom.

Minutes after events are reported, they are included in the World Report. The editing process continues throughout the day, so that major stories appearing in the morning may be replaced by updated versions or by other late-breaking stories in the afternoon or evening. Typically, stories in the World Report run three pages, although breaking events of major importance may run longer.

The World Report is continuously updated from around 8 a.m. to about 8 p.m. (Eastern time), Monday through Friday. It is one of the General News and Information Services data bases, and costs standard subscribers 60 cents per minute in prime time or 30 cents per minute in non-prime times. It can be accessed by typing //NEWS.

After subscribers enter //NEWS to get into the World Report a copyright statement will appear at the top of the page, followed by headlines of the five most important national and foreign stories access codes. This is the World Report "front page".

Subscribers can press the N key to view a list of other national news headlines and the access codes to obtain the stories behind the headlines. Or, they can press the F key to view a list of additional foreign news headlines and story access codes.

In addition to the five "front page" stories, there are approximately 25 national stories and 25 foreign stories, with the most important headlines at the top and lesser ones farther down.

DAILY HISTORICAL QUOTES EXPANDED

The daily historical quotes, a data base, is being expanded to cover a full year of historical stock quotes, and quotes are available in composite form or by individual exchange.

News/Retrieval will enhance the Historical quotes data base in two other important ways:

- Composite historical quotes for monthly and quarterly requests
- Daily historical quotes accessible by specific date

To retrieve daily historical quotes for a full year use the same method that you currently do for daily quotes, e.g. for the last 12 days of Coca-Cola quotes type KO P1. For the preceding 12 days type KO P2. The new system extends back to P22. By typing KO P22, you would access price and volume data for days 253 through 264.

In the new system KO P1 will retrieve composite quotes for Coca Cola. To get historical quotes on an individual exchange, type the number of the exchange before the company code, e.g. 1KO P1. This will retrieve quotes from the New York Stock Exchange. Composite quotes will be retrieved automatically if an exchange number is not used.

To retrieve monthly and quarterly composites do not use an exchange, e.g. KO 82 M or KO 82 Q. To get quotes by exchange, use the same method as for daily historical quotes.

To retrieve historical quotes by date type the date, including month, day, and year, e.g. for June 30, 1982, type KO 6/30/82.



TO GET THAT QUOTE . . .

The following tables will help you determine which code should be entered to retrieve information on the trading days you desire, and which exchange code should be entered to get specific stock exchange information rather than the automatically retrieved composite information. For example, if you're interested in the trading days from 157 to 168 days ago, type P14 after the desired company code. And if you

want information from the Pacific exchange instead of composite quotes, type 3 before the desired company code.

Code	Accesses Trading Days	Code	Accesses Trading Days	Code	Exchange
P1	1-12	12	133-144	1	New York
2	13-24	13	145-156	2	American
3	25-36	14	157-168	3	Pacific
4	37-48	15	169-180	4	Midwest
5	49-60	16	181-192	.	Composite or OTC
6	61-72	17	193-204		
7	73-84	18	205-216		
8	85-96	19	217-228		
9	97-108	20	229-240		
10	109-120	21	241-252		
11	121-132	22	253-264		

punch the holes and tear off the right margin. This gives rugged binding edge and leaves all the printing intact.

Tape Mailing List and Two Across Labels

Randy Blinn
717 Ellwood-Zelienople Road
Ellwood City, PA 16117

I was using a Line Printer VII with single across labels but they did not work very well because of the thickness of labels. But when I went to the dry gum labels (26-1456) program would only print one across. I have Radio Shack Tape Mailing List 26-1503, so I did some changing and I came up with these changes. I thought others might be interested so I am sending it to you.

TAPE MAILING LIST

```

2015 INPUT"ARE THE LABELS TWO ACROSS < Y OR N >";
19000 K=1
19001 IF MID$(FI$(I,0),K,1)="" THEN IF K=1 THEN
RETURN
: ELSE LPRINT MID$(FI$(I,0),K+1);
" "; LEFT$(FI$(I,0),K-1);
: IF ZZ$="Y" THEN GOTO 19100 ELSE LPRINT
: GOTO 19005
19002 K=K+1
: IF K <= LEN(FI$(I,0)) THEN 19001 ELSE LPRINT
FI$(I,0);
: IF ZZ$="Y" THEN GOTO 19200 ELSE LPRINT
19005 IF E=1 THEN LPRINT FI$(I,1);
: IF ZZ$="Y" THEN LPRINT CHR$(16); "45";
FI$(I+1,1) ELSE LPRINT
19010 LPRINT FI$(I,2);
: IF ZZ$="Y" THEN GOTO 19250 ELSE LPRINT
: LPRINT FI$(I,3); " "; FI$(I,4); "-"; FI$(
: LPRINT CHR$(138)
: LPRINT CHR$(138)
19015 IF E=0 THEN LPRINT CHR$(138)
: ELSE IF FI$(I,1)="" THEN LPRINT CHR$(138)
19020 RETURN
19100 IF B$(0)="" THEN 19120
19110 IF B$(0)=LEFT$(FI$(I+1,S), LEN(B$(0))) THEN
19120
19120 K=K-1
19200 IF MID$(FI$(I+1,0),K,1)="" THEN IF K=1 THEN
RETURN
: ELSE LPRINT CHR$(16); "45"; MID$(FI$(I+1,0)
K+1); " "; LEFT$(FI$(I+1,0),K-1)
: GOTO 19005
19210 K=K+1
: IF K <= LEN(FI$(I+1,0)) THEN 19200 ELSE LP
CHR$(16); "45"; FI$(I+1,0)
: GOTO 19005
19250 LPRINT CHR$(16); "45"; FI$(I+1,2)
: LPRINT FI$(I,3); " "; FI$(I,4); "-"; FI$(
19300 LPRINT CHR$(16); "45"; FI$(I+1,3);
" "; FI$(I+1,4);
: I=I+1
: LPRINT FI$(I,5)
: LPRINT
: GOTO 19015
20000 DATA A, R, S, L, P, W

```

microRESEARCH Problem #4

Richard V. Andree
Department of Mathematics
University of Oklahoma at Norman

A cryptarithm is an arithmetic problem in which the digits have been replaced by letters. Each letter represents a distinct digit and no digit has more than one letter replacement.

The cryptarithm

$$I \cdot LOVE = TANDY$$

has 4 * 7039 = 28156 as one possible solution, but there are a dozen other solutions. Find them.

Possible extensions:

$$FIRE + BURNS = PAPER \text{ (2 solutions)}$$

$$FEAR \cdot FEAR = FASTSTAR$$

$$RUT \cdot RUT = CAREER$$

A SOLUTION FOR microRESEARCH PROBLEM #3

Here is a simple program to find the four three-digit whole numbers where the sum of the cubes of each of the three digits of the number is equal to the three digit number.

```

100 FOR N=100 TO 999
110 X=INT(N/100)
120 Y1=N-X*100
125 Y=(INT(Y1/10))
130 Z=Y1-(Y*10)
160 T=INT(X^3)+INT(Y^3)+INT(Z^3)
170 IF T=N THEN PRINT T
180 NEXT N

```

Binding Fan Fold Paper

Earl R. Kooi
2196 Albright Ave.
Upland, CA 91786

To bind program listings, etc., LLISTed on 9 1/2 by 11 inch fan-fold paper where punching holes after removal of the tear-off margins would punch through the left edge of the printing, place a strip of 3/4 inch transparent tape on the left side of the page over the pin holes and tear-off line. Then

PROFILE III + : Optimizing Storage Capacity

Profile
The small Computer Company

Written for the users of PROFILE II, PROFILE + and PROFILE III +

PROFILE Editor
P.O. Box 2910
Fort Worth, TX 76113-2910

By Ivan Sygoda, Director, Pentacle

For your PROFILE III + database to be as large, efficient and flexible as possible, it has to be made to "fit" well in a number of places—on the diskette(s) where it lives, so to speak, and on the screens, reports and labels where it interacts with the outside world. Optimizing usage takes a bit of juggling and maneuvering on the user's part, but the effort pays dividends in increased system capacity.

We can't do the juggling for you, because only you can decide what fields to include, their relative lengths, and where they are to be arrayed on various screens, reports, labels and form letters. However, we can help you with the decision process, showing you how to avoid inefficient solutions.

WORKSHEETS FOR DEVELOPING A DATABASE

We recommend that you develop your entire system on paper first, rather than on the computer. Although you can't set up screens, reports and labels without a working set of fields, you may regret decisions concerning field lengths unless you anticipate how the fields will fit on these screens, reports, and labels. You should plot out likely formats on Radio Shack's video worksheets (cat. no. 26-2105). Reports and labels require special care because their size and layout is so confined.

LABELS AND REPORTS

For instance, the second line of a label format might include both "Title" and "Department." Suppose you had allotted 12 characters for the title and 30 characters for the department name: 12 + 30 + 1 space between the two fields = 43 characters. But typical address labels are less than four inches wide—at 10 characters per inch (c.p.i.) the line might require 4.3 inches. A miss is as good as a mile.

Shave three characters off the department field, or two off department and one off title, or any feasible combination. What you're doing is trading off an occasional awkward abbreviation against misprinted labels. The time to make such adjustments is before you define fields.

Suppose you're setting up a mailing list file and know that report printing will be an important function. Say that you had defined your fields as follows: Last Name = 16, First Name = 12, Address = 24, City = 20, State = 2, Zip = 10. You'd also need spaces between fields, a total of five in this case. The grand total is 89 characters, which makes it impossible to

print one-line reports on your 80-character-wide printer. Solution: Shave one character off the last name field, two off the address field, one off the city field, and reduce zip code to five (the nine-digit code seems mainly to be for business-reply mail . . . so far). Most of your data will still fit, and you can do efficient one-line reports. Don't forget that PROFILE III + lets you include the record number (indicated by @; leave five spaces) on reports and labels. Since accessing records for updating by record number is so efficient and easy, you might consider including this field in your formats.

Work out report formats on the coding side of the video worksheets. Counting the space separating the variables list, there are 80 characters across the sheet, perfect for planning 80-character-wide reports. You will often discover that a little judicious shaving of field lengths will increase the number of fields that will fit in a report line or make it possible to group fields more logically. The PROFILE III + manual discusses file definition and the use of data fields.

ALLOCATING SEGMENTS TO DRIVES

When you've finally determined your fields and their lengths, and decided which are key fields, your next task is to array these fields across the four segments available. The variables are: 1) the number of disk drives you have or plan to have; 2) the disk space available on each of these drives. Again, you have to juggle things a bit. There's a spot of math involved, but it's easily done with a calculator.

The TRSDOS directory utility will tell you the amount of free space in granules (1 gran = 768 bytes or characters) remaining on any particular diskette. To see a directory, at TRSDOS type: DIR :n ("n" is the drive number) and press <ENTER>. Remember, however, that your data files aren't alone in taking up space on the diskettes. You also have to decide which diskettes will hold the screen, label and report formats, user menus, math formulas, et cetera. Since each of these formats takes up disk space, it will be useful to predict how much.

HOW MUCH SPACE ON A DISKETTE?

A 5¹/₄ inch data diskette with 40 tracks, 18 sectors per track and 256 bytes per sector (which is the same as 40 tracks at 6 grans each) can contain 184,320, or 180K, bytes

(180 x 1024) of information. The directory occupies track 17 (6 grans, or 4608 bytes).

A system diskette's operating system occupies an additional 21 sectors or seven grans. Also taking up space on a system diskette are: system information (not shown on the directory) at 49 grans, BASIC at 7 grans, CONVERT at 4 grans, and XFERSYS at 2 grans, for a total of 62 grans. The Runtime diskette programs delivered with the PROFILE III+ system (EFC7, EFC8, EFC9, EFCA, EFCB, EFCC, EFCD and the Runtime menu M) take up another 76 grans.

So a data diskette will have approximately 171K free space for user data, and a system diskette—your Runtime drive 0 diskette—will have only 95K free space. Much of this 95K could easily be used up by the format files.

SPACE USAGE

User-generated files or programs are of two sorts: 1) formats (screen, report, label, SCRIPSIT and VisiCalc selections, plus user menus, math formulas and BUILD files) whose lengths are relatively constant and independent of the number of records currently in the data base; and 2) the actual data segments and the index and merge files, all of which grow in size as the user's database grows.

You can anticipate that formats will occupy space as follows—the grand totals indicate the amounts of space you'd use if you defined all available formats:

Screen formats	: [yourfilename]/PM1-/PM5 @ 2 grans each = 10 grans total
Report formats	: [filename]/PR1-/PR5 @ 3 grans each = 15 grans total
Label formats	: [filename]/LB1-/LB5 @ 3 grans each = 15 grans total
SCRIPSIT formats	: [filename]/SL1-/SL5 @ 3 grans each = 15 grans total
VisiCalc formats	: [filename]/VC1-/VC5 @ 3 grans each = 15 grans total
User menus	: each menu occupies 6 grans
BUILD files	: used with menus; occupy 1 gran each
Math formulas	: [filename]/MTH occupies 2 grans

Data, index and merge files, as mentioned above, grow in size as you expand files. The length in grans of any data file may be computed as follows:

$$(LRL \times \#REC) / 768 = \#GRANS \text{ (round to next higher integer)}$$

LRL = Logical Record Length
#REC = number of records in the file
768 = number of bytes in one gran.

All these factors can be taken from the directory listing.

It is just as easy to solve for the number of records which will fill a known quantity of grans:

$$\#REC = (\#GRANS \times 768) / LRL$$

These two equations give you the information you need to calculate the space proposed data segments will occupy.

The LRL of an index file (filename/IX1) is the length of the sort field plus 2 characters. The LRL of a SCRIPSIT merge file (filename/SR1-/SR5) is a single character; the number of records, on the other hand, is highly variable, and the space necessary must be evaluated on a case-by-case basis.

Note: "Bytes" in PROFILE III+ are equivalent to "characters." While you're defining your fields on paper, you can get a rough idea of the number of bytes per segment by totalling the numbers of characters being used for the fields.

The approximate Logical Record Length is the number of characters being used for all fields in a segment.

DISTRIBUTING THE DATA

As you can see, at their theoretical limit of complexity PROFILE III+ Runtime programs and formats can fill the system diskette. However, it is not necessary to group these files together on the same diskette.

Fact: Any PROFILE III+ Runtime program or any user-generated format program or data segment can be moved at any time to any disk drive which has room for it. PROFILE III+ will find the program when it needs it by scanning the directories. This gives you a lot of flexibility in arranging disk usage for maximum storage capacity.

You have, therefore, two basic ways to go. One, you could array your data segments across the available drives as per the principles explained in the PROFILE III+ user manual. Any screen, label, etc. format could be fit in around these segments like mortar around uneven bricks. Or, two, you could reserve the system (drive 0) diskette for Runtime and format programs and relegate all data segments to drives 1 to 3.

This latter method has certain advantages. When your data base outgrows the capacity of your system, you can easily add a disk drive or, less expensive, divide your data base by some relevant criterion: geography—by zip, state or area code; alphabetically—by name, A-M and N-Z. Then you can switch data disk sets, using the same Runtime diskette for each data set—you'll have one less diskette to back up at the end of each session!

A GOOD SENSE OF PROPORTION

If you do elect to reserve the drive 0 disk for Runtime programs and formats, the chart in Figure 1 will help map out the permutations involved in arraying various numbers of segments across various numbers of disk drives. There's a problem distributing segments when the number of segments and the number of available drives are evenly divisible by one another. It gets tricky, however, when they are not. Especially tricky is arraying three data segments across three drives.

The problem is that when one segment fills its drive, the entire system is full as far as the computer is concerned. Segments are not divisible—you can't put the first half of segment 1 on one drive, and the second half on a second drive. You also want to be able to incorporate another drive efficiently when the budget allows.

If you consider your entire file equal to 1, and establish relative lengths of $1/2$, $1/4$ and $1/4$ for three data segments while using two drives (grouping the two smaller segments and reserving drive 0 for programs and formats), both drives will fill up at the same rate. However, when you add a third drive, you cannot increase the system's capacity by moving one of the two smaller segments onto the new drive—the drive containing the larger segment will fill its disk at the same rate as before. The answer is to make four segments with ratios of $1/3$, $1/3$, $1/6$ and $1/6$. While using two drives, bundle them as follows: drive 1— $1/3$ and $1/6$, drive 2— $1/3$ and $1/6$. After adding a third drive, move the segments this way: drive 1— $1/3$, drive 2— $1/3$, drive 3— $1/6$ and $1/6$.

EXTRA HINTS

A few nuts and bolts suggestions to make life easier:
1) When you format your data diskettes, name them in su

way as to identify the database and drive. For example, SALES 1.2 identifies the disk as your SALES program, region 1 territory, drive 2 diskette. Print the same name on the label of the diskette.

2) Set up your entire system on one disk, expanding your files by only five or ten records to test the formats. That is, keep your Creation diskette in drive 0 and your Runtime disk in drive 1, and always answer "1" when prompted for "Drive number to hold this segment." Then, when everything is working to your satisfaction, transfer the various segments and formats to the drives you've determined they should inhabit and expand your files to working levels. This will save extra steps and disk-switching during file creation procedures.

PROFILE Editor's Note: This is Mr. Sygoda's second article in a series of 'how-to' PROFILE III+ articles. Other articles in the series will be published over the next few issues in this column. We hope that you enjoy this new feature, and we look forward to your comments and questions on PROFILE III+.

Pentacle is a New York City-based non-profit service organization specializing in administrative services for performing art groups.

Number of Segments In Your Data Base	Number of Drives Available For Data Assuming Drive 0 Is Reserved For Runtime Programs & Formats		
	1 Drive	2 Drives	3 Drives
1 segment	filename/KEY, /KX1, /KX2, /KX3 will fill additional disk drives as needed (true only of single-segment data bases)		
2 segments	no problem	keep segments equal	not possible — segments cannot be divided
3 segments	no problem	use proportions of thirds & sixths	keep segments equal
4 segments	no problem	keep segments equal	use proportions of thirds & sixths

Figure 1

ACCESSING PROFILE DATA FROM AN INDEX, PART 2

In the last issue of this newsletter, we published a program that lets you access PROFILE data sequentially from a PROFILE index. This issue, we've written a program that lets you access PROFILE data randomly (in other words, in a way other than in record number order), again using an index.

The computer does "binary" searches of the index to find a particular piece of data. A binary search is one in which the computer cuts the indexed list in half and in half again as many times as is necessary to pinpoint the desired piece of information. The charm of this demonstration program is that you can watch the process on the screen.

You can also use the program as a module, building a BASIC program around it. Please note that the program shown is a demonstration program only — you should adapt it for your own purposes.

A few notes are in order. This demonstration program accesses data only from the key segment, which is the most difficult to access because it must be unblocked. If you want to access non-key data, include statements between lines 800 and 900 reading as follows:

```
OPEN "D", 3, FS + "/DAT," nnn
```

where "nnn" is the length of the segment and "/DAT" is the

second data segment; DA2 and DA3 are the other data segments.

```
FIELD 3, n AS XX$, m AS YY$ . . .
```

where "n", "m", etc. are the lengths of your PROFILE fields, and "XX\$", "YY\$" etc. are field names.

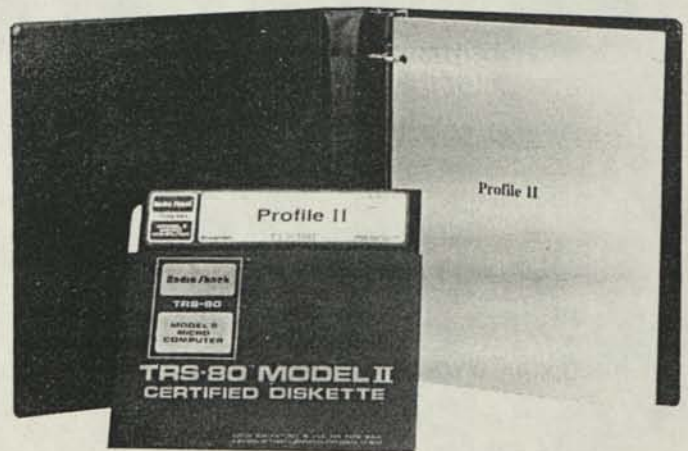
If you want an entire record, you can then access it by typing (between lines 3600 and 3700; make sure you have a third file):

```
GET 3, LR
```

where "LR" is the logical record.

To open: BASIC FIND1/BAS-F:2 (two files are needed, one for the key and one for the index)

```
10 ' FIND1/BAS
20 ' The small Computer Company, Inc.
30 '
40 ' Index demo program
50 ' Random access of PROFILE+ records from BASIC using a PROFILE+ index
60 '
70 '
100 CLS : CLEAR 5000 : DEFINT A-Z
200 INPUT "ENTER FILE NAME"; FS
300 FS = LEFT$(FS+"00000000",8) ' Pad with zeros to make 8-byte file name
400 INPUT "ENTER KEY LENGTH"; KY
500 OPEN "D", 1, FS + "/IX1", KY+2 ' Open index file
600 FIELD 1, (KY) AS KY$, 2 AS LR$ ' Key field and two-byte pointer
700 OPEN "D", 2, FS + "/KEY" ' Open /KEY segment
800 '
900 '
1000 PRINT : INPUT "ENTER KEY"; K$
1050 PRINT
1100 '
1200 PT = 0 ' PT will point to current position in index
1300 HI = LOF(1)+1 ' Set HI to last index entry plus one
1400 LO = 4 ' Set LO to first index entry
1500 '
1600 OP = PT ' Save old pointer
1700 PT = INT((LO+HI)/2) ' Set pointer to middle of unexamined portion
1800 IF OP = PT THEN 4400 ' Search completed; no match has been found
1900 GET 1, PT ' Read index record
1950 PRINTUSING "###: "; PT; ' Display index record number being examined,
2000 PRINT " "; KY$; " "; ' key value, and
2050 PRINTUSING "###: "; CVI(LR$) ' PROFILE+ record number
2100 GOSUB 4800 ' Set index key length to length of search key
2200 IF K$ = RK$ THEN 2800 ' Match found; display PROFILE+ record
2300 IF K$ > RK$ THEN LO = PT ' Index pointer was still too low
2400 IF K$ < RK$ THEN HI = PT ' Index pointer was still too high
2500 GOTO 1600 ' Loop and try again
2600 '
2700 '
2800 GET 1, PT-1 : GOSUB 4800 ' Now look backwards through index
2900 IF K$ <> RK$ THEN 3100 ' Did we overshoot the first match?
3000 PT = PT-1 : GOTO 2800 ' If not, keep looking backwards
3100 GET 1, PT ' This is the first match
3200 LR = CVI(LR$) ' PROFILE+ record number from index pointer
3300 PR = INT((LR-1)/3) ' Deblock the KEY segment record:
3400 SR = LR - PR*3 - 1 ' SR = sub-record number (0, 1 or 2)
3500 PR = PR+1 ' PR = physical record number
3600 GET 2, PR ' Get the /KEY record
3700 FIELD 2, (85*SR) AS DDS, 85 AS AS ' AS = /KEY segment data fields
3800 PRINT : PRINT LR; " "; AS ' Print record number and data
3900 PT = PT+1 ' Advance pointer to next index record
3920 IF PT > LOF(1) THEN 1000 ' Past end of file?
3950 GET 1, PT : GOSUB 4800 ' Get it and adjust key field for compare
4000 IF K$ = RK$ THEN 3200 ' Loop back if it's also a match
4100 GOTO 1000 ' Otherwise, ask for next key
4200 '
4300 '
4400 PRINT "NO MATCH FOUND"
4500 GOTO 1000
4600 '
4700 '
4800 RK$ = LEFT$(KY$,LEN(K$)) ' Make index key equal length to K$
4900 RETURN
```



Introducing SATN, The Journal for VisiCalc Users

Reprinted by permission of Software Arts, Inc.

These articles originally appeared in SATN, Vol. 1, No. 1, Copyright ©, on September/October, 1981, Software Arts, Inc., P.O. Box 494, Cambridge, MA 02139.

Editor's Note: From time to time we will be publishing articles from the SATN journal in hopes that they may aid our readers and users of VisiCalc® to better understand and utilize the many features of this powerful software package.

The user of any program deserves strong technical and professional support. When a program is as widely applicable as the VisiCalc® program is, its users benefit from the support already provided by a variety of sources: the manual, the reference card, the distributor, computer dealers, and users' clubs. The SATN pronounced "satin"—journal is designed to supplement, not supplant those sources of support.

The VisiCalc program, a visible calculator program created by Software Arts, and published and distributed exclusively by VisiCorp under their registered trademark VisiCalc, is used by many people in many ways. Some users are novices, others experts; some use it casually, others intensively.

The purpose of SATN, a publication from Software Arts, is to show all VisiCalc users how to use the program more effectively and extensively. The articles in SATN will focus as much as possible on information that is not specific to any particular computer in order to be helpful to all users of the VisiCalc program. SATN's editorial aim is to provide VisiCalc users with informative, in-depth articles about the program drawn from experts both within Software Arts and from outside the company.

This and subsequent issues will include practical applications of the program in sufficient detail to allow you to solve similar problems immediately. To expand your scope of applications, we will also include tutorials and techniques useful in solving a wide variety of problems. We will pass on some pointers about the VisiCalc program to clarify misunderstandings, and describe brief accounts of why VisiCalc features were designed as they were, to provide insight into better ways to use the program.

A CLOSE LOOK AT @LOOKUP

The @LOOKUP function of the VisiCalc program can be an invaluable tool in a number of applications. It is particularly helpful in tax calculations, and in fact was designed with tax tables in mind. But you can use @LOOKUP with tables that you create yourself for your own particular applications.

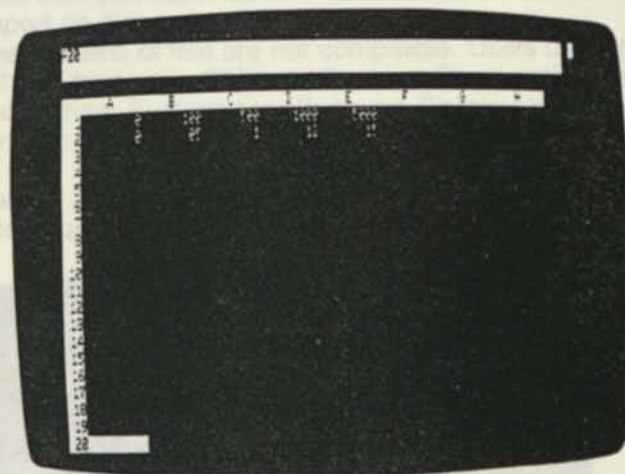
As we will see @LOOKUP can be used to search an inventory table, or a table of discount percentages; to help

calculate the costs of items in which a number of variables come into play; or to perform some payroll calculations involving taxes. Among the other uses of @LOOKUP that come to mind are searches of insurance premium and interest rate tables.

As entered on the sheet, an @LOOKUP table consists of a range of values to be searched and their corresponding values. The corresponding values must be entered on the sheet in the row directly below or the column directly to the right of the range of values.

In operation, the @LOOKUP function takes a number that you have entered, the input value, and sequentially searches the range of the table until it reaches the first number in the range that is greater than the input value. It then returns to the preceding number and selects the corresponding value. The @LOOKUP function assumes that the range is in ascending order, though the corresponding values need not be.

Our first example shows how @LOOKUP is used with a small table of discount percentage. Screen 1 contains the sheet for such a table.

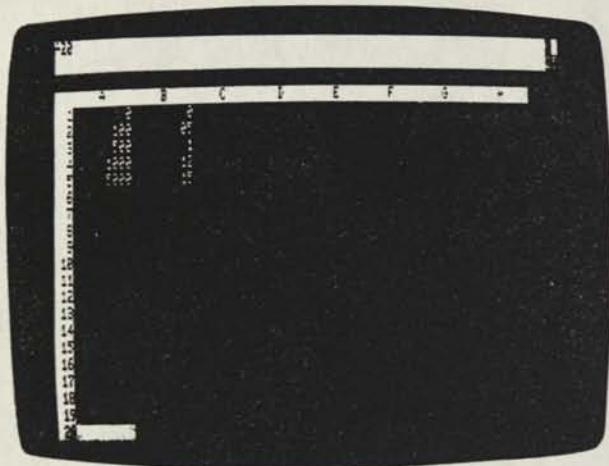


Now suppose you need, at C5, the discount applicable to a purchase of 1000 items. To get the appropriate percentage in that location on the sheet you would enter:

```
>C5:@LOOKUP(1000,A1..E1)
```


@LOOKUP will find the value in the specified range A1 . . . E1, and will return .12, the corresponding value in the second row.

If the sheet is set up in columns, as in Screen 2, the @LOOKUP search is the same. The range, however, is A1 . . . A5 and the corresponding value is found in the column immediately to the right of the input value.



If you wish to find the discount applied to 800 items, @LOOKUP will search the table, matching 800 to each value in the range consecutively. When it reaches 1000, it will stop, back up to the preceding value, and return .10, the correct discount to apply to a purchase of 800 items.

If you use @LOOKUP to find a value that is lower than the first one in the range, the result will be NA (Not Available). If you try to find a discount, for example, for - 50 items in the tables in Screens 1 and 2, NA would be returned. If you use @LOOKUP to find a value that is greater than the last value at the end of the table, it goes back to the last value it found. Thus 6000 would return to .15.

Once you have entered an @LOOKUP table on your VisiCalc sheet, you can save it and use it repeatedly, either to retrieve a value that is an end in itself or one that will be used for further calculations. You may of course, enter that table at any location on the sheet and retrieve values by giving the coordinates for the range or by pointing at them with the cursor. 

SATN

THE JOURNAL FOR VISICALC USERS

Business Checkwriter

Business Checkwriter (Cat. No. 26-1585) has been introduced by Radio Shack to provide another approach to the previously released Checkwriter 80. Those customers who felt that Checkwriter 80 did not fully meet their needs, may find Business Checkwriter more to their satisfaction. Additionally, Radio Shack will provide Business Checkwriter to any present owners of Checkwriter 80 at a minimal fee.

The differences between the two packages should be fully understood before a purchase or a switch is made. For instance, a current bank balance, distribution of check to multiple expense codes, detail check stubs, and payee summaries are all available on the new Business Checkwriter, where they were non-existent on the Checkwriter 80 package.

Checkwriter 80 has year-to-date summaries which are available on check register and end of year report. Business Checkwriter does not offer this particular facility.

On current expense summaries, the Business Checkwriter user determines the period for which he desires a summary. Current detail is cleared when an expense summary is printed. Capacity limitations suggest that one month is the maximum length of time. Checkwriter 80 users see current period total printed on the check register.

Business Checkwriter maintains only one account directly, but the user has the option of setting up multiple systems. Checkwriter 80 handles up to a maximum of nine accounts.

Business Checkwriter sustains up to nine daily deposits. Checkwriter 80 is limited to one deposit per day.

The entry of manual checks is recorded and detail printed with Business Checkwriter. It is simply recorded with Checkwriter 80.

With both packages bank reconciliations are available with report on demand.

The two sets of files are not compatible. Users cannot move their Checkwriter 80 files to Business Checkwriter files and vice versa. Furthermore, Business Checkwriter uses different check forms than Checkwriter 80.

A note to the buyer/user: Please be sure to provide verification, such as a sales receipt, for Checkwriter 80 in order to acquire Business Checkwriter at the minimal fee.



Have We Got Printers for You!

While writing about the new Radio Shack printers, it was hard to keep the superlatives that kept flowing from my keyboard to a minimum, but a concerted effort was made to simply state the facts so that you could see for yourselves the abundance of features to be found in the new printer line.

DMP, DWP, CGP?

In reading about the new printer line, you may notice right off that when naming our latest and greatest in printers we have gone in for acronyms in a big way. DMP stands for Dot Matrix Printer, and DWP? . . . Right, Daisy Wheel Printer. You may even have noticed in October that CGP stood for Color Graphics Printers. We have a lot of clever things like that up our sleeves.

So much for chit chat, on to the facts.

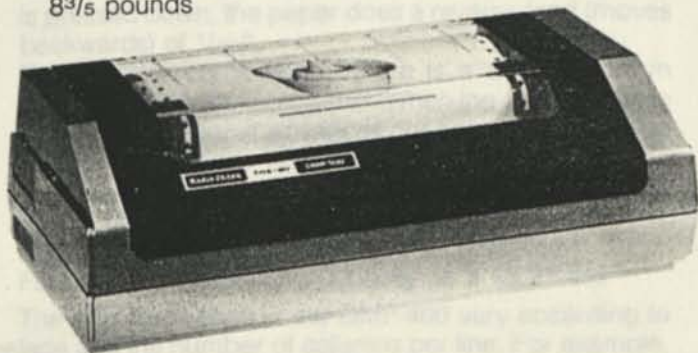
DMP-100

The DMP-100 has all the capabilities of the Line Printer VII along with the three important differences below.

- It's faster. For instance, the DMP-100 prints 50 characters per second on a ten character per inch (cpi) line at 27 lines per minute.
- It has underlining capability.
- It has a 480 byte full-line dot buffer for much faster graphic printing. Compare this to the 90 byte buffer of the Line Printer VII.

Other features of the DMP-100 are:

- A bit image mode with 480 addressable 7 dot columns for printing high-density graphics—up to 3780 dots per square inch 80 upper/lower case 5 x 7 matrix characters (a total of 96 ASCII characters) on an 8" line that can be printed at either 5 or 10 cpi
- Both parallel and serial interfaces to run with the Color Computer, Model I, II, III, or 16
- A ribbon cassette and an adjustable 4½ x 9½" tractor which can use up to 9½" fanfold paper
- A compact 5⁵/₁₆ x 16 x 8¹/₄" printer that weighs only 8³/₅ pounds



The many features of the DMP-100 (catalog no. 26-1253, \$399.00) make it a very useful printer at a very attractive price.

THREE MORE DOT MATRIX PRINTERS

The DMP-200, DMP-400, and DMP-500 share many outstanding features. They all have built-in graphics capability, bi-directional logic-seeking 9-wire print heads, and the ability to print lower-case descenders and true underlines.

More features of these three printers are:

- A Word Processing mode to emulate the Daisy Wheel printers which includes:
 - Proportionally spaced and mono-spaced (Standard at 10 CPI, Elite at 12 CPI, and Correspondence at 10 CPI character sets as well as 16.7 CPI Condensed.
- Backspacing, boldface, underlined characters, subscripts, and superscripts
- A Bit Image dot-addressable mode for graphics and special repeat and column addressing codes to simplify graphic programming
- A Bold mode to allow a wide range of character styles and a character elongation mode for printing wider characters which like the bold mode can be used in all fonts
- A Data Processing Mode for faster throughput

The DMP-200, DMP-400, and DMP-500 also include switches to allow manual selection of modes, pitches and interface options as well as the ability to select these options through software.

Because all three printers have a friction platen, they can use 8" single sheet paper (The 15" printers can use wider paper.) in addition to standard fanfold paper on the adjustable tractor. The general specifications for these three printers are:

- Character Set
 - 96 ASCII, 32 special and 30 block graphics character
- Line Feed
 - Forward, reverse; 1/6, 1/12, 1/72, 1/216
- Print Density
 - 10, 12, or 16.7 characters per inch in either a 9 x 9 or 15 x 9 dot matrix, n-23 matrix in a proportionally spaced font—all fonts can be elongated
- Resolution
 - 60 dots per inch at 10 cpi
 - 72 dots per inch at 12 cpi
 - 100 dots per inch at 16.7 cpi
- Buffer
 - 2K RAM

For a more detailed comparison of these three printers plus the DMP-100 and the predecessors of these three printers, refer to the Radio Shack Dot Matrix Printer Comparison Chart included in this article.

Now for a look at the features that make the DMP-200, DMP-400, and DMP-500 unique.

DMP-200

The DMP-200 is about the size of the Line Printer VIII but there are some noticeable differences. It has a paper guide

which functions similarly to a typewriter paper guide. When in the up position, the paper guide holds the paper up as it comes through the printer so the page can be seen as it is printed. When the paper guide is in the down position, it works as a paper separator for multipart, fanfold paper. Other new features of the DMP-200 include:

A removable, adjustable tractor instead of the non-adjustable pin feed of the Line Printer VIII

Bi-directional carriage to give printing speed of 55 full lines per minute and 120 characters per second at 10 characters per inch

Selectable parallel and serial interfaces (600/1200 baud)



The DMP-200 includes line feed increments of $1/6$, $1/8$, $1/72$, and $1/216$ and on/off line, restart, and paper feed controls. The 8" line lets you print 80 columns of characters at 10 pitch or 143 characters at 16.7 pitch. The DMP-200 (catalog #26-1254, \$799.00) is a compact— $5 \times 6\frac{1}{2} \times 13\frac{1}{2}$ " , 16 lbs.—but very versatile printer.

DMP-400

This terrific 15" wide printer offers the same size efficient styling found in the Line Printer VI. But just look at some of the additional features of the DMP-400.

Cover design which aids in noise reduction

Line feeds in $1/6$, $1/8$, $1/72$, $1/216$ increments

Selectable parallel and serial interfaces

Paper movement switch which when in the up position advances the paper $1/72$ ". If the switch is held up continuously, the paper continues to advance. When the switch is pressed down, the paper does a reverse feed (moves backwards) of $1/72$ ".

For operator convenience there is a LineFeed/Form Feed switch on the front panel. When the switch is set to Line Feed, the paper advances one line. If held in the up position, the paper will feed continuously. When the switch is set to Form Feed, the paper will advance to the logical top of the next form.

Software controllable Form Feed for graphics.

Full 132 column printing capacity on a 13.2" line

The printing speeds of the DMP-400 vary according to typeface and the number of columns per line. For example, with a standard type face and a 20 column line the DMP-400 prints 181 lines per minute (lpm) while when using the same type face to print 132 columns across the page, it prints at a



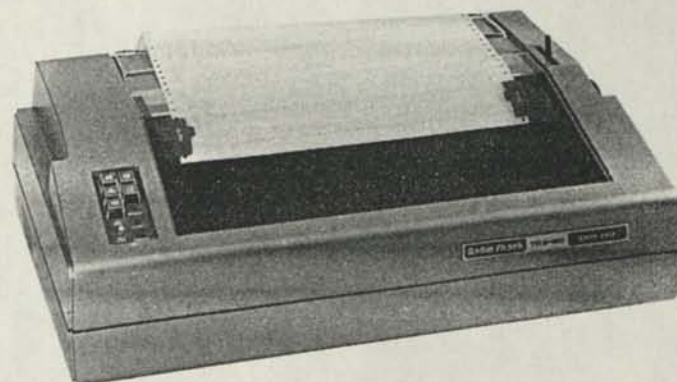
rate of 51 lpm. Using the correspondence type face at 10 cpi, it prints at a rate of 705 words per minute (wpm).

The DMP-400 (catalog no. 26-1251, \$1195.00) includes an adjustable, removable tractor and don't forget the list of features that apply to this dot matrix printer as well as the DMP-200 and DMP-500.

DMP-500

The DMP-500 is a heavy duty, high performance, dot matrix printer which lends itself favorably to a business environment requiring large volumes of through-put. One of the most outstanding features of the DMP-500 is its speed. This high speed printer prints 220 characters per second and seventy-six 132-column lines per minute.

Since it can line feed in $1/6$, $1/8$, $1/72$, $1/216$ increments, the paper can advance in fine-line increments for aligning pre-printed forms.



The DMP-500 offers:

A standard parallel interface so it can be used with the Models I, II, III, or 16

Paper feed and form feed controls plus an on-line indicator

The DMP-500 (catalog no. 26-1252, \$1795.00) is a great value in a powerful printer.

DOT MATRIX PRINTER COMPARISON CHART

The printer comparison chart gives you an indication of the family history of the new dot matrix printers. The old printers are those that have been replaced in the Radio Shack printer line by the corresponding new printers, e.g. the Line Printer VII has been replaced by the DMP-100. Many similarities between the old and the new exist in appearance, capability, and price, but there are, as we have seen, noteworthy differences.

Radio Shack Dot Matrix Printer Comparison Chart

	Old LP VII 26-1167	New DMP-100 26-1253	Old LP VIII 26-1168	New DMP-200 26-1254
Matrix Format				
Size in dots (H x W)	7 x 5	7 x 5	9 x 7 = 23	9 x 7 = 23
Descenders	No	No	Yes	Yes
Underline Mode	No	Yes	Yes	Yes
Bold Mode	No	No	No	Yes
Max. Line Length	8"	8"	8"	8"
Horizontal Pitches (cpi)				
Fixed	5, 10	5, 10	5, 10, 16, 6	10, 12, 16, 7
Proportional	No	No	Yes	Yes
Elongate any pitch	NA	Yes	NA	Yes
Mixed pitch per line	Yes	Yes	Yes	Yes
Line Pitches (LF per inch)	6	6(9)	6, 8, 12	6, 8, 12, 72, 216
Speed (Throughput)				
CPS @ 10 cpi	30	50	80	120
LPM @ 10 cpi	19/80 col 53/20 col	26/80 col 47/40 col 78/20 col	23/80 col 55/20 col	55/80 col 80/40 col 104/20 col
Logic Seeking (1)	Short Line	Short Line	Yes	Yes
Bi-directional	No	No	No	Yes
Character Set				
Standard ASCII (7 bit)	Yes	Yes	Yes	Yes
International (8 bit)	No	No	Yes	Yes
Graphics				
Block Characters	No	No	Yes	Yes
Bit Image: Dots per line	480	480	960 (3)	480-800
Paper Handling				
Maximum Size	9 1/2"	9 1/2"	9 1/2"	9 1/2"
Friction Feed	No	No	Yes 8.5" (s,r)	Yes 8.5" (s,r)
s—Single Sheet				
r—Roll Paper				
Tractor (Fan fold)	Adj (4.5"-9.5")	Adj (4.5"-9.5")	Fixed 9.5"	Adj. 10"
Form Feed	NA	No	NA	Yes
Copies	Orig. + 2 (2)	Orig.	Orig. + 2	Orig. + 2
Word Processing Features				
Correspondence Fonts				
10 cpi	NA	No	NA	Yes
PS	No	No	Yes	Yes
Incremental Spaces				
Backspace	No	No	Yes	Yes
For/Rev Linefeed	No	No	Yes	Yes
For/Rev 1/2 Linefeed	No	No	Yes	Yes
Controls	Power	Power	Power On/Off Line Reset Restart	Power On-line/Off-line Reset Paper feed form feed
Indicators	1	1	2	2
User Self Test	No	No	Yes	Yes
Reliability				
Head Life (Chars)(2)	30 Million	30 Million	>30 Million	30 Million
MTBF@100%	300 Hrs (50% Duty)	>300 Hrs (50% Duty)	300 Hrs	>300 Hrs
Ribbon				
Type	Re-inking cass	Re-inking cass	Cartridge	Refillable Cartridge
Life	1 Mil chr.	1 Mil chr	1 Mil chr	4 Mil Char
Catalog Number	26-1424	26-1424	26-1418	26-1483 cart 26-1489 refill
Interface				
Buffer	90 bytes	480 bytes	160 bytes	2K bytes
Parallel Data Bits	8	8	8	8
Status: Busy	Yes	Yes	Yes	Yes
Bits: Fault	No	No	Yes	Yes
: Paperout	No	No	Yes	Yes
: Busy	No	No	Yes	Yes
: Select	No	NA	Yes	NA
Serial (3)				
Data Bits	7, 8	7, 8	8	8
Baud Rate	600	600/1200	600/1200	600/1200
Size	5.32 x 16 x 8.25"	5.32 x 16 x 8.25"	4.7 x 15.4 x 11"	5 x 16.5 x 13.5"
Weight	8.6 lbs.	8.6 lbs.	16.5 lbs.	16.5 lbs.
Power 120 VAC ± 15%	15W	15W	85W	85W
Price (less cable)	NA	\$399.00	NA	\$799.00

NOTE: Specifications are based on the most accurate information available at the time of publication, subject to change.

	Old LP VI 26-1166	New DMP-400 26-1251	Old LP V 26-1165	New DMP-500 26-1252
Matrix Format				
Size in dots (H x W)	7 x 9	9 x 9 = 23	9 x 9	9 x 9 = 23
Descenders	No	Yes	Yes	Yes
Underline Mode	No	Yes	Yes	Yes
Bold Mode	No	Yes	Yes	Yes
Max. Line Length	13.2"	13.2"	13.2"	13.2"
Horizontal Pitches (cpi)				
Fixed	5,7,5,10,15	10,12,16,7	5,7,5,10,15	10,12,16,7
Proportional	No	Yes	No	Yes
Elongate any pitch	NA	Yes	NA	Yes
Mixed pitch per line	Limited	Yes	Limited	Yes
Line Pitches (LF per ")	6,8,12	6,8,12,72,216	6,8,12	6,8,12,72,216
Speed (Throughput)				
CPS@10 cpi	100	140	160	220
LMP@10 cpi	33/132 col 42/80 col 120/20 col	51/132 col 70/80 col 126/40 col 181/20 col	60/132 col 100/66 col 198/22 col	76/132 col 119/80 col 200/40 col 293/20 col
Logic Seeking (1)	Yes	Yes	Yes	Yes
Bi-directional	Yes	Yes	Yes	Yes
Character Set				
Standard ASCII (7 bit)	Yes	Yes	Yes	Yes
International (8 bit)	Yes	Yes	Yes	Yes
Graphics				
Block Characters	Yes	Yes	Yes	Yes
Bit Image: Dots per line	No	792-1320	No	792-1320
Paper Handling				
Maximum Size	15"	15"	15"	15"
Friction Feed	Yes (s)	Yes (s)	No	Yes (s)
s—Single Sheet				
r—Roll Paper				
Tractor (Fan Fold)	Adj. (4"-15")	Adj. (4"-15")	Adj. (4"-15")	Adj. (4"-15")
Form Feed	NA	Yes	NA	Yes
Copies	Orig. +3	Orig. +3	Orig. +4	Orig. +4
Word Processing Features				
Correspondence Fonts				
10 cpi	NA	Yes	NA	Yes
PS	No	Yes	No	Yes
Incremental Spaces				
Backspace	No	Yes	No	Yes
For/Rev Linefeed	No	Yes	No	Yes
For/Rev 1/2 Linefeed	No	Yes	No	Yes
Controls				
Power	Power	Power	Power	Power
On-line/Off-line	On-line/Off-line	On-line/Off-line	Reset	On-line/Off-line
Reset	Reset	Paperfeed/Form Feed	Restart	Form Feed
Line feed	Line feed	Restart	Line feed	Restart
1/12 line feed	1/12 line feed	1/12" paper feed up, down	For 1/8 line feed	1/12" paper feed up,down
Rev 1/8 line feed			Rev 1/8 line feed	
Indicators	3	3	3	3
User Self Test	Yes	Yes	Yes	Yes
Reliability				
Head Life (Chars)(2)	100 Million	100 Million	100 Million	100 Million
MTBF@100%	300 Hrs.	>300 Hrs.	300 Hrs.	>300 Hrs.
Ribbon				
Type	Cartridge	Cartridge	Cartridge	Cartridge
Life	1 Mil Char	1 Mil Char	2 Mil Char	4 Mil Char
Catalog No.	26-1418	26-1418	26-1414	26-1482
Interface				
Buffer	1 Line	2K bytes	1 Line	2K bytes
Parallel Data Bits	8	8	8	8
Status: Busy	Yes	Yes	Yes	Yes
Bits: Fault	Yes	Yes	Yes	Yes
: Paperout	Yes	Yes	Yes	Yes
: Busy	NA	Yes	NA	Yes
: Select	Yes	NA	Yes	NA
Serial (3)				
Data Bits	NA	8	NA	NA
Baud Rate	NA	600/1200	NA	NA
Size	6.3 x 24.2 x 13.3"	6.3 x 24.2 x 13.3"	7.4 x 24.4 x 15.9"	7.4 x 24.4 x 15.9"
Weight	28 lbs.	28 lbs.	42 lbs.	42 lbs.
Power	85W	85W	85W	85W
Price (less cable)	NA	\$1195.00	NA	\$1795.00

Notes: (1) Eliminates leading and trailing spaces in print line
(2) Heavy use of graphics can accelerate head wear
(3) 4 Pin DIN Connector (RS232 DATA, STATUS, GROUND)

BEYOND DOT MATRIX

All the news in Radio Shack printers is not just dot matrix printers. There's the CGP-155 Color Graphics Printer which we told you about in October and a new Daisy Wheel printer, the DWP-410.

DWP-410

The DWP-410 is ideal for giving a professional look in word processing applications using Scripsit. Some of its features are:

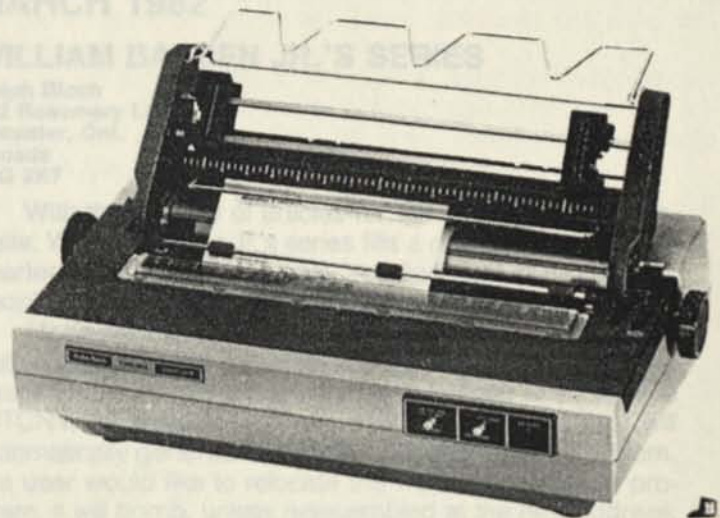
- Switch selectable print densities (10 or 12 characters per inch or proportional pitch) on a 13.6" line
- Forward and reverse paper feed and 1/2-line feed
- Underline, boldface and programmable backspace
- 1/120" space and 1/48" line feed
- Automatic paper set to ease paper insertion. Simply place the sheet of paper at the rear of the 15" platen and pull the Autoset lever to insert the paper.
- It accepts paper up to 16" wide and it will print an original plus 5 copies.
- An external program mode to let you use print wheels with different pitch or special characters
- Interchangeable 124 character print wheels (Courier at 10 pitch, Prestige Elite at 12 pitch, Madeline PS, Cubic PS, Tile Italic at 12 pitch, OCR B, Letter Gothic at 12 pitch, Cubic 15, and PS Bold)
- A nominal print speed of 25 characters per second and 11.7 lines per minute at 10 cpi on a 132 column printout which is over 300 wpm
- Character set: 96 ASCII characters plus special and international for a total of 124 characters plus 21 control codes
- Ribbon—multi-strike carbon or nylon cartridge
- Standard 8 bit parallel interface
- Unidirectional, optimal motion seeking carriage return



Also available for the DWP-410 is an optional bi-directional tractor feed (26-1459 at \$239.95). With the addition of the easily installed tractor feed, the DWP-410 is ideal for printing labels, multi-part forms or long DP runs. The DWP-410 (catalog no. 26-1250, \$1495.00) is an outstanding Daisy Wheel printer.

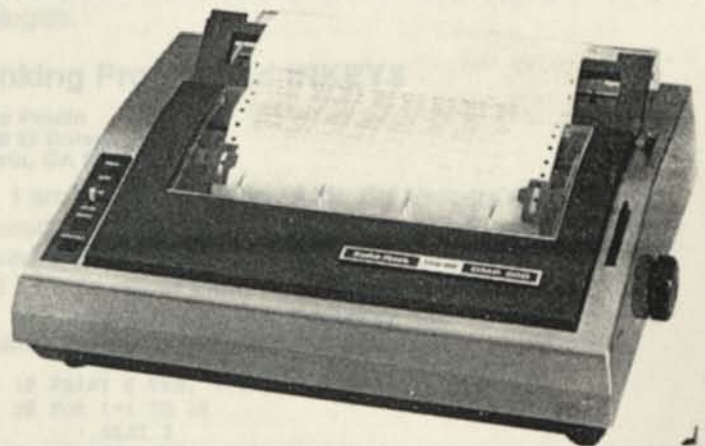
THE FINAL ANALYSIS IS . . .

Each printer whether Dot Matrix or Daisy Wheel is impressive in its own right, and most printing applications from graphics to word processing can be met by these printers.



Note to SuperSCRIPSIT Users with the DMP-200

If you are using SuperSCRIPSIT (26-1590) with the DMP-200 (26-1254), you will need to use the printer driver for the DMP-400 which is on the SuperSCRIPSIT program disk. To load the driver, type DMP400 for the printer type when opening a document.



Notes on Previous Newsletters

JANUARY 1982

Cassette Telephone File

Keith Theriot
P.O. Box 46
Luling, LA 70070

When attempting to create the cassette telephone file I experienced trouble in obtaining the particular record desired.

I remedied this by rewriting program line 540:

From:
540 PRINT
: PRINT A\$, B\$

To:
540 IF A\$ = Z\$ THEN PRINT
: PRINT A\$, B\$

After this correction the proper file appeared every time program execution occurred.

FEBRUARY 1982

Random Files

John K. Segars, Ph.D.
P.O. 829
142-B South Dargan Street
Florence, SC 29503

I would like to note an error in the feature article. On page 5, you stated that if the record number is omitted from the PUT statement, the record written will be number one. This is not true on my Model II (TRSDOS 2.0) or Model III (TRSDOS 1.3). The first record used will be number two. Record one is skipped and contains hexadecimal "e". This can cause problems if you later read record one by using a record number with the GET statement.

Typewriter

John E. Blank, M.D.
3352 E. State St.
Sharon, PA 16146

The February issue included a very neat little program that was written by David Salisbury for use on the Model II. The program runs perfectly well on the Model III with one tiny little flaw — no cursor. You can get by without it as long as you keep your mind on your work. Or a simple fix in Line 500 eliminates the problem.

```
500 CLS  
: PRINT CHR$(14)  
: F=0
```

Mailing List Files

Neal H. Krape
986 Northbriar Drive
York, PA 17404

Mr. William G. S. Smith suggests that the solution to the problem of alphabetizing a file of people's names for mailing lists and other purposes by using two fields, one for the last name and the other for the first name, middle name, etc. "is not an ideal solution" because it "can lead to some pretty strange looking mailing labels, with huge gaps between first name and last name because of unused spaces". He suggests typing the last name twice, once in a special field for alphabetizing, and again in the field used for printing.

He might wish to go back to the solution which he said is not an ideal one, if he includes one more line in his program, probably just before the "PRINT" in which the name is printed. Assuming that the field of last names is "NL" and the field of first names, etc. is "NF" (placing "N" first makes one-variable definition as string possible: DEFSTR N), then insert the following line just before the PRINT instruction:

```
nnnn NP = NF "+" "+" NL
```

Then print NP instead of NF and NL separately. Concatenating the strings to obtain a single string assures that the huge gaps will not appear between first and last name, and avoids the necessity of typing the last name twice.

MARCH 1982

WILLIAM BARDEN JR.'S SERIES

Ralph Bloch
202 Rosemary Lane
Ancaster, Ont.
Canada
L9G 2K7

With the scarcity of articles for the TRS-80 Color Computer, William Barden Jr.'s series fills a great need. His light-hearted style makes for easy reading and alleviates the neophyte's fear of machine language programming.

However, I would like to suggest a change in the use of subroutine calls. In the first statement of the SET routine in the Graphics program, he makes a Long Branch to subroutine INTCNV, an absolute address in ROM. The assembler will automatically generate the correct offset, that is no problem. If a user would like to relocate the machine language program, it will bomb, unless reassembled at the new address, since with the LBSR instruction, only the relative offset is carried in the machine code. The same holds true for line 00310.

To his four rules, I would therefore like to add three more (at least, if you value relocatability):

5. Distinguish between subroutine calls within your own program, and subroutine calls to utilities residing in ROM.

6. Access utilities with absolute calls, e.g. JSR \$B3ED.

7. Access your own subroutines relatively, e.g. BSR ADDMSK, (LBSR, if your program is longer).

I hope that letters like this one will gradually lead to a correspondence of interested assembly language programmers, to discuss the pros and cons of various programming strategies.

Blinking Prompt and INKEY\$

Marc Poulin
6750 El Colegio #103
Goleta, CA 93117

I am writing in regard to Mr. Sal Vescera's letter which appears on page 17. Mr. Vescera describes a simple technique he uses to make a prompt message flash on the screen and I would like to expand on his comments.

Mr. Vescera uses the following four lines of code to achieve the blinking effect:

```
10 PRINT @ 960, "< >ES OR < >0";  
20 FOR I=1 TO 20  
: NEXT I  
30 PRINT @ 960, "<Y>ES OR <N>0";  
40 GOTO 10
```

This results in a rapid flickering of the "Y" and "N" rather than a steady blinking.

The University of Michigan's Highway Safety Research Institute has determined that a light flashing at about four times per second produces the strongest response in automobile drivers (see "Headlight Modulators: Their Function and Use," *Rider*, July 1982, p.51). With this in mind, I would like to suggest modifying Mr. Vescera's algorithm slightly:

```
10 PRINT @ 960, "<Y>ES OR <N>0";  
20 FOR I=1 TO 50  
: NEXT I  
30 PRINT @ 960, "< >ES OR < >0"  
40 FOR I=1 TO 30  
: NEXT I  
50 GOTO 10
```

This flashes the "Y" and "N" at about four times per second and leaves the letters "on" for about 60% of the flashing cycle.

MAY 1982

SuperSCRIPSIT—An Overview

There is a correction to the article. The SuperSCRIPSIT catalog number should be 26-1590.

JUNE 1982

Disk Head Cleaner

Paul G. Kriz
25 Stonehedge Drive
West Nyack, NY 10994

The program on page 17 is a useful addition to everyone's disk system program library. It needs, however, one change to make it work properly on the Model III.

Statement line 100 issues an 'OUT 241,TR' to set track number. While it is true that PORT 241 addresses the floppy disk controller's track register, this register should contain the track you are currently at, not the track that you want.

The track that you desire to seek should be sent to the floppy disk controller's data register which is addressed by PORT 243. Since the "RESTORE OPERATION" on LINE 65 also resets the track register to zero, no further track register commands are needed.

To make the program work properly, change the 'OUT 241,TR' on Line 100 to 'OUT 243,TR'.

Software Driver for Model I Voice Synthesizer

Stephen D. Griffis
RD #1, Truver Road
Valencia, PA 16059

The following routine is a software driver for the TRS-80 Voice Synthesizer. It is an alternate method to that given in the operator's manual. Instead of entering strings of 32 characters or less, you can enter strings of up to the 254 characters. This will save considerably on subroutine calls.

Lines 20-110 demonstrate the use of the subroutine. The driver is in lines 130-220. Line 140 adds the space to the end if the string in the last character is other than a space. This space is necessary. Without it, you will get an "Illegal function error". The main part of the program explains how the spaces are used. Line 150 looks for the first space. VC then points to the end of the first words plus one. Line 160 then stores the first word minus the space into variable VX\$. VY\$ is then shortened by removing the first word and its space. Lines 170-190 are directly out of the manual. They load the individual characters of VX\$ into the Synthesizer buffer. Line 200 is a delay so that the separate words are not loaded into the buffer too fast. If this delay is too short, the words will sound clipped. Line 210 checks to see if done. If VY\$ is only one character long that character has to be the final space. Line 220 is

another delay, though this one is for effect. Without it, the Synthesizer will sound like its talking in one complete sentence.

The timing delay in line 200 is about the best compromise I could come up with between speed (or should I say slowness) of speech and intelligibility. If you keep each word short, you can shorten the delay to about 130.

```
10 CLS
   : CLEAR 500
20 PRINT "THIS PROGRAM IS A DEMONSTRATION OF AN
   ALTERNATE METHOD OF DRIVING THE TRS-80 VOICE
   SYNTHESIZER."
30 VY$="<=ISS P0R[UGR95M IISS 66
   D0435M678NSTR)>4N 768VV 99NN 12LTRN435T
   M435=78D0 768VV DR;#&VIE+ <86 TE. ;8R 43#SS
   @*6D& VO85&SS SIN=78SA;50ZR/"
   : GOSUB 140
40 PRINT "THE SUBROUTINE BELOW ACCEPTS COMPLETE
   SENTENCES AND PHRASES EVEN THOUGH THEY MAY
   BE LONGER THAN 32 CHARACTERS."
50 VY$="<86 S678BR(UUTE.N BEEL[OU 6KS435PTS
   K7MPOLE.T S43NT43NS43S 995NDD FR))*Z435
   EV435NN =[OU <5))*6 M0** BEE& LI2NCR/ <99N
   ==/RD& T((UU K34/45KTR/Z"
   : GOSUB 140
60 PRINT "THE SUBROUTINE BREAKS THE SENTENCE INTO
   ITS INDIVIDUAL WORDS AND THEN PROCESSES THEM
   THROUGH THE VOICE SYNTHESIZER."
70 VY$="<86 S678BR(UUTE.N BR@*6KS <86 S43NT43NS
   IINT(UU IITT0S IIND0DIIVIIDJ(U2LW/RDZ 995NND
   <43NN PR;AS43S43S <43M =R((U <86 VOR5&SS
   SIN=78SA;50ZR/"
   : GOSUB 140
80 PRINT "THE FIRST LINE CHECKS TO SEE IF THE
   LAST CHARACTER IN THE ENTRY IS A SPACE. IF
   NOT, IT ADDS ONE TO THE END."
90 VY$="<86 F/RST L;5#&N TC35KS T((UU SE. IIF <86
   L995ST K34/45KTR/ IINN <86 55NT0R& IISS 66
   SP)*6S"
   : GOSUB 140
   : VY$="IIF NA;8T IITT 995DZ W877NN T((UU <86
   35NDD"
   : GOSUB 140
100 PRINT "THIS SPACE IS NECESSARY SINCE THE
   SPACE IS USED TO DELINEATE THE INDIVIDUAL
   WORDS."
110 VY$="<=ISS SP)*6S IIZZ N435S435S)@R& SINS
   <86 SP)*6S IIZZ &(UUD0 T((UU DE.LINE))*T
   <86 IIND0DIIVIID0(U2L W/RDZ"
   : GOSUB 140
120 END
130 REM DRIVER SUBROUTINE
140 IF RIGHT$(VY$,1)<>CHR$(32) THEN
   VY$=VY$+CHR$(32)
150 FOR VC=1 TO LEN(VY$)
   : IF ASC(MID$(VY$,VC,1))<>32 NEXT
160 VX$=LEFT$(VY$,VC-1)
   : VY$=RIGHT$(VY$,LEN(VY$)- VC)
170 POKE 16383,63
   : POKE 16383,32
180 FOR VX=1 TO LEN(VX$)
   : POKE 16383,ASC(MID$(VX$,VX,1))
   : NEXT
190 POKE 16383,32
   : POKE 16383,63
   : POKE 16383,32
200 FOR I=1 TO 200
   : NEXT
210 IF LEN(VY$)>1 THEN 150
220 FOR I=1 TO 500
   : NEXT
   : RETURN
```

LDOS—The New Operating System for Your TRS-80

by Logical Systems Inc.

Back in 1980 several alternative operating systems offering added features for TRS-80 users with special needs appeared for the TRS-80 Model I and soon after for the Model III. One of the alternative operating systems to appear at that time was LDOS from Logical Systems, Inc.

LDOS is the operating system that Radio Shack has selected for the new Hard Disk, and it will also be available on floppy for those who have not yet purchased Hard Disk.

LDOS AND THE HARD DISK

The power and speed of your Model I or III can now be increased many-fold with the new Hard Drive and its LDOS operating system. Five million characters of storage in one small, quiet and highly reliable 14 by 15 inch package, with access speeds many times faster than floppy drives. Best of all, this new storage device is very cost effective at less than .05 (1/20th) cents per byte versus double density floppy storage at more than .25 (1/4th) cents per byte. This means that Radio Shack Hard Disk storage is about FIVE times more cost effective than floppy drive's for on-line storage.

Getting up and running on the new Radio Shack Hard Drive system is simple and straight-forward as LDOS allows many existing applications to run without any modifications. The Hard Drive system features simplicity, compatibility, speed, reliability and cost effectiveness, too . . . Think about it. As your storage demands increase (and there never does seem to be enough disk space) consider a Radio Shack Hard Drive. It is probably the answer.

LDOS ON FLOPPIES

LDOS is available in floppy disk versions as catalog numbers 26-2213 (Mod I) and 26-2214 (Mod III). The LDOS package is similar in both the floppy and the hard disk versions. Therefore, the following discussion of features will pertain to all versions.

LDOS—AN OVERVIEW

LDOS offers the TRS-80 user many features and enhancements that are not included in TRSDOS. The disk BASIC (LBASIC) that is provided with LDOS is, for the most part, upward compatible with the disk BASIC on TRSDOS Model III. Most BASIC programs for either the Model I or III should run on LDOS without change. LBASIC has several additional features that TRSDOS BASIC does not have. One of the most important features being the high speed LOAD and SAVE of programs and the default extension of /BAS on LBASIC programs. LBASIC also has the ability to go from one program to another and have the values of all variables carried from one program to the next. This makes it possible

to break a program into small pieces and yet have it run as one program. Combined with the high speed load this makes a whole new world available to BASIC programmers because memory constraints become less of a problem. Re-numbering and variable cross-referencing are available as well as several new file modes and single character abbreviations for commonly used command line directives like LIST, AUTO, EDIT and DELETE. There are many other features in LBASIC which I hope to address in future articles, along with ideas on how to use these features.

COMMUNICATIONS

LDOS comes with a very versatile communications utility called LCOMM. LCOMM allows direct upload and download from your RS-232 to/from disk. It also provides many high level "terminal" type functions like printer support (with spooling) and the ability to generate and send all 128 ASCII characters from the keyboard, plus many other features. If communications is a major consideration for you then LDOS is probably worth having for LCOMM alone.

DEVICE INDEPENDENCE

Now I'll touch on the system itself. There is a major design difference between TRSDOS and LDOS that is "Device Independence." Sounds impressive, but what is it? Well, it's really quite simple. Each input or output device (Keyboard, video, printer, RS-232, Diskfile, etc.) is treated very similarly by LDOS. For the most part, these "devices" can be interchanged, interconnected, and redirected. For example, you could tell LDOS to send all characters that would have gone to the video screen to the printer instead. This is called a ROUTE. You could even ROUTE your printer output to a disk file. Then all characters that would have gone to the printer will actually go to the disk file instead. The file can then be viewed or printed out later. You can also have all output to a given device go to another device simultaneously. This is called a LINK.

You can even create new devices or alter the utility of existing devices by giving them a name and turning on a compatible "driver." This is called a SET. For the most part this independence is the same for input and output devices. The standard LDOS devices are *KI for the keyboard, *DO for the video display, and *PR for the printer. Additional devices can be created for special purposes. For instance, we could create a device called *CL to act as a communications line. To use *CL, we might SET *CL TO RS232T. This command sequence would SET the our newly created communications line (*CL) to the Model III RS-232 driver (RS232T), using default settings for the RS-232.

LDOS FILTERS

This Device Independent concept also brings about another powerful feature available in LDOS, the FILTER command. Let us imagine that your computer sends information around inside it, to and from its various devices, one character at a time. This then is the I/O stream between devices and files. Now let us assume that we know the start and end of a certain data path in the machine. We know that a certain character will from time to time pass down this data path on its way to its destination. When this character does appear we wish to take some predetermined action. We may wish it changed to another character, count the number of times this character comes by, or maybe simply throw this particular character away. A good example would be FILTERing your printer to keep certain control characters, (which it is unable to handle), from being presented to it. Some filters are provided with LDOS and are invoked by simply telling LDOS to filter a certain device, with a certain filter. LDOS will then attach this filter to the I/O stream for the specified device, and the data associated with that device will pass through the filter.

PR/FLT

One filter that is provided is the LDOS Printer Filter, PR/FLT. This filter allows you to provide complete control to your printer. With this filter the user can set lines per page, length of page, characters per line, indent on wrap around, left margin, tab expansion, and translate a character from/to and more.

MINIDOS

LDOS also provides a filter called MINIDOS. This filter allows several common DOS commands to be invoked directly from the keyboard, EVEN IN THE MIDDLE OF A RUNNING PROGRAM. MINIDOS will also allow the last DOS command to be repeated with a single key stroke as well as several other features, like sending a Top of Form command to the printer or even killing a file or viewing a disk's directory. All of this is available with single keystroke commands and even from within most running programs.

KSM

Another filter that is provided is the Key Stroke Multiply filter (KSM). This one allows the user to predefine 26 strings or phrases which can then be invoked with a single keystroke. Words, strings and phrases that are commonly used in your day-to-day operations become very easy to enter. This avoids errors and will increase the speed at which you can provide keyboard input to your machine.

LIBRARY COMMANDS

Almost all the library commands in the LDOS system have been endowed with the same functions as those in TRSDOS. These functions in most cases form but a subset of the full utility of the comparable LDOS command. To clarify this a bit, let's just say that a user may execute most commands in the identical manner and syntax and expect a similar result from LDOS. However, in most cases the command has had several additional extensions and/or enhancements added that allow additional functions to be performed. For instance, all LDOS users are familiar with the DIRectory command. In LDOS you can do a DIR :1 just like you can in

TRSDOS, but the LDOS directory will be sorted in alphabetical order when it is displayed on the screen. Wherever possible, every effort has been made to maintain upward compatibility to TRSDOS command syntax in order to make transition by existing TRSDOS users painless, if not pleasant.

Here is a complete list of the LDOS library commands. Most of these commands have many parameters associated with them to modify and control their actions. It should also be noted that there are TWO libraries (A) and (B). This is so the user may remove (KILL) one of these library modules from a system type disk to gain back the space when needed and still have access to a selected library. Further discussions on the unique features of some of the LDOS library commands will be provided in future articles.

Library (A) (SYS6)			
Append	Copy	Device	Dir
Do	Filter	Kill	Lib
Link	List	Load	Memory
Rename	Reset	Route	Run
Set	Spool		
Library (B) (SYS7)			
Attrib	Auto	Boot	Build
Clock	Create	Date	Debug
Dump	Free	Purge	System
Time	Trace	Verify	

JOB CONTROL LANGUAGE

LDOS contains a full Job Control Language (JCL) very similar to those EXEC languages used on Minis and Mainframes. JCL allows the user to write a computer control job stream. JCL is much more than the simple "DO" type command systems that are available in other microcomputer operating systems. LDOS JCL allows the construction of very sophisticated job streams with conditionals (if, then, else) tokens (variables), macros (keywords), user inputs, flashing prompts, sound (if you are set up for sound) and even special start points in the JCL file. JCL can be used in a very simple manner such as going into a language and then running a specified program automatically. As you become familiar with JCL, it could do things for you like cause all the needed commands to be invoked to assemble, link together and create an optimized master LDOS disk, all without user intervention. (Note that when we at LSI assemble LDOS and its utilities in this manner it takes several hours but it would take much longer and be subject to much human error were it to be done without JCL.)

JOBLOG

Most large computers also have a function they call the JOBLOG. With this feature invoked, a record of all commands issued to the DOS and any error reports are recorded to either a disk file or a device, most likely your printer. Each entry to this JOBLOG contains a "time stamp" to show when the action occurred. Of course, the reason for mentioning this high level feature of large systems is that it is fully implemented (as described above) in the LDOS operating system.

PRINTER SPOOLER

A printer SPOOLER that makes use of both RAM and DISK is included with LDOS. The spooler is very simple to invoke and use, and in most cases will greatly increase the

through-put of programs that produce large amounts of printed output. In effect, what a printer spooler does is allow the program to output to the printer even when the printer is still busy printing some previous line. Because the printer is one of the slowest devices (relatively speaking) in a computer system, it is not unusual for the rest of your system, and your program, to be idle while waiting for the printer. The LDOS print spooler helps correct this time wasting situation.

SPEED

In almost all cases, the overall speed of LDOS is slightly greater than that of TRSDOS and in some cases LDOS is much faster than TRSDOS. This can make certain disk oriented programs function faster, and therefore, more pleasant to use.

FILE MAINTENANCE

The file maintenance done by LDOS is a bit different than that of TRSDOS. All files have information in their directory entry to indicate whether or not the file has been backed up since it was last written to, and the date that the file was last written to. This information can be used in conjunction with BACKUP and PURGE, and is viewable with the DIR command.

BACKUP

Speaking of the backup command, I should mention that this command is very much enhanced. The LDOS backup will do a mirror image backup like TRSDOS, but it will also do much more. You can "backup by class" with LDOS. This could be done by specifying to the BACKUP command a partial file specification (partspec). This could be as simple as BACKUP /CMD:0 TO :1, this command will move all the files with the extension "CMD" from drive :0 to drive :1. Backup also allows you to use wildcard characters in the partspec. You can also do controlled backups based on the MOD flag. This would backup only the files that had been written to since they were last backed up. Controlled backups are also available by date or range of dates, by file type, or by their presence or absence on the destination disk. Maintenance of your backups and archive diskettes is simple and very efficient in the LDOS system.

DOCUMENTATION

The documentation provided with LDOS consists of a large three ring binder of about 350 pages, plus an easy to use "User's Guide". It is all logically organized and written in plain English with examples for every function. There is even a glossary and a handy cross reference index. A complete section is devoted to providing detailed technical information on the structure and the entry points of LDOS, as well as RAM storage assignments and detailed specifications regarding the layout of an LDOS type diskette.

COMPATIBILITY

One of the main design features of the LDOS environment is COMPATIBILITY!!! Any LDOS Data disk can be read from and written to by either a Model I or Model III version of LDOS. If you have a Model I and a Model III and use the appropriate LDOS on each machine, the data diskettes are completely compatible. Yes, LDOS will read and write single density 35 track disks on the MODEL III and it will also read

and write 40 track double density disks on the MODEL I (with the Radio Shack Double Density modification and a 40 track drive). With LDOS you can have data such as SCRIPSIT files, BASIC program files, PROFILE data file, VISICALC "data files . . . etc. and use them between models without any type of conversions. LDOS will automatically determine the density and track count of the disk it is accessing and respond accordingly (provided the disk is an LDOS type disk). This powerful feature makes combined Model I and III installations both practical and pleasant.

The only problem with cross compatibility is with some of the programs themselves. It must be noted that programs written to be machine dependent will not be cross compatible between models. Should you have any question about the direct cross compatibility of a program or about its compatibility with LDOS, contact the author or the publisher of the program involved. They should be able to provide you with answers.

Please note that while most Radio Shack programs will run on LDOS without change or with only slight modification, they are only supported by Radio Shack on the operating system that Radio Shack released them on (i.e., TRSDOS 2.3 for Model I and TRSDOS 1.3 for Model III.) Many Radio Shack packages are being checked for compatibility with LDOS and the information will be included with the product. Please note that we will NOT be attempting to correct, modify, patch, or in any other way make non-Radio Shack programs or hardware work with LDOS or the Hard Drive system. This would be an impossible undertaking. For non-Radio Shack software, you will have to contact the author or publisher of the product.

One more important point should be made before you run down to your local Radio Shack and order yourself a copy of LDOS. You should be aware that the authors of LDOS and Radio Shack strongly recommend that LDOS not be purchased by single drive owners. LDOS will function with a single drive, but some LDOS functions will be very cumbersome, and certain high level functions will not function at all with a single drive.

DO YOU NEED LDOS?

All this may sound great, but one major question must be addressed. Do you need to have LDOS? The answer is, in many cases, a simple "No, you don't." If the major use of your computer is with functional applications that you are comfortable and pleased with, then chances are LDOS would be of little value to you.

I don't want you to think that I'm saying most users should not buy LDOS. LDOS is a fine operating system that can do many things that TRSDOS cannot. But you must justify some use for these additional features, or LDOS will be of little value other than the increase in the speed of disk activity. The price of LDOS is \$129 and this cost should be weighed against the additional features that you may use. Also, if you are just beginning your use of the TRS-80 you will find TRSDOS to be sufficient in most cases. In all instances it is suggested that you become thoroughly familiar with the use of TRSDOS before "moving up" to the additional capabilities of LDOS.

If you are totally familiar with TRSDOS and/or you actively write software for your own use or for commercial purposes, you will find many new and intriguing possibilities

with the LDOS system and should, therefore, seriously consider obtaining LDOS.

In the months to come this column will contain hints and tips on the use of LDOS and its special features, which are aimed at the general user and the BASIC programmer, as well as technical information on the inner workings of LDOS and interfacing machine language programs to LDOS in the proper manner.

Printing Titles

Bill Christ
3514 Balfour Street
Saskatoon, Saskatchewan
CANADA S7H 3Z6

This is a little routine which will center a title of your choice and underline it.

```
10 INPUT "ENTER YOUR TITLE HERE"; A$
20 L=LEN(A$)
   : T=35-INT(.5*L)
30 LL=6*L*2
35 LPRINT CHR$(31)
40 LPRINT TAB(T/2); A$;
   : LPRINT CHR$(30); CHR$(26)
45 FOR X=1 TO T:LPRINT " ";
   : NEXT
50 LPRINT CHR$(18);
60 LPRINT CHR$(28); CHR$(LL); CHR$(129)
65 LPRINT CHR$(30)
70 END
```

A Pascal Division Program

James W. Wood
424 N. Missouri
Atwood, IL 61913

Do you ever wonder what numbers don't show on your calculator or computer when dividing? My calculator display reads .07692308 when I divide one by thirteen. I know from Algebra that if the answer to this problem has a repeating block of numbers that it is no longer than twelve digits long (one less than the divisor).

I decided to write a computer program to calculate the answer "exactly." Since I want to increase my knowledge of Pascal, I decided this was a good time to become bilingual. The Pascal program works fine unless you try to divide numbers which exceed Tiny Pascal's integer range. On a repeating decimal press <ENTER> once to pause, twice to terminate program.

```
VAR A,B,C,D,E,F:INTEGER;
BEGIN
  WRITE(28,31,'ENTER YOUR DIVIDEND');
  READ(A#);
  WRITE(13,13,'ENTER YOUR DIVISOR');
  READ(B#);
  C:=A DIV B;
  WRITE(13,13,C#,'. ');
  D:=(A MOD B)*10;
  REPEAT
```

```
E:=D DIV B;
WRITE(E#);
F:=D-(E*B);
D:=F*10;
UNTIL D=0;
END.
```

Model I/III Bugs, Errors, Fixes

NOTE TO USERS:

The following changes and corrections are provided for your information. If you have an applications program which is working correctly, you should probably NOT make any changes to it. If you feel that changes should be made, but you do not feel qualified to make the change yourself, please contact your local Radio Shack Computer Center or Expanded Computer Department for assistance. If you do not have access to one of these stores, then you may want to call Computer Customer Service in Fort Worth for assistance.

MEDICAL OFFICE SYSTEM (26-1568)

The program does not print zero balance accounts when option 2 of statement printing is selected. Make the following changes to the program BLLPRT.

```
1100 PRINT@728,"SELECT OPTION: ";FL=-1:GOSUB210
   :Q=VAL(IN$):PRINT:IFQ<1ORQ>3THEN1100ELSE
   PF=Q-1:X0=Q<>2:IFQ=1THENPOKE16424,43ELSEPOKE
   16424,67
1130 IFA=AHTHEN1140ELSEIFX0AND(BA=0ORIS="N")THEN
   1210ELSEGOSUB1215:IFPFTHENLPRINTLF$:LPRINT
   LF$:LPRINTLF$:IFBA<0THELPRINTTAB(40)"CREDIT
   BALANCE"ELSELPRINTTAB(42)"PLEASE REMIT"
1140 IFX0AND(BA=0ORIS="N")THENGOSUB1250:GOTO1210
   ELSEGOSUB1215
```

MODEL III DISK COURSE (26-2014)

In Lesson 1 Part 2 when running the section of the course that says:

A BASIC program is saved with the filename PAYROLL.BAS.RT33N:0

Which of the following commands will load and run the program?

1. RUN"PAYROLL/BAS.RT33N"
 2. RUN"PAYROLL.RT33N:0"
 3. RUN"PAYROLL/BAS"
 4. LOAD"PAYROLL/BASD.RT33N:0"
- (1, 2, 3, OR 4) 4

The following error message will appear:
Syntax Error in 450
READY
450

The first colon (:) in line 450 should be deleted so that the line reads:

```
450 PRINT"No, LOAD"Z$"PAYROLL/BAS.RT33N:0"Z$" is
   incorrect because ":PRINT""LOAD" will not
   execute the program after it is loaded
   into:PRINT" memory.":GOSUB 7
```


More TAB and USING in a Total Inventory Program

Robert E. Perkins
2914 Ashley River Road
Charleston, SC 29407

Recent discussion on how to use both the "TAB" and "USING" functions to line up decimal points prompted the following program. It works on a Model I with a LP VI. First look at what an example run of the program might look like.

```
SUMMARY OF OTHER BRANDS
ARMSTRONG      $ 123.55
SEMPERIT       $ 1,256.25
OTHER          $ 1,254.25
VALVES         $ 12.25

TOTAL OTHER BRANDS      $ 2,646.30
CARLISLE              $ 12,365.25
MICHELIN              $ 0.00
GILLETTE              $ 12,354.25
GOODRICH              $ 32,145.25
INDUSTRIAL            $ 13.22
TUBES                 $ 1,225.25

TOTAL INVENTORY      $ 60,749.52
```

The Total Inventory Program

```
2 DEFBL A, O, S, G, C, H, Z, Y, I
4 K$="###.###.##"
5 LPRINT CHR$(31)
10 LPRINT TAB(5) "SUMMARY OF INVENTORY LISTING"
15 INPUT "ENTER THE DATE-MM/DD/YY"; A$
20 LPRINT TAB(5); A$
30 FOR X=1TO6
   : LPRINT CHR$(138)
   : NEXT X
40 LPRINT TAB(10) "SUMMARY OF OTHER BRANDS"
50 INPUT "ARMSTRONG"; A
   : INPUT "OTHER BRANDS"; O
   : INPUT "SEMPERIT"; S
60 INPUT "GILLETTE"; G
   : INPUT "CARLISLE"; C
65 INPUT "GOODRICH"; H
70 INPUT "TUBES"; T
75 INPUT "VALUES"; V
76 INPUT "INDUSTRIAL"; I
80 INPUT "MICHELIN"; M
90 LPRINT TAB(15) "ARMSTRONG"; TAB(30) USING K$; A
100 LPRINT TAB(15) "SEMPERIT"; TAB(30) USING K$; S
110 LPRINT TAB(15) "OTHER"; TAB(30) USING K$; O
115 LPRINT TAB(15) "VALVES"; TAB(30) USING K$; V
120 Y=A+S+O+V
130 LPRINT TAB(20) "TOTAL OTHER BRANDS"; TAB(50)
   USING K$; Y
140 LPRINT TAB(20) "CARLISLE"; TAB(50) USING K$; C
150 LPRINT TAB(20) "MICHELIN"; TAB(50) USING K$; M
160 LPRINT TAB(20) "GILLETTE"; TAB(50) USING K$; G
165 LPRINT TAB(20) "GOODRICH"; TAB(50) USING K$; H
166 LPRINT TAB(20) "INDUSTRIAL"; TAB(50) USING K$; I
170 LPRINT TAB(20) "TUBES"; TAB(50) USING K$; T
180 LPRINT CHR$(138)
190 Z=Y+C+M+G+T+H+I
200 LPRINT TAB(20) "TOTAL INVENTORY"; TAB(50) USING
   K$; Z
```

Line Printer Set Up

William Allen
2012 Metairie Court, #24
Metairie, LA 70002

```
10 ' LINE PRINTER SET UP PROGRAM (TITLE - LPSU)
20 CLS
30 PRINT "LINE PRINTER SET UP COMMANDS"
40 PRINT "FOR QUICK PRINTER II      2"
50 PRINT "FOR LINE PRINTER IV       4"
60 PRINT "FOR LINE PRINTER VII     7"
70 INPUT "<ENTER> WHICH PRINTER ";W
   : CLS
80 IF W=2 THEN 200
90 IF W=4 THEN 300
100 IF W=7 THEN 500
200 PRINT "FOR QUICK PRINTER II"
210 PRINT "CARRIAGE RETURN 1"
220 PRINT "DOUBLE WIDTH 2"
230 INPUT "<ENTER> CHOICE ";C
240 IF C=1 THEN 1180
250 IF C=2 THEN 1140
300 PRINT "FOR LINE PRINTER IV"
310 PRINT "DOUBLE WIDTH 1"
320 PRINT "PROPORTIONAL 2"
330 PRINT "STANDARD PRINT 3"
340 PRINT "CONDENSED PRINT 4"
350 PRINT "START UNDERLINE 5"
360 PRINT "STOP UNDERLINE 6"
370 INPUT "<ENTER> CHOICE ";C
380 IF C=1 THEN 1000
390 IF C=2 THEN 1020
400 IF C=3 THEN 1040
410 IF C=4 THEN 1060
420 IF C=5 THEN 1140
430 IF C=6 THEN 1160
500 PRINT "FOR LINE PRINTER VII"
510 PRINT "CARRIAGE RETURN/LINE FEED 1"
520 PRINT "GRAPHIC MODE 2"
530 PRINT "STANDARD PRINT 3"
540 PRINT "DOUBLE WIDTH 4"
550 INPUT "<ENTER> CHOICE ";C
560 IF C=1 THEN 1160
570 IF C=2 THEN 1120
580 IF C=3 THEN 1100
590 IF C=4 THEN 1080
1000 LPRINT CHR$(27); CHR$(14)
1010 NEW
1020 LPRINT CHR$(27); CHR$(17)
1030 NEW
1040 LPRINT CHR$(27); CHR$(19)
1050 NEW
1060 LPRINT CHR$(27); CHR$(20)
1070 NEW
1080 LPRINT CHR$(31)
1090 NEW
1100 LPRINT CHR$(30)
1110 NEW
1120 LPRINT CHR$(18)
1130 NEW
1140 LPRINT CHR$(15)
1150 NEW
1160 LPRINT CHR$(14)
1170 NEW
1180 LPRINT CHR$(13)
1190 NEW
```

Changing Print Size on the LP V and VI

Robert P. Graham
2709 N. Sibley Street
Metairie, LA 70003

Here are four short machine language programs that can be run on the Model II to change the print size of the Line Printer V and VI. They reside above user RAM, which means they can be run from TRSDOS READY or "called" from BASIC (using the SYSTEM " " command) or from VisiCalc (using /SE). We find them particularly useful in VisiCalc where it is often necessary to print more than 132 columns on a line. With SET15, you can print 198 columns, with SET75 or SET5, you can print anything in large size type, and with SET10, you can return the printer to normal operation.

You can execute these while the printer is anywhere on the page. The only restriction is that they must be executed while the printer is at the beginning of a line (no problem in VisiCalc). This is because the printer initialization function sets the print head to the beginning of the current line, and anything already printed on that line will be printed over. There is nothing copyrighted here, as it is just a series of Model II supervisor calls.

Program lines 110-130 get the current printer status. PUSH HL saves the current line count. Lines 150-190 initialize the printer, setting a different number of columns for each program. The next five lines send a series of codes to the printer, setting the print size. The current line count is then POP-ed off the stack. Finally, a supervisor call restores the line count to TRSDOS. This is necessary because the printer initialization sets it to zero.

The assembly language and the hex code is included, so if you do not have an assembler, you can enter the hex code in DEBUG beginning at address F380. To store the programs, return to TRSDOS after entering the code, and use the DUMP command (DUMP SET15 [START=F380, END=F3A7]). Use a different file name for each program. The beginning and ending addresses shown above can be used for all four.

I hope these can be as useful to you as they have been to

us.

SET5 (For printing in large size type)

F380		00100	ORG	0F380H
F380	3E5F	00110	START LD	A,95
F382	0600	00120	LD	B,0
F384	CF	00130	RST	8
F385	E5	00140	PUSH	HL
F386	3E11	00150	LD	A,17
F388	0642	00160	LD	B,66
F38A	0E3C	00170	LD	C,60
F38C	1642	00180	LD	D,66
F38E	CF	00190	RST	8
F38F	3E13	00200	LD	A,19
F391	21A1F3	00210	LD	HL,MSG1
F394	0604	00220	LD	B,MSG1
F396	0E00	00230	LD	C,00
F398	CF	00240	RST	8
F399	E1	00250	POP	HL
F39A	3E5F	00260	LD	A,95
F39C	4D	00270	LD	C,L
F39D	0603	00280	LD	B,3
F39F	CF	00290	RST	8

F3A0	C9	00300		RET	
F3A1	1B	00310	MSG1	DEFB	27
F3A2	0F	00320		DEFB	15
F3A3	1E	00330		DEFB	30
F3A4	1F	00340		DEFB	31
F3A5	00	00350	MSG2	DEFB	00
0004		00360	MSG1	EQU	MSG2-MSG1
F380		00370		END	START
00000	Total Errors				

MSG2	F3A5
MSG1	0004
MSG1	F3A1
START	F380

SET75 (To print with large size type)

F380		00100		ORG	0F380H
F380	3E5F	00110	START	LD	A,95
F382	0600	00120		LD	B,0
F384	CF	00130		RST	8
F385	E5	00140		PUSH	HL
F386	3E11	00150		LD	A,17
F388	0642	00160		LD	B,66
F38A	0E3C	00170		LD	C,60
F38C	1642	00180		LD	D,98
F38E	CF	00190		RST	8
F38F	3E13	00200		LD	A,19
F391	21A1F3	00210		LD	HL,MSG1
F394	0606	00220		LD	B,MSG1
F396	0E00	00230		LD	C,00
F398	CF	00240		RST	8
F399	E1	00250		POP	HL
F39A	3E5F	00260		LD	A,95
F39C	4D	00270		LD	C,L
F39D	0603	00280		LD	B,3
F39F	CF	00290		RST	8
F3A0	C9	00300		RET	
F3A1	1B	00310	MSG1	DEFB	27
F3A2	0F	00320		DEFB	15
F3A3	1E	00330		DEFB	30
F3A4	1F	00340		DEFB	31
F3A5	1B	00350		DEFB	27
F3A6	0E	00360		DEFB	14
F3A7	00	00370	MSG2	DEFB	00
0006		00380	MSG1	EQU	MSG2-MSG1
F380		00390		END	START
00000	Total Errors				

MSG2	F3A7
MSG1	0006
MSG1	F3A1
START	F380

SET10 (Returns printer to normal operation)

F380		00100		ORG	0F380H
F380	3E5F	00110	START	LD	A,95
F382	0600	00120		LD	B,0
F384	CF	00130		RST	8
F385	E5	00140		PUSH	HL
F386	3E11	00150		LD	A,17
F388	0642	00160		LD	B,66
F38A	0E3C	00170		LD	C,60
F38C	1684	00180		LD	D,132
F38E	CF	00190		RST	8
F38F	3E13	00200		LD	A,19
F391	21A1F3	00210		LD	HL,MSG1
F394	0603	00220		LD	B,MSG1
F396	0E00	00230		LD	C,00
F398	CF	00240		RST	8
F399	E1	00250		POP	HL
F39A	3E5F	00260		LD	A,95
F39C	4D	00270		LD	C,L
F39D	0603	00280		LD	B,3
F39F	CF	00290		RST	8
F3A0	C9	00300		RET	
F3A1	1B	00310	MSG1	DEFB	27

```

F3A2 0F      00320      DEFB      15
F3A3 1E      00330      DEFB      30
F3A4 00      00340      MSG2     DEFB      00
0003      00350      MSG1     EQU      MSG2-MSG1
F380      00360      END      START
00000 Total Errors

```

```

MSG2      F3A4
MSG1     0003
MSG1     F3A1
START     F380

```

SET15 (To print 198 columns)

```

F380      00100      ORG      0F380H
F380 3E5F    00110      START  LD      A,95
F382 0600    00120      LD      B,0
F384 CF      00130      RST     8
F385 E5      00140      PUSH   HL
F386 3E11    00150      LD      A,17
F388 0642    00160      LD      B,66
F38A 0E3C    00170      LD      C,60
F38C 16C6    00180      LD      D,198
F38E CF      00190      RST     8
F38F 3E13    00200      LD      A,19
F391 21A1F3  00210      LD      HL,MSG1
F394 0605    00220      LD      B,MSG1
F396 0E00    00230      LD      C,00
F398 CF      00240      RST     8
F399 E1      00250      POP    HL
F39A 3E5F    00260      LD      A,95
F39C 4D      00270      LD      C,L
F39D 0603    00280      LD      B,3
F39F CF      00290      RST     8
F3A0 C9      00300      RET
F3A1 1B      00310      MSG1   DEFB     27
F3A2 0F      00320      DEFB     15
F3A3 1E      00330      DEFB     30
F3A4 1B      00340      DEFB     27
F3A5 0E      00350      DEFB     14
F3A6 00      00360      MSG2   DEFB     00
0005      00370      MSG1   EQU     MSG2-MSG1
F380      00380      END     START
00000 Total Errors

```

```

MSG2      F3A6
MSG1     0005
MSG1     F3A1
START     F380

```

Descenders on the LP VII

Joe Diel
337 Rhode Island NE
Albuquerque, NM 87108

This program uses the graphics capability of the Line Printer VII to print descenders or lower case g, j, p, q, and y. All other symbols are standard line printer symbols. The program was written for Model III BASIC, 16K RAM. Input is in response to a prompt; to exit use the <BREAK> key.

```

10 REM TRS-80 MODEL III LINE PRINTER VII
20 REM PROGRAM TO PRINT LOWER CASE WITH DESCENDERS
30 REM J.DIEL, ALBUQUERQUE, NM, DECEMBER 1981
35 CLEAR 1000
40 DEFINT I-N
   : DIM L(13,4),IP(20),ID(20),B$(4)
45 REM SET UP PATTERNS FOR LOWER CASE G, J, P, Q, Y
50 DATA 18, 184, 196, 196, 168, 252, 30
60 DATA 18, 128, 132, 132, 132, 131, 30

```

```

70 DATA 18, 128, 128, 128, 128, 253, 30
80 DATA 18, 128, 132, 132, 132, 131, 30
90 DATA 18, 252, 168, 196, 196, 184, 30
100 DATA 18, 135, 128, 128, 128, 128, 30
110 DATA 18, 184, 196, 196, 168, 252, 30
120 DATA 18, 128, 128, 128, 128, 135, 30
130 DATA 18, 188, 192, 192, 160, 252, 30
140 DATA 18, 128, 132, 132, 132, 131, 30
150 FOR I=0 TO 4
160 FOR J=0 TO 13
170 READ L(J,I)
180 NEXT J
190 NEXT I
200 REM SET UP ARRAY B$ FOR TESTING FOR LOWER CASE
   G,J,P,Q, OR Y
210 DATA "g", "j", "p", "q", "y"
220 FOR I = 0 TO 4
   : READ B$(I)
   : NEXT I
300 REM INPUT LINE TO BE PRINTED
310 REM FOR MORE COMPLEX LPRINT PROGRAMS MAIN PROGRAM
320 REM SHOULD BE BETWEEN 300 AND 5999
330 PRINT "INPUT LINE(S) TO BE PRINTED"
340 INPUT A$
   : GOSUB 6000
   : GOTO 340
6000 REM SUBROUTINE TO FIND LOWER CASE G,J,P,Q, OR Y
   IN A$
6010 REM AND PRINT A$ TO LINE PRINTER VII
6020 NP=0
6030 FOR I=1 TO LEN(A$)
6040 FOR J=0 TO 4
6050 IF MID$(A$,I,1)=B$(J) THEN 6080
6060 NEXT J
6070 GOTO 6090
6080 NP=NP+1
   : IP(NP)=I
   : ID(NP)=J
6090 NEXT I
6100 IF NP=0 THEN 6290
6110 IF IP(NP(1)) <> 1 THEN LPRINT
   MID$(A$,1,IP(1)-1);
6120 FOR J=0 TO 6
   : LPRINT CHR$(L(J,ID(1)));
   : NEXT J
6130 LPRINT CHR$(18); CHR$(128); CHR$(30);
6140 IF NP=1 THEN 6185
6150 FOR I=2 TO NP
6160 LPRINT MID$(A$,IP(I-1)+1,IP(I)-IP(I-1)-1);
6170 FOR J=0 TO 6
   : LPRINT CHR$(L(J,ID(I)));
   : NEXT J
6180 NEXT I
6185 IF LEN(A$)>IP(NP) THEN LPRINT MID$(A$,IP(NP)+1);
6190 LPRINT CHR$(18); CHR$(10); CHR$(30);
6200 IF IP(NP(1)) <> 1 THEN LPRINT STRING$(IP(1)-1,
   " ");
6210 FOR J=7 TO 13
   : LPRINT CHR$(L(J,ID(1)));
   : NEXT J
6215 LPRINT CHR$(18); CHR$(128); CHR$(30);
6220 IF NP=1 THEN 6270
6230 FOR I=2 TO NP
6240 LPRINT STRING$(IP(I)-IP(I-1)-1," ");
6250 FOR J=7 TO 13
   : LPRINT CHR$(L(J,ID(I)));
   : NEXT J
6260 NEXT I
6270 LPRINT CHR$(18); CHR$(10); CHR$(30);
6280 RETURN
6290 LPRINT A$;
   : FOR I=1 TO 2
   : LPRINT CHR$(18); CHR$(10); CHR$(30);
   : NEXT I
   : RETURN

```

Model II Scripsit 2.0, LP VIII and the New Printers

Bruce Elliott

As stated in the Scripsit manual, Scripsit 2.0 was designed to work with a number of different printers. In this article we will suggest several techniques which will help optimize the use of Scripsit 2.0 and Radio Shack's LP VIII, the new DMP series printers and the new DWP-410. This information should also provide you with a guide for using other dot matrix printers. Check your printer manual for specific information on how your printer works.

START-UP

Page 74 of the Scripsit 2.0 manual tells how to initialize a printer automatically during startup. This is especially valuable for a printer like the Line Printer VIII. The Line Printer VIII powers up in data processing mode, and you will want it to be in the word processing mode for use with Scripsit 2.0. Page 24 of the LP VIII manual tells you what control codes are available in the LP VIII.

The Scripsit manual goes on to show a sample startup procedure for an LP VIII:

STARTUP 141B0E1B14

This code sequence can be deciphered by checking page 24 of the LP VIII manual:

- 14—Select Word Processing Mode
- 1B0E—Start Character Elongation
- 1B14—Select the Condensed Character Set

This startup method works very well, but you should remember one thing: Don't RESET the printer! Resetting the printer will reset the standard character set and data processing modes, not at all what you want.

Radio Shack's new DMP (Dot Matrix Printer) series allows you to "custom-set" the power-up mode and pitch, so you should not need the special start-up capabilities of Scripsit 2.0.

With any printer which offers changeable pitch and font, you may find yourself wanting to use these features in the documents you create.

You can do this by using Scripsit's printer codes. I use "I" (for Initialize) to allow resetting my LP VIII if needed. Here is how I set up the "I" key: Using the above sequence, I see that there are 5 pairs of hexadecimal (hex) values—14 1B 0E 1B and 14. In the Scripsit Printer Control Code Utility, I select "I", then enter a 5 for the number of ASCII characters, then 0 for the Justification Value. (Since these are printer control codes, not printable characters, I don't want them counted when Scripsit tries to justify a line.) Next, I enter the 5 pairs of values. Finally I press <ENTER> and enter a short description "Init 8.3 CPI".

Now, anytime I need to initialize my LP VIII to 8.3 cpi (Characters Per Inch) I simply place a <CTRL><X><I> at the beginning of the text to be printed.

Navigation with a CC

Earlier this year I purchased my Enhanced BASIC Color Computer with the intent of programming Navigation programs. It was during the process of code-referencing the old from one file to another.

Every thing worked out fine with the bottom line. The whole CC system was given as an argument of COSM.

UNDERLINING AND BOLDFACE

Two features of Scripsit 2.0 which enhance the look of a document are underlining and boldface. If you are using one of the printers we are discussing, and you type in a line of text containing Scripsit's underline code, you may not be overly pleased by the result. It will be obvious that something is not quite right.

If you watch the printer closely, you will notice that while underlining (or trying to), the printer prints a character, backs up (the wrong distance), prints an underscore, prints the next letter, and continues until the text to be underlined has been printed.

The problem is that Scripsit comes set-up for the DW II printer. The DW II accepts a single hex character (08) as the command for backspacing one full character. Printers like the LP VIII, DMP series and the DWP-410 use a two character hex sequence to indicate the distance to move the printhead back.

Page 74 of the Scripsit manual gives us information about changing the "distance for backspace." Scripsit is designed to be patched for printers which use the two hex digit backspace.

When you patch Scripsit 2.0's backspace distance, we recommend that you put in the value for your most commonly used character size. The reason for this is that you can only insert a single value, and you can not change that value during the course of printing a document.

In Scripsit 2.0 there are two basic functions which use the patched backspace distance: underlining and bold face.

The DMP series of printers and the DWP-410 have built-in underline and boldface capabilities (LP VIII has built-in underline, but not boldface.) This means that you can avoid having to patch Scripsit, and use the capabilities of your printer to the fullest.

For the LP VIII, a quick glance at the manual shows that the printer uses hex 0F to turn underline on, and hex 0E to turn underline off. I have set these codes up as "U" to turn underline on, and "O" to turn underline off.

We can now underline any phrase, in any of the LP VIII type sizes, by using the codes <CTRL><X><U> and <CTRL><X><O>.

To use boldface on normal LP VIII characters, we make the following patch to Scripsit 2.0:

```
PATCH SCRIPSIT A = DDFC F = 00 C = 0C
```

The "0C" in the C = string is hex for 12, which is the number of dot columns to backspace on the LP VIII. I also recommend reducing the number of strikes for bold characters from the preset five to three:

```
PATCH SCRIPSIT A = DDFD F = 05 C = 03
```

This change produces a slightly thinner bold character, which I prefer. You may wish to experiment with several values to see what you like best.

For the DMP and DWP-410 printers, the printer manual contains a chart which gives you the proper codes to turn underline or boldface on or off. The manuals also include information of the dot column widths of characters for patching Scripsit.

CHANGING CHARACTER SIZE

One of the reasons for having a dot matrix printer like the ones we are discussing is that the printer is capable of printing more than one size character. Page 34 of the LP VIII manual shows six different combinations of character font and size.

LP VIII is capable of printing three basic character fonts: Proportional, Condensed and Ordinary. Since Model II Scripsit does not provide justification with proportional characters, I do not use this font. The two remaining fonts, when used normal size or elongated, provide me with four character sizes that I can use in my documents.

To be able to use all of these character sizes, I need some additional printer control codes. Again checking the LP VIII manual, I find the following codes that look useful:

- 1B 13—Select Normal Character Set
- 1B 14—Select Condensed Character Set
- 1B 0E—Start Elongation
- 1B 0F—Stop Elongation

I have set these up as printer codes "N", "C", "E" and "P" respectively. You need to remember that elongation will remain on until you turn it off. Use care when you go from one character size to another, and turn elongation off when you don't want it.

Here is how Scripsit 2.0 is set up for my LP VIII, and my printer control codes. As you can see from the chart, I have defined print codes for several special characters as well as the control codes we have discussed.

SCRIPSIT PRINTER CONTROL CODES

xA 1 1 AB	Copyright	xN 2 0 1B13	Normal Char
xB 1 1 AB	Dagger	xO 1 0 0E	Undrline Off
xC 2 0 1B14	Condensed Ch	xP 2 0 1B0F	Elongate Off
xD 1 1 A6	Degree Symb	xQ 1 1 AC	1/4
xE 2 0 1B0E	Start Elongt	xR 3 3 285229	(R)
xF 1 1 BF	script f	xS 1 1 B4	cents
xG 0 0		xT 4 4 28544D29	(TM)
xH 1 1 AE	1/2	xU 1 0 0F	Undrline On
xi 5 0 141B0E1B14	Init 8.3 cpi	xV 0 0	
xJ 0 0		xW 0 0	
xK 0 0		xX 0 0	
xL 0 0		xY 0 0	
xM 0 0		xZ 0 0	

Which control code would you like to change?

Enter A—Z, ESC to store, BREAK to cancel, or HOLD to restore previous:

SUBSCRIPTS AND SUPERSCRIPTS

Once initialized to the word processing mode, (regardless of character size), the printers we have been discussing will handle subscripts and superscripts correctly. No special codes needed other than the normal Scripsit codes.

One more comment about Scripsit and Dot Matrix Printers. If you use character justification, speed of the printer is reduced dramatically. In order to place the proper number of micro-spaces between characters, the printer must stop its printhead, back up, regain its forward speed, locate the proper dot column, skip it, and then print the next character. This process takes a lot of time, and the speed of the printer suffers.

Navigation with a CC

Leo McNeil
6665 Koralee
Tucson, AZ 85710

Earlier this year I purchased my Extended BASIC Color Computer with the intent of programming Navigation problems, thus eliminating the tedium of cross-referencing one list from one table with another.

Every thing worked out fine until the bottom line: The answer, by formula, was given as an argument of COS(M). No built in ARCCOS!

I resolved the problem through the laws of Sines, Cosines and Tangents. The formula for Great Circle Sailing:

$$\cos(c) = \cos(a)\cos(b) + \sin(a)\sin(b)\cos C$$

was resolved thus:

$$\sin^2(c) + \cos^2(c) = 1$$

$$\sin^2(c) = 1 - \cos^2(c)$$

$$\sin(c) = \text{SQR } 1 - \cos^2(c)$$

$$\tan(c) = \sin(c)/\cos(c)$$

Where M in Line 300 = Cos(c), CC in Line 310 converts M to Tan(c).

```

10 CLS
15 CLEAR 300
20 'A PROGRAM TO COMPUTE - WITH EXTENDED COLOR
    COMPUTER - GREAT CIRCLE DISTANCES
21 'CASE I :LAT/LONG SAME NAMES
22 'CASE II:LATS SAME NAMES;LONGS DIFFERENT NAMES
23 'CASE III:LAT DIFFERENT NAME;LONGS SAME NAMES
24 'CASE IV:LATS/LONGS DIFFERENT NAMES
25 'K IS THE RAD/DEG/RAD CONVERSION FACTOR
30 K=.0174532933
35 'KM IS THE MILES/MS/MILES CONVERSION FACTOR
40 KM=.6213699
45 DS="#####.## % Z"
100 'PD=POINT(LAT/LONG) OF DEPARTURE
105 'PA=POINT(LAT/LONG) OF ARRIVAL
150 'INPUT LARGEST LATITUDE FIRST
155 INPUT "LAT:DG, MN, N OR S"; DA, MA, AS
160 INPUT "LONG:DG, MN, E OR W"; AD, AM, AAS
165 INPUT "LAT:DG, MN, N OR S"; DB, MB, BS
170 INPUT "LONG:DG, MN, E OR W"; BD, BM, BBS
175 INPUT "PORT OF DEPARTURE"; DP$
176 INPUT "PORT OF DESTINATION"; DS$
200 A=DA+MA/60
205 AL=AD+AM/60
210 B=DB+MB/60
215 BL=BD+BM/60
220 A1=(90-1)*K
225 B1=(90-B)*K
230 IF AL<BL THEN 240 ELSE 235
235 C=(BL-AL)*K
    : GOTO 300
240 C=(AL-BL)*K
    : GOTO 300
300 M=((COS(A1)*COS(B1))+SIN(A1)*SIN(B1)*COS(C))
305 'CC IS COS(M) EXPRESS AS ARCTAN(M)
310 CC=ATN((SQR(A-M^2)))/(M)/K
315 'DM IS THE DISTANCES IN MILES:DK IN KMS
320 DM=CC*60
325 DK=DM*KM
400 PRINT#-2, "DISTANCE FROM "; DP$;
401 PRINT#-2, " TO "; DS$ " IS "
    : PRINT#-2
405 PRINT#-2, USING DS$; DM, "MILES"
    : PRINT#-2
410 PRINT#-2, USING DS$, DK, "KMS"
    : PRINT#-2
    : PRINT#-2
415 PRINT#-2, "L. MCNEIL, CDR USN (RET)"
500 END

```

Word Processing on the Pocket Computer

Don Hergert
2300 Fremontia Dr. #106
San Bernardino, CA 92404

Editor's Note: We are very pleased to announce that an excellent unpublished manuscript containing programs and text for the pocket computer (PC-1) was offered to us by Don Hergert. And of course, we accepted. In the coming months we will be publishing several segments from the manuscript.

INTRODUCTION

The Pocket Computer represents perhaps one of the greatest challenges available to the experienced BASIC programmer: to produce valuable working programs with practicality, for an extremely limited computer. As much as I like the Pocket Computer, the facts of life in the personal computer market say that any computer with under 64K bytes of RAM, or without the capacity to upgrade to Disk Operating Systems, or without the other features available in many of today's computers, is limited. If you've never done any programming on a larger computer, you probably don't understand the comments made by those who have. If they, in turn, haven't done programming on the Pocket Computer, they probably cannot fathom working on something so small.

Undoubtedly the biggest hurdle to overcome in working with the Pocket Computer is not tied to the limitations of the hardware. The biggest hurdle is a limited imagination. We all have it. For you it might be impossible to imagine this "alpha-numeric calculator" being able to play a game. For me, imagining the Pocket Computer doing an inventory is difficult. It's all possible, though. We may have to sacrifice in some applications, but we'll reach our major goals if our imagination will let us.

The Pocket Computer has only 1.9K bytes of Random Access Memory. That means that a program usually cannot be over forty lines long. However, the BASIC which the Pocket Computer uses in programming incorporates a "chain" command which is more simple and easy to use than the "chain" commands found in most larger systems. By "chain"ing we can write programs in segments that would, when put together, make a 64K computer blush. Imagine that!

The Pocket Computer's BASIC is missing a few common functions. There is no provision for "DATA" or "READ" instructions. There is no "INSTRING" capability for looking into and taking apart the contents of a String Variable. There is no provision for "PEEK"ing or "POKE"ing into the Machine Language of the Pocket Computer, or using Assembly Language with it. There are other limitations, too. They all make the challenge more exciting!

My first programming experience was on the Pocket Computer. Later I got a computer with more capabilities. I like my larger system a great deal, but I always come back to the Pocket Computer. It's like recreation. I guess I like to move mountains.

There are two interfaces available for the Pocket Computer. One is the Cassette Interface, which allows programs and data to be saved on a Cassette Recorder for storage and later loading back into the Pocket Computer. The other is the Printer/Cassette Interface, which is a Cassette Interface and Printer for producing hard copy of program runs and listings. Most of the programs in this series utilize and are formatted for use either with or without the Printer/Cassette Interface and a Cassette Recorder for program and data loading and storing. If you have these peripherals, you can get them, they very definitely will broaden the scope of applications for the Pocket Computer. They, and the programs in this series, make up a few of the unlimited number of Pocket Operating Systems.

Now, imagine with me a program . . .

THE WORD PROCESSING CHALLENGE

Perhaps the most useful and popular type of program used in real world systems today is the word processor. They are used for many needs, from producing many letter-quality form letters, to developing manuscripts into books. Usually word processors come as a package, which completely take over the computer functions, turning it into a page oriented editing device for producing final copy suitable for whatever application the user has in mind. Available in many packages are spelling editors, grammar editors, and even editors that will go through the entire manuscript and find cliches. Most word processors are expected to at least be able to justify lines to the left and right margins without hyphenating any words, thus making very presentable finished copy.

By definition, a word processor is a computer-based operating system that can take raw manuscript and transform it into finished copy. That implies, but does not require, that editing functions may exist within the package. Also not mentioned in the definition is the disk system which normally would be involved, allowing speedy access of different programs and data as needed. There are many other functions that are normally associated with word processors, and in the future, there will undoubtedly be many more. It is in the nature of the computer industry itself to continue advancing the capabilities of operating systems, in order to make them more competitive on the market.

Obviously, with all that can be involved with word processors, it makes it difficult to talk about them and the Pocket Computer in the same sentence. However, if we stick to our definition, there are some programs that can allow the Pocket Computer to do some word processing functions. Whether you want to call them word processors or not is up to you.

The two programs we will be working with in this article both require that the user have some knowledge of what the Pocket Computer can do with string variables. The user must, when asked for input by the computer, enter the copy seven (or less) characters at a time. This means that an example sentence like "Mary had a little lamb." could be entered like this:

```
(USER) MARY HA _ <ENTER>
      D A LIT _ <ENTER>
      TLE LAM _ <ENTER>
      B. _ <ENTER>
```

The " _ " is what the computer does.

There's another complication, though. If you are using the Printer Interface, as normally would be run with these programs, the printed copy is limited to a 16 character per line format. Since our maximum string length is 7 characters, we can either limit our output (and input) to two strings (14 characters), or to three strings per line, and limit the third string to only two characters. With the second method, the capacity of each line is greater, that being 16 characters. Because of the greater capacity for line length, this is the method that is used in both Pocket Typewriter, and Pocket Processor. Here is an example of how to enter copy into these systems:

When the "?" appears—

```
(USER) MARY HA _ <ENTER> OUTPUT:
      D A LIT _ <ENTER>
      - _ <ENTER> MARY HAD A LIT-
      TLE LAM _ <ENTER>
      B WHOSE _ <ENTER>
      - <ENTER> TLE LAMB WHOSE
      FLEECE _ <ENTER>
      WAS AS _ <ENTER>
      - <ENTER> FLEECE WAS AS
      WHITE A _ <ENTER>
      S SNOW. _ <ENTER>
      - <ENTER> WHITE AS SNOW.
```

It takes a little practice to get used to, but the two extra characters will be worth it. Just remember that if you enter any more than two characters into the third string of any given line, the printed copy of that line will scroll off into the space that the next line should be in. Lines like that can be corrected, but they take some time and effort.

The main differences between Pocket Typewriter and Pocket Processor are in the way they deal with data storage. Pocket Typewriter fills the computer memory first if necessary, then upon the user's command, stores the data on tape for later use. Since the data remains within the computer, individual strings can be called up (manually) for correction and editing. Pocket Processor, on the other hand, enters each line into tape storage upon completion, avoiding the need to watch for overloading the computer. Editing on this system can be accomplished either before the lines are stored on tape, or after.

You've probably noticed that automatic editing has not been mentioned yet. Nor has justification of the margins, or a

dictionary function. That is because our little computer cannot handle the string manipulation necessary to do these things. But, for what it can do, there's nothing to be ashamed of. The main purposes of these programs are to take in the user's manuscript, provide editing opportunities, make a hard copy, and store the data in an accessible form for later use. The computer can handle these functions very well. The user, then, must handle the editing.

These programs are both fun to use for notes, or letters, etc. I had even considered using Pocket Processor in the development of this series, as it could provide the finished copy (in 16 character per line format) in a suitable format for printing. As you will see, though, this would take a great deal of time. But it was sort of fun just imagining it . . .

POCKET TYPEWRITER

Sample Run:

```
(PAUSE) POCKET TYPEWRITER
C/1981 BY DON HERGERT
TO ENTER SHFT A
TO PRINT SHFT B
TO LOAD DATA SHFT C
TO SAVE DATA SHFT D
TO CLEAR SHFT F
(USER) SHFT A
```

```
(THE
PROGRAM
NOW ASKS
FOR DATA,
STORES IT
INTO THE
COMPUTER'S
MEMORY,
AND PRINTS
IT ON THE
PRINTER.)
```

```
THIS IS A GOOD
EXAMPLE OF HOW
THIS PROGRAM
WORKS.
I CAN ENTER ANY
THING I WANT, AS
LONG AS I KEEP
IN MIND THAT ALL
THE WORDS THAT I
ENTER MUST BE
ENTERED SEVEN
CHARACTERS AT A
TIME.
THE MISTAKES
CAN BE CORRECTED
BY MANUALLY
CALLING UP THE
STRING VARIABLE
AND CHANGING THE
CONTENTS.
THAT IS WHAT I
AM GOING TO DO
ABOUT THE CRAZZY
MISTOOKS IN THIS
LINE.
```

```
(USER) * <ENTER> * <ENTER> * <ENTER> A <ENTER>
(PRINT TO DISPLAY) 103
(USER) A$(98) <ENTER>
(PRINT) MISTOOK
(USER) A$(98) = "MISTAKE" <ENTER>
(PRINT) MISTAKE
(USER CAN CONTINUE CORRECTIONS IN THIS
MANNER)
(USER) SHFT D (AFTER PREPARING CASSETTE)
(COMPUTER NOW RECORDS DATA ON TAPE)
(USER) SHFT F
(COMPUTER NOW CLEARS ALL VARIABLES)
(USER) SHFT C (AFTER PREPARING CASSETTE)
```

```
(COMPUTER NOW LOADS DATA FROM TAPE)
(USER) SHFT B
(INPUT) DATE _
(USER) JAN. 19 <ENTER>, 1982 <ENTER>
(INPUT) NAME
(USER) WATCH T <ENTER> HIS: <ENTER>
(PRINT)
```

```
JAN. 19, 1982
WATCH THIS:
THIS IS A GOOD
EXAMPLE OF HOW
THIS PROGRAM
WORKS.
I CAN ENTER ANY
THING I WANT, AS
LONG AS I KEEP
IN MIND THAT ALL
THE WORDS THAT I
ENTER MUST BE
ENTERED SEVEN
CHARACTERS AT A
TIME.
THE MISTAKES
CAN BE CORRECTED
BY MANUALLY
CALLING UP THE
STRING VARIABLE
AND CHANGING THE
CONTENTS.
THAT IS WHAT I
AM GOING TO DO
ABOUT THE CRAZY
MISTAKES IN THIS
LINE.
```

Probably even if we could handle justification of the margins, the lines would still look a little strange, because we can only get so many words into a 16 character line. For this system, it might be more trouble than it's worth.

As mentioned earlier, the BASIC that resides within the Pocket Computer has limited string handling capabilities. You may want to further explore or experiment in order to get an idea of exactly what those limitations are. It is possible to do some word searching, and even editing, but you will find that the more you force into this program, the less practical it will become.

In the sample run, the user first keys in <SHFT>, and <SPC>. The computer then presents the introduction and menu. In order to enter new copy, the user keys in <SHFT> <A>, and upon the appearance of the cursor, enters the data for processing 7 characters at a time for the first two inputs, and a maximum of 2 characters at a time for the third input of each line. At the end of the third input, the computer prints the line, and asks for the first input of the next line. When the user has completed the final line, an asterisk is entered for the next three inputs for what would be the next line. The computer reads these as an escape command and presents the menu again.

After the escape from the data entering subroutine, the user in this example keys in an <A> and <ENTER>. The variable A in this program is holding the number of the last dimensioned string variable that was used. In the sample that number was 103. The user then stepped back through the contents of strings A\$(103) to A\$(98), where the first correction was made by simply keying in manually the string name

and what it should contain, and then entering it. The remainder of the sample run is self-explanatory.

Although it is limited by the amount of memory available for strings (enough for about 40 lines), the convenience of not needing to be connected to a cassette recorder in order to print is welcomed. For a note or letter, the copy can be printed as often as the user wants, using as many headings or greetings as wanted. If it is a manuscript that the user wants to re-use later, the data can be saved on tape, and loaded at will. If the user wants, manuscript can be longer than normal memory space would allow. This is accomplished by setting up multiple files, and simply calling one file after the other as needed, thus making it possible to have pre-recorded letters or notes that could have interchangeable paragraphs for use in different situations.

Program Listing:

```
1 " "POCKET TYPEWRITER"
: PAUSE "C/1981 BY DON HERGERT"
3 PAUSE "TO ENTER SHFT A"
: PAUSE "TO PRINT SHFT B"
: PAUSE "TO LOAD DATA SHFT C"
4 PAUSE "TO SAVE DATA SHFT D"
: PAUSE "TO CLEAR SHFT F"
: END
5 "F"
: CLEAR
: END
10 "A"
: A=26
20 B=A+1
: C=B+1
: D=C+1
30 INPUT A$(B),A$(C)
: BEEP 1
: INPUT A$(D)
35 IF A$(B)="*" IF A$(C)="*" IF A$(D)="*" LET
A$(B)=" "
: A$(C)=" "
: A$(D)=" "
: GOTO 3
40 PRINT A$(B);A$(C);A$(D)
50 A=D
: GOTO 20
60 "B"
: INPUT "DATE",E$,F$, "NAME",G$,H$
: PRINT E$;F$
: PRINT G$;H$
: FOR A=27 TO D STEP 3
: B=A+1
: C=B+1
70 PRINT A$(A);A$(B);A$(C)
80 NEXT A
90 GOTO 3
100 "C"
: INPUT #"DATA"
: GOTO 3
150 "D"
: PRINT #"DATA"
: GOTO 3
```

Now let's move on to the second program.

POCKET PROCESSOR

```
Sample Run:
(PAUSE) POCKET PROCESSOR
C/1981 BY DON HERGERT
TO ENTER PRESS SHFT A
FOR COPIES PRESS SHFT B
(USER) SHFT A
(INPUT) TITLE: _
(USER) SAMPLE (ENTER)
```


(PAUSE) COPY:

(INPUT) ?

(USER) THIS IS A GOOD

(INPUT) ANY CORRECTIONS? (Y/N) _

(USER) Y (ENTER)

(PAUSE) COPY:

(INPUT) ?

(USER) THIS IS A GOOD
(HERE THE EXAMPLE OF WHAT
PROGRAM THIS PROGRAM CAN
MAKES THE DO.
COMPUTER IT IS NOT LIMIT-
ASK THE ED TO THE AMOUNT
USER FOR MEMORY AVAILABLE
COPY, IN THE COMPUTER.
PRINT THAT THE ONLY LIMIT-
COPY, ASK ATION IS IN HOW
IF THERE LONG YOUR TAPE
ARE ANY IS.

CORRECTIONS, AND IF NOT, RECORD THAT LINE ON TAPE. IF THERE ARE CORRECTIONS, THE PROGRAM WILL ASK FOR THE CORRECT LINE, AND AFTER GETTING IT, PRINT, VERIFY, AND RECORD IT. THE USER MUST AFTER THE LAST LINE TO BE RECORDED ENTER A SPACE FOR EACH OF THE NEXT THREE ?S, OR THE COMPUTER WILL, WHEN SEARCHING THE TAPE FOR DATA, NEVER KNOW WHEN TO STOP.)

(USER) SHFT B

(INPUT) TITLE TO FIND: _

(USER) SAMPLE (ENTER)

(PRINT)

THIS IS A GOOD
EXAMPLE OF WHAT
THIS PROGRAM CAN
DO.
IT IS NOT LIMIT-
ED TO THE AMOUNT
MEMORY AVAILABLE
IN THE COMPUTER.
THE ONLY LIMIT-
ATION IS IN HOW
LONG YOUR TAPE
IS.

(END)

This sample run is fairly self-explanatory. The program has only two modes, those being a mode for data entering, and a mode for calling and printing data. Since this program immediately records the edited data onto tape, there is no need to be concerned about remaining memory space when in the entry mode. As mentioned in the sample run, the user must at the completion of the final line, key in (SPC) and (ENTER) for the next three inputs. This not only serves as a key to a conditional escape routine, but it also provides a pointer to the program when in the calling and printing mode the computer reaches the end of the current data file on tape.

This program does an interesting thing as it loads data from tape for eventual printing. It searches from file to file for the lines that carry the title that the user has instructed it to find, and as it does this, it will skip over those lines that do not have the correct title. This means, among other things, that data blocks for different manuscripts can be mixed without mixing up the hard copy.

This program can provide some good examples of the nature of data storage on tape. One of the exciting possibilities of this system is editing already existing tape files. Pro-

vided that care is taken to not crash surrounding files, editing can be done by simply recording over the line (file) that is to be changed (or deleted), using the data entry mode of the program. If you are careful, you can even add a line (or two if you are tricky), by recording the new line(s) on the leading 6-second non-signal beep that normally precedes each data block on tape. This takes practice, so don't try it on an important file until you know you can do it without crashing the surrounding lines. The ability to do this with this system makes editing possible on two levels, and generally much easier and simpler than with Pocket Typewriter, thus making Pocket Processor the choice for heavy duty usage.

Program Listing:

```
10 " "PAUSE "POCKET PROCESSOR"  
: PAUSE " C/1981 BY DON HERGERT  
20 CLEAR  
: BEEP 5  
: PAUSE "TO ENTER PRESS SHFT A"  
: PAUSE "FOR COPIES PRESS SHFT B"  
: END  
100 "A"INPUT "TITLE: ";A$(142)  
110 PAUSE "COPY: "  
: INPUT A$(143),A$(144)  
: BEEP 1  
: INPUT A$(145)  
120 PRINT A$(143);A$(144);A$(145)  
130 INPUT "ANY CORRECTIONS? (Y/N) ";C$  
: IF C$="Y" THEN 110  
140 PRINT #"TITLE";A$(142)  
150 IF A$(143)=" "IF A$(144)=" "IF A$(145)=" "THEN 20  
160 GOTO 110  
200 "B"INPUT "TITLE TO FIND: ";T$  
210 INPUT #"TITLE";A$(142)  
: IF T$=A$(142)THEN 230  
220 GOTO 210  
230 IF A$(143)=" "IF A$(144)=" "IF A$(145)=" "THEN 20  
240 PRINT A$(143);A$(144);A$(145)  
: GOTO 210
```

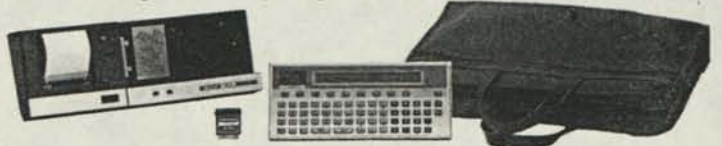
SUMMARY

The key to any successful program is convenience. One of the main reasons that a program or system might fail on the market is the lack of this convenience, or "User Friendliness".

Convenience doesn't just include the functions that a program might do. Speed is an important factor, as are ease of operation and clarity of the output. Any of these attributes require memory space, though, and in our little computer, that is what is missing. Consequently function by function we have to go through a perspective program and decide what is important enough to keep, and then file the remaining ideas for use in another time and place.

This is what was done with these two programs, and in fact most of the programs in this series. It's the challenge that we've been discussing and working with. It is such a thrill to see a Pocket Computer at work on some of these projects that no matter how many conveniences we need to give up, the outcome is worth it.

Pocket Typewriter and Pocket Processor are practical programs. It is difficult in some ways to equate these programs with their real world counterparts. Large word processing systems definitely eclipse these little systems, but for what we have to work with, these programs are fairly impressive. Let's go on imagining from here . . .



Circle Subroutine For PC-2

Here is a subroutine that you might want to use in some of your graphics programs. It will draw a circle using a set of parameters that you can specify. The parameters are as follows:

X,Y coordinate—the coordinate of the centerpoint of the circle

Radius—the radius of the circle (1-108)

Color—the color of the circle (0-3)

Height/Width Ratio—This parameter gives you the ability to make ellipses. A value of 1 will make the circle a "perfect" circle. A value from 2 to 99 will make the circle range from a near "perfect" circle (99) to a vertical line (2). A value from .01 to .99 will make the circle range from a near "perfect" circle (.99) to a horizontal line (.01).

Start and End Points—These parameters will allow you to make arcs. Both values must be between 1 and 360 (inclusive). Also, you should take note that the circle will always be drawn counterclockwise.

Density—The density is the number of lines that will be used to draw the circle. A value of 3 would actually give you a triangle; a value of 4, a square; a value of 5, a pentagon, etc. The value can range from 2 to ????. However, use 20 to 100 for a circle.

Additional Note: You must be in GRAPH mode and already have your origin set before calling this subroutine.

```

100:GRAPH
110:LPRINT "      ";
120:SORGN
130:INPUT "X COORDINATE: ";X
135:IF X<-107 OR X>107 GOSUB 300
      :GOTO 130
140:INPUT "Y COORDINATE: ";Y
145:IF Y<-107 OR Y>107 GOSUB 300
      :GOTO 140
150:INPUT "RADIUS: ";R
160:INPUT "COLOR: ";C
164:C=INT C
165:IF C<0 OR C>3 GOSUB 300
      :GOTO 160
170:INPUT "HEIGHT/WIDTH RATIO: ";H
171:H=ABS H
172:IF H<1 LET H=INT (H*100)/100
      :GOTO 180
175:H=INT H
180:INPUT "START POINT: ";S
184:S=INT S
185:IF S<0 OR S>360 GOSUB 300
      :GOTO 180
190:INPUT "END POINT: ";E
194:E=INT E
195:IF E<0 OR E>360 GOSUB 300
      :GOTO 190
200:INPUT "DENSITY: ";D
210:IF D<2 GOSUB 300
      :GOTO 200
220:GOSUB 200000
240:GOTO 130
300:PAUSE "INPUT ERROR! - TRY AGAIN"
      :RETURN
500:REM .....
600:REM .....
10000:REM CIRCLE SUBROUTINE
10010:REM X = X COORDINATE
10020:REM Y = Y COORDINATE

```

```

10030:REM R = RADIUS
10040:REM C = COLOR
10050:REM H = HEIGHT/WIDTH RATIO
10060:REM S = START POINT
10070:REM E = END POINT
10080:REM D = DENSITY
20000:DEGREE
20005:IF S>=ELET E=E+360
20030:IF H=1LET XM=1
      :YM=1
20040:IF H>1LET XM=H/100
      :YM=1
20050:IF H<1LET XM=1
      :YM=H
20052:OX=X+COS (S)*R*XM
      :LX=OX
20057:OY=Y+SIN (S)*R*YM
      :LY=OY
20060:Z=360/D
20070:FOR I=STO E+1STEP Z
20075:J=I
      :IF I>360 LET J=I-360
20080:XC=X+COS J*R*XM
20090:YC=Y+SIN J*R*YM
20100:LINE (OX,OY)-(XC,YC),0,C
20110:OX=XC
      :OY=YC
20120:NEXT I
20140:RETURN

```

Quick Printer Graph Routine


Greg Patek
Route 1 Box 309
Katy, TX 77449

I own a TRS-80 Model I 16K Level II with the Quick Printer (26-1153). The program that I wrote will let the printer do "Character Graphics" on the printer line by line (a total of 16 lines). The program can be used as a subroutine in your program, or the program can store the drawing in memory and print it at a later time. It's up to you. I use this program with a program that draws a graph on the screen and then the printer types it on the line printer.

```

1000 REM * PRINTING ROUTINE *
1010 FOR S=15360 TO 16383
      : P=PEEK(S)
      : IF P=32 OR P=128 THEN LPRINT " ";
      : POKE S, 32
      : ELSE LPRINT "#";
      : POKE S, 35
1020 T=T+1
      : IF T>63 THEN LPRINT CHR$(13)
      : T=0
1030 NEXT S
1040 FOR H=1 TO 4
      : LPRINT " "
      : NEXT H
1050 RETURN

```

I use this routine a lot with all of my graphic drawings and other graphic designs. 

Christmas Tree Letterhead

Art Voisard
1310 Cornwall
Ocean Springs, MS 39564

I continue to marvel at the 'classic' programs some of the readers send, including their Christmas efforts.

I thought your followers might enjoy (and wish to modify) a program I wrote to put a letterhead on the notes we include with Christmas cards (using my Line Printer VII, of course). It does not contain mathematical wizardry, but I am proud of the result it achieves.

How about devoting more space to the Color Computer . . . after all, that is the playful one, isn't it?

```

2 '-----> XMAS TREE LETTERHEAD
4 '-----> ART VOISARD
6 '-----> 1310 CORNWALL
8 '-----> OCEAN SPRINGS, MS 39564
10 '-----> WRITTEN 1981
15 PCLEAR 4
20 PMODE 4,1
25 PCLS 0
30 SCREEN 1,0
32 PSET (128,4)
35 PSET (128,8)
   : PSET (126,10)
   : PSET (130,10)
   : PSET (128,6)
40 FOR X = 124 TO 132 STEP 4
   : PSET (X,12)
   : NEXT X
45 FOR X = 126 TO 130 STEP 4
   : PSET (X,14)
   : NEXT X
50 FOR X = 124 TO 132 STEP 4
   : PSET (X,16)
   : NEXT X
55 FOR X = 122 TO 134 STEP 4
   : PSET (X,18)
   : NEXT X
60 FOR X = 120 TO 136 STEP 4
   : PSET (X,20)
   : NEXT X
65 FOR X = 122 TO 134 STEP 4
   : PSET (X,22)
   : NEXT X
70 FOR X = 118 TO 138 STEP 4
   : PSET (X,24)
   : NEXT X
75 FOR X = 122 TO 138 STEP 4
   : PSET (X,26)
   : NEXT X
80 FOR X = 120 TO 136 STEP 4
   : PSET (X,28)
   : NEXT X
85 FOR X = 118 TO 138 STEP 4
   : PSET (X,30)
   : NEXT X
90 FOR X = 114 TO 142 STEP 4
   : PSET (X,32)
   : NEXT X
95 FOR X = 118 TO 142 STEP 4
   : PSET (X,34)
   : NEXT X

```

```

100 FOR X = 116 TO 138 STEP 4
   : PSET (X,36)
   : NEXT X
105 FOR X = 112 TO 140 STEP 4
   : PSET (X,38)
   : NEXT X
110 FOR X = 114 TO 144 STEP 4
   : PSET (X,40)
   : NEXT X
115 FOR X = 116 TO 142 STEP 4
   : PSET (X,42)
   : NEXT X
120 FOR X = 114 TO 144 STEP 4
   : PSET (X,44)
   : NEXT X
125 FOR X = 110 TO 146 STEP 4
   : PSET (X,46)
   : NEXT X
130 FOR X = 112 TO 148 STEP 4
   : PSET (X,48)
   : NEXT X
135 FOR X = 114 TO 144 STEP 4
   : PSET (X,50)
   : NEXT X
140 FOR X = 112 TO 146 STEP 4
   : PSET (X,52)
   : NEXT X
145 FOR X = 110 TO 148 STEP 4
   : PSET (X,54)
   : NEXT X
150 FOR X = 108 TO 150 STEP 4
   : PSET (X,56)
   : NEXT X
155 FOR X = 112 TO 148 STEP 4
   : PSET (X,58)
   : NEXT X
160 FOR X = 110 TO 150 STEP 4
   : PSET (X,60)
   : NEXT X
165 FOR X = 108 TO 152 STEP 4
   : PSET (X,62)
   : NEXT X
170 FOR X = 106 TO 150 STEP 4
   : PSET (X,64)
   : NEXT X
175 FOR X = 104 TO 152 STEP 4
   : PSET (X,66)
   : NEXT X
180 FOR X = 102 TO 154 STEP 4
   : PSET (X,68)
   : NEXT X
185 FOR X = 108 TO 158 STEP 4
   : PSET (X,70)
   : NEXT X
190 FOR X = 106 TO 158 STEP 4
   : PSET (X,72)
   : NEXT X
195 FOR X = 104 TO 154 STEP 4
   : PSET (X,74)
   : NEXT X
200 FOR X = 102 TO 156 STEP 4

```

```

: PSET (X,76)
: NEXT X
205 FOR X = 98 TO 158 STEP 4
: PSET (X,78)
: NEXT X
210 FOR X = 102 TO 156 STEP 4
: PSET (X,80)
: NEXT X
215 FOR X = 100 TO 160 STEP 4
: PSET (X,82)
: NEXT X
220 FOR X = 96 TO 158 STEP 4
: PSET (X,84)
: NEXT X
225 FOR X = 98 TO 160 STEP 4
: PSET (X,86)
: NEXT X
230 FOR X = 94 TO 162 STEP 4
: PSET (X,88)
: NEXT X
235 FOR X = 90 TO 160 STEP 4
: PSET (X,90)
: NEXT X
240 FOR X = 92 TO 158 STEP 4
: PSET (X,92)
: NEXT X
245 FOR X = 94 TO 160 STEP 4
: PSET (X,94)
: NEXT X
250 FOR X = 96 TO 164 STEP 4
: PSET (X,96)
: NEXT X
255 FOR X = 94 TO 162 STEP 4
: PSET (X,98)
: NEXT X
260 FOR X = 88 TO 164 STEP 4
: PSET (X,100)
: NEXT X
265 FOR X = 92 TO 166 STEP 4
: PSET (X,102)
: NEXT X
270 FOR X = 90 TO 162 STEP 4
: PSET (X,104)
: NEXT X
275 FOR X = 88 TO 164 STEP 4
: PSET (X,106)
: NEXT X
280 FOR X = 86 TO 68 STEP 4
: PSET (X,108)
: NEXT X
285 PSET (90,110)
: PSET (138,110)
: PSET (96,112)
290 PSET (94,110)
: PSET (98,110)
295 FOR X = 102 TO 110 STEP 4
: PSET (X,110)
: NEXT X
300 FOR X = 114 TO 134 STEP 4
: PSET (X,110)
: NEXT X
305 FOR X = 142 TO 170 STEP 4
: PSET (X,110)
: NEXT X
310 PSET (104,112)
: PSET (108,112)
315 PSET (116,112)
: PSET (120,112)
320 PSET (124,112)
: PSET (128,112)
: PSET (136,112)
325 PSET (142,112)
: PSET (146,112)
330 PSET (150,112)
: PSET (152,112)
: PSET (156,112)
335 PSET (160,112)
: PSET (164,112)
: PSET (118,114)
340 PSET (144,114)
: PSET (132,112)
345 FOR X = 124 TO 134 STEP 2
350 FOR Y = 114 TO 118 STEP 2
355 PSET (X,Y)
: NEXT Y,X
360 FOR X = 112 TO 146
365 FOR X = 120 TO 126
370 PSET (X,Y)
: NEXT Y,X
400 '=====> MERRY CHRISTMAS
405 '==> M
410 FOR Y = 132 TO 146 STEP 2
: PSET (86,Y)
: NEXT Y
415 PSET (88,134)
: PSET (90,136)
: PSET (92,138)
420 PSET (94,136)
: PSET (96,134)
425 FOR Y = 132 TO 146 STEP 2
: PSET (98,Y)
: NEXT Y
430 '==> E
435 FOR Y = 132 TO 146 STEP 2
: PSET (104,Y)
: NEXT Y
440 FOR X = 106 TO 116 STEP 2
: PSET (X,132)
: NEXT X
445 PSET (106,140)
: PSET (110,140)
: PSET (108,140)
450 FOR X = 106 TO 116 STEP 2
: PSET (X,146)
: NEXT X
455 '==> R
460 FOR X = 122 TO 132 STEP 2
: PSET (X,132)
: NEXT X
465 FOR Y = 134 TO 146 STEP 2
: PSET (122,Y)
: NEXT Y
470 FOR X = 124 TO 132 STEP 2
: PSET (X,140)
: NEXT X
475 FOR Y = 134 TO 138 STEP 2
: PSET (134,Y)
: NEXT Y
480 PSET (130,142)
: PSET (132,144)
: PSET (134,146)
485 '==> R
490 FOR X = 140 TO 150 STEP 2
: PSET (X,132)
: NEXT X
495 FOR Y = 134 TO 146 STEP 2
: PSET (140,Y)
: NEXT Y
500 FOR X = 142 TO 150 STEP 2
: PSET (X,140)
: NEXT X
505 FOR Y = 134 TO 138 STEP 2
: PSET (152,Y)
: NEXT Y
510 PSET (148,142)
: PSET (150,144)
: PSET (152,146)
515 '==> Y
520 PSET (158,132)
: PSET (170,132)
: PSET (160,134)
: PSET (168,134)
525 PSET (162,136)
: PSET (166,136)
530 FOR Y = 138 TO 146 STEP 2
: PSET (164,Y)
: NEXT Y
535 '==> C
540 FOR Y = 154 TO 164 STEP 2
: PSET (56,Y)
: NEXT Y
545 FOR X = 58 TO 68 STEP 2
: PSET (X,152)
: NEXT X
550 FOR X = 58 TO 68 STEP 2
: PSET (X,166)
: NEXT X
555 '==> H
560 FOR Y = 152 TO 166 STEP 2
: PSET (74,Y)
: NEXT Y
565 FOR X = 76 TO 84 STEP 2
: PSET (X,160)
: NEXT X
570 FOR Y = 152 TO 166 STEP 2
: PSET (86,Y)
: NEXT Y
575 '==> R
580 FOR Y = 152 TO 166 STEP 2
: PSET (92,Y)
: NEXT Y
585 FOR X = 94 TO 102 STEP 2
: PSET (X,152)
: NEXT X
590 FOR X = 94 TO 102 STEP 2
: PSET (X,160)
: NEXT X
595 FOR Y = 154 TO 158 STEP 2
: PSET (104,Y)
: NEXT Y
600 PSET (100,162)
: PSET (102,164)
: PSET (104,166)
605 '==> I
610 FOR Y = 152 TO 166 STEP 2
: PSET (110,Y)
: NEXT Y
615 '==> S
620 FOR X = 118 TO 128 STEP 2
: PSET (X,152)
: NEXT X
625 PSET (116,154)
: PSET (116,156)
: PSET (128,160)
630 FOR X = 118 TO 126 STEP 2
: PSET (X,158)
: NEXT X
635 PSET (128,160)
: PSET (128,162)
: PSET (128,164)
640 FOR X = 116 TO 126 STEP 2
: PSET (X,166)
: NEXT X
645 '==> T
650 FOR X = 134 TO 146 STEP 2
: PSET (X,152)
: NEXT X
655 FOR Y = 154 TO 166 STEP 2
: PSET (140,Y)
: NEXT Y
660 '==> M
665 FOR Y = 152 TO 166 STEP 2
: PSET (152,Y)
: NEXT Y
670 FOR Y = 152 TO 166 STEP 2
: PSET (164,Y)
: NEXT Y
675 PSET (154,154)

```

```

: PSET(156,156)
: PSET(158,158)
680 PSET(160,156)
: PSET(162,154)
685 '==> A
690 FOR X = 172 TO 180 STEP 2
: PSET(X,152)
: NEXT X
695 FOR Y = 154 TO 166 STEP 2
: PSET(170,Y)
: NEXT Y
700 FOR Y = 154 TO 166 STEP 2
: PSET(182,Y)
: NEXT Y
705 FOR X = 172 TO 180 STEP 2
: PSET(X,160)
: NEXT X
710 '==> S
715 FOR X = 190 TO 200 STEP 2
: PSET(X,152)
: NEXT X
720 FOR X = 190 TO 198 STEP 2
: PSET(X,158)
: NEXT X
725 FOR X = 188 TO 198 STEP 2
: PSET(X,166)
: NEXT X
730 PSET(188,154)
: PSET(188,156)
735 FOR Y = 160 TO 164 STEP 2
: PSET(200,Y)
: NEXT Y
740 '==> F
745 FOR X = 32 TO 40 STEP 2
: PSET(X,172)
: NEXT X
750 FOR Y = 174 TO 182 STEP 2
: PSET(32,Y)
: NEXT Y
755 PSET(34,178)
: PSET(36,178)
760 '==> R
765 FOR X = 44 TO 50 STEP 2
: PSET(X,172)
: NEXT X
770 FOR X = 46 TO 50 STEP 2
: PSET(X,178)
: NEXT X
775 FOR Y = 174 TO 182 STEP 2
: PSET(44,Y)
: NEXT Y
780 PSET(52,174)
: PSET(52,176)
: PSET(50,180)
785 PSET(52,182)
790 '==> O
795 FOR X = 58 TO 62 STEP 2
: PSET(X,172)
: NEXT X
800 FOR X = 58 TO 62 STEP 2
: PSET(X,182)
: NEXT X
805 FOR Y = 174 TO 180 STEP 2
: PSET(56,Y)
: NEXT Y
810 FOR Y = 174 TO 180 STEP 2
: PSET(64,Y)
: NEXT Y
815 '==> M
820 FOR Y = 172 TO 182 STEP 2
: PSET(68,Y)
: NEXT Y
825 FOR Y = 172 TO 182 STEP 2
: PSET(76,Y)
: NEXT Y
830 PSET(70,174)
: PSET(72,176)
: PSET(74,174)
835 '==> T
840 FOR X = 88 TO 96 STEP 2
: PSET(X,172)
: NEXT X
845 FOR Y = 174 TO 182 STEP 2
: PSET(92,Y)
: NEXT Y
850 '==> H
855 FOR Y = 172 TO 182 STEP 2
: PSET(100,Y)
: NEXT Y
860 FOR Y = 172 TO 182 STEP 2
: PSET(108,Y)
: NEXT Y
865 FOR X = 102 TO 106 STEP 2
: PSET(X,178)
: NEXT X
870 '==> E
875 FOR X = 112 TO 120 STEP 2
: PSET(X,172)
: NEXT X
880 FOR X = 112 TO 120 STEP 2
: PSET(X,182)
: NEXT X
885 FOR X = 114 TO 118 STEP 2
: PSET(X,178)
: NEXT X
890 FOR Y = 174 TO 182 STEP 2
: PSET(112,Y)
: NEXT Y
895 '==> V
900 FOR Y = 172 TO 176 STEP 2
: PSET(134,Y)
: NEXT Y
905 FOR Y = 172 TO 176 STEP 2
: PSET(142,Y)
: NEXT Y
910 PSET(136,178)
: PSET(136,180)
: PSET(140,178)
: PSET(140,180)
: PSET(138,182)
: PSET(138,180)
915 '==> O
920 FOR X = 148 TO 152 STEP 2
: PSET(X,172)
: NEXT X
925 FOR X = 148 TO 152 STEP 2
: PSET(X,182)
: NEXT X
930 FOR Y = 174 TO 180 STEP 2
: PSET(146,Y)
: NEXT Y
935 FOR Y = 174 TO 180 STEP 2
: PSET(154,Y)
: NEXT Y
940 '==> I
945 FOR Y = 172 TO 182 STEP 2
: PSET(158,Y)
: NEXT Y
950 '==> S
955 FOR X = 164 TO 170 STEP 2
: PSET(X,172)
: NEXT X
960 FOR X = 164 TO 168 STEP 2
: PSET(X,176)
: NEXT X
965 FOR X = 162 TO 168 STEP 2
: PSET(X,182)
: NEXT X
970 PSET(162,174)
: PSET(170,178)
: PSET(170,180)
975 '==> A
980 FOR Y = 174 TO 182 STEP 2
: PSET(174,Y)
: NEXT Y
985 FOR Y = 174 TO 182 STEP 2
: PSET(182,X)
: NEXT Y
990 FOR X = 176 TO 180 STEP 2
: PSET(X,172)
: NEXT X
995 FOR X = 176 TO 180 STEP 2
: PSET(X,178)
: NEXT X
1000 '==> R
1005 FOR X = 186 TO 192 STEP 2
: PSET(X,172)
: NEXT X
1010 FOR X = 188 TO 192 STEP 2
: PSET(X,178)
: NEXT X
1015 FOR Y = 174 TO 182 STEP 2
: PSET(186,Y)
: NEXT Y
1020 PSET(194,174)
: PSET(194,176)
: PSET(192,180)
: PSET(194,182)
1025 '==> D
1030 FOR X = 200 TO 204 STEP 2
: PSET(X,172)
: NEXT X
1035 FOR X = 200 TO 204 STEP 2
: PSET(X,182)
: NEXT X
1040 FOR Y = 172 TO 182 STEP 2
: PSET(198,Y)
: NEXT Y
1045 FOR Y = 174 TO 180 STEP 2
: PSET(206,Y)
: NEXT Y
1060 '==> S
1065 FOR X = 212 TO 218 STEP 2
: PSET(X,172)
: NEXT X
1070 FOR X = 212 TO 216 STEP 2
: PSET(X,176)
: NEXT X
1075 FOR X = 210 TO 216 STEP 2
: PSET(X,182)
: NEXT X
1080 PSET(210,174)
: PSET(218,178)
: PSET(218,180)
1085 FOR X = 128 TO 130
1090 FOR Y = 76 TO 78
1095 PSET(X,Y)
: NEXT Y,X
1100 FOR Y = 42 TO 50
: PSET(126,Y)
: NEXT Y
1105 FOR Y = 4 TO 12
: PSET(128,Y)
: NEXT Y
1110 FOR Y = 16 TO 24
: PSET(118,Y)
: NEXT Y
1115 FOR Y = 10 TO 18
: PSET(134,Y)
: NEXT Y
1120 FOR Y = 24 TO 32
: PSET(130,Y)
: NEXT Y
1125 FOR Y = 36 TO 44
: PSET(110,Y)
: NEXT Y
1130 FOR Y = 32 TO 40
: PSET(144,Y)

```

```

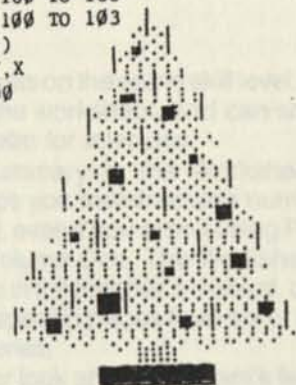
: NEXT Y
1135 FOR Y = 42 TO 50
: PSET(126,Y)
: NEXT Y
1140 FOR Y = 58 TO 66
: PSET(102,Y)
: NEXT Y
1145 FOR Y = 54 TO 62
: PSET(152,Y)
: NEXT Y
1150 FOR Y = 62 TO 70
: PSET(132,Y)
: NEXT Y
1155 FOR Y = 72 TO 80
: PSET(112,Y)
: NEXT Y
1160 FOR Y = 82 TO 90
: PSET(92,Y)
: NEXT Y
1165 FOR Y = 74 TO 82
: PSET(160,Y)
: NEXT Y
1170 FOR Y = 76 TO 84
: PSET(142,Y)
: NEXT Y
1175 FOR Y = 92 TO 100
: PSET(104,Y)
: NEXT Y
1180 FOR Y = 100 TO 108
: PSET(86,Y)
: NEXT Y
1190 FOR Y = 94 TO 102
: PSET(120,Y)
: NEXT Y
1195 FOR Y = 100 TO 108
: PSET(146,Y)
: NEXT Y
1200 FOR Y = 100 TO 108
: PSET(168,Y)
: NEXT Y
1205 FOR X = 124 TO 126
1210 FOR Y = 16 TO 18
1215 PSET(X,Y)
: NEXT Y,X
1220 FOR X = 120 TO 124
1225 FOR Y = 32 TO 34
1230 PSET(X,Y)
: NEXT Y,X
1235 FOR X = 132 TO 135
1240 FOR Y = 36 TO 40
1245 PSET(X,Y)
: NEXT Y,X
1250 FOR X = 116 TO 120
1255 FOR Y = 56 TO 60
1260 PSET(X,Y)
: NEXT Y,X
1265 FOR X = 140 TO 142
1270 FOR Y = 52 TO 54
1275 PSET(X,Y)
: NEXT Y,X
1280 FOR X = 148 TO 152
1285 FOR Y = 68 TO 72
1290 PSET(X,Y)
: NEXT Y,X
1295 FOR X = 104 TO 106
1300 FOR Y = 76 TO 78
1305 PSET(X,Y)
: NEXT Y,X
1310 FOR X = 112 TO 118
1315 FOR Y = 96 TO 102
1320 PSET(X,Y)
: NEXT Y,X
1325 FOR X = 96 TO 100
1330 FOR Y = 104 TO 108
1335 PSET(X,Y)
: NEXT Y,X

```

```

1340 FOR X = 132 TO 134
1345 FOR Y = 88 TO 90
1350 PSET(X,Y)
: NEXT Y,X
1355 FOR X = 136 TO 139
1360 FOR Y = 104 TO 107
1365 PSET(X,Y)
: NEXT Y,X
1370 FOR X = 152 TO 156
1375 FOR X = 84 TO 88
1380 PSET(X,Y)
: NEXT Y,X
1385 FOR X = 160 TO 163
1390 FOR Y = 100 TO 103
1395 PSET(X,Y)
: NEXT Y,X
1400 GOTO 1400

```



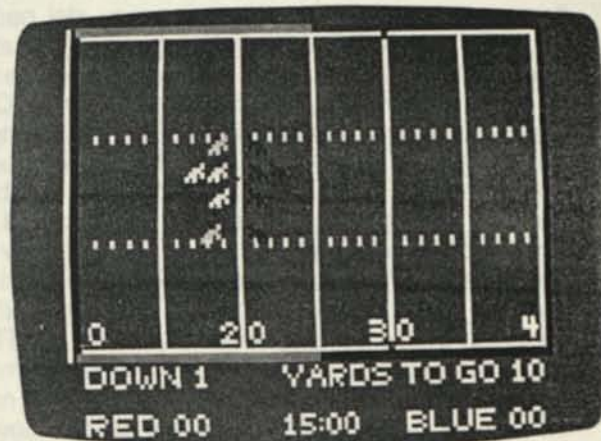
MERRY
CHRISTMAS
FROM THE VOISARDS

Football Program Pak Tip

Matthew Staller
200 Winston Drive
Cliffside Park, NJ 07010

For all you Football PROGRAM PAK owners, you can return a punt. But you have to place the "receiver" right along the sideline. Usually you can "grab" the ball with your head or feet on the upper and lower sidelines respectively. Sometimes the punt will stop next to the sideline and be treated as a "Quick Kick."

Note: I have 32K Extended Color BASIC.



Using Your Printer To Make Math Worksheets For Grades K-8

The Radio Shack K-8 Math Worksheet Generator program lets you use any Radio Shack line printer to print out math worksheets and answer sheets for grades K through 8. In this article, we'll look at the features of this time-saving program; then we'll look at some helpful hints from program user Lou Weiner, a fourth-grade teacher in Saugus, California.

Teachers know that making math worksheets can be time consuming if you use traditional methods. The teacher has to make up the exercises, write or type them onto duplicating masters, and often make corrections on the masters before mimeographing can be done. To give a class two or more worksheets on the same level of difficulty, or to vary the worksheets for different classes or semesters, means that the teacher has to repeat the process from scratch.

The Radio Shack K-8 Math Worksheet Generator program (Cat. No. 26-2162) helps teachers produce worksheets and answer sheets in a fraction of the time it usually takes. And once the teacher has designed a worksheet, the computer can print out any number of different worksheets on the same topic and level of difficulty.

Worksheets can be printed on lesson levels selected from the addition, subtraction, multiplication, and division sequences of the Radio Shack K-8 Math Program, Volume One (Cat. No. 26-1715; or K-8 Math With Student Management, Cat. No. 26-1725). You select the topic(s) for the worksheet, the lesson number(s), and the number of problems from each lesson. The computer then uses these guidelines to randomly generate exercises. For each worksheet generated, an answer sheet is also generated and printed. Special features make the program very versatile. You can choose the total number of worksheets to be printed and can have these worksheets be all alike, or you can have the computer

print different worksheets on the same skill level. You can also design headings for the worksheet, and can save the worksheet design on diskette for later use.

A Curriculum Summary in the Worksheet Generator teacher's manual helps you match lesson numbers with the skills you are teaching, even if you aren't using Radio Shack's K-8 Math Program, Volume One. The Worksheet Generator Cross-Reference, also in the teacher's manual, correlates K-8 lesson numbers with specific pages from six of the major K-8 basal mathematics series.

Let's take a closer look at the program's features. Once Worksheet Generator is loaded and running, you'll see a list of options:

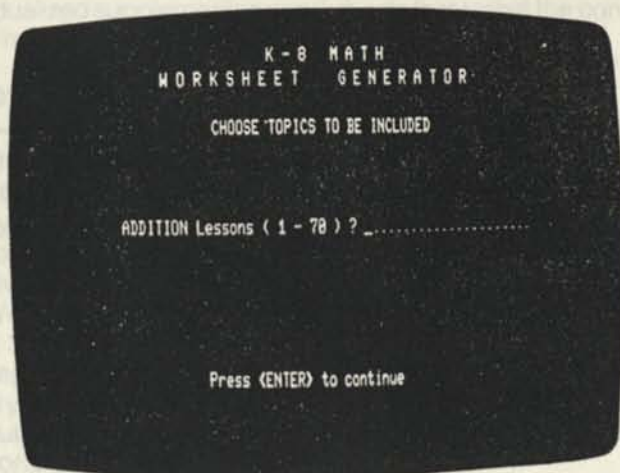


To design a worksheet, you'll need to use Options 2 ("Design headings"), 3 ("Choose topics to be included"), and 4 ("Choose the number of multiple copies"). Option 7 lets you take a look at the overall design of your worksheet. Option 5 then lets you save your worksheet design, and Option 6 lets you print out worksheets. Option 1 lets you use any worksheet design that has been saved on disk (using Option 5). Option 8 ends the program.

Through Option 2, you can enter as many as four headings (for example: NAME, DATE, CLASS, and GRADE). These headings will appear at the top of the worksheet, followed by blanks for the teacher or student to fill in. Entering headings is very easy. When you choose Option 2, some brief directions and a numbered dotted line appear on the screen. To enter a heading, you type the heading on the line and press <ENTER>. When you have entered all the headings you wish, simply press <ENTER> at the dotted line to return to the option list.

When you choose Option 3 ("Choose topics to be included"), you'll see this screen:

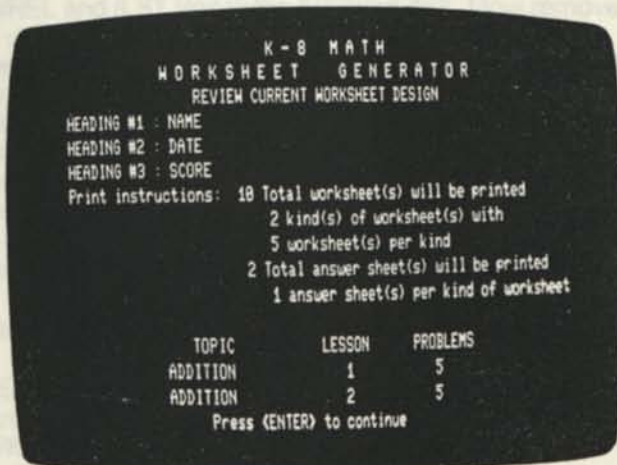




The screen displays the name of each topic (addition, subtraction, multiplication, and division) in turn, and asks which lessons from that topic you want the worksheet to cover. For example, the screen above is asking you to select lessons from the topic "addition." To include some addition problems, you'd type the numbers of the desired lessons, separated by commas, and press <ENTER>. For each lesson number you chose, you'd be asked for the number of problems to include from that lesson. To "skip" addition, you'd simply press <ENTER>, without selecting any lessons. Before you return to the option list, the program gives you the chance to redo your selections.

Option 4 lets you choose the total number of worksheets to be printed, the number of kinds of worksheets (different worksheets on the same level of difficulty), and the number of answer sheets you want for each kind of worksheet.

Option 7 lets you review the current worksheet design. Included in the review are the names of the headings entered in Option 2, the print instructions entered in Option 4, and the topics, lesson numbers, and number of problems entered in Option 3. For example:



Once you have completed Options 2, 3, and 4, your worksheet is ready to be saved on diskette. When you select Option 5 to "Save worksheet design," you'll be asked to enter a worksheet name. You are also given the option to enter a password, which protects your worksheet design from unauthorized users. Finally, you enter a drive number, and the computer saves the worksheet design.

Option 6 lets you print out the worksheets. First, the computer pauses to generate exercises. Then you simply ready the printer and start the printing. The computer does the rest.

Option 1 lets you recall a worksheet design that you saved on the diskette earlier, in order to print out more copies. Each time you recall and print a specific worksheet, the problems which appear on the worksheet will be different. This is because the computer randomly generates new problems, using the guidelines of your worksheet design, each time that worksheet is printed. This feature is particularly convenient for a teacher who will be using a worksheet for more than one class or for more than one semester. Not only through Option 4, but also with every printing, the teacher can get different worksheets on the same topic and level of difficulty without having to rewrite the worksheet. Since an answer key is generated for each new worksheet, the time spent in grading homework and exercises is no greater than it would be if all the worksheets were identical.

HELPFUL HINTS FOR USING THE K-8 MATH WORKSHEET GENERATOR

Fourth-grade teacher Lou Weiner, of Saugus, California, has been using TRS-80 microcomputers for about three and one-half years, and has been using the Worksheet Generator program for almost a year. Before using Worksheet Generator, Weiner programmed some of his own worksheet-producing software. But Weiner is impressed with Worksheet Generator because, he says, it has "a lot more parameters." For example, his multiplication programs didn't force carrying, while many of the lesson levels in Worksheet Generator's multiplication topic do. He also uses Radio Shack's K-8 Math With Student Management Program, Volume One, in his classroom.

Weiner has come up with some good ideas for using Worksheet Generator in the classroom. When a worksheet contains a whole page of long-division problems, for example, it takes time for the computer to print out enough sheets for an entire class. So instead of starting up the printer and leaving it, Weiner puts a spirit master in the printer. The computer prints the worksheet directly onto the master, which Weiner then uses to make as many copies as desired.


Weiner says that the answer sheet that is also printed can be a single answer key for the teacher's use, or it can be used to make many copies so that students can check their work in class. (If you don't plan to use spirit masters, remember that the program allows you to choose how many copies of the answer sheet you want to print.) Weiner finds it useful for students to check their work in class because, he says, the answer sheet "shows each line of the problem as it should be worked." Students can look at the answer sheet and see, for example, whether they are multiplying incorrectly or just making mistakes in addition.

Weiner has also found a diagnostic use for the Worksheet Generator program. He uses the program not only for worksheets, but also says, "all my tests are done with it." When he designs a worksheet, he prints out the screen display from Option 7 ("Review current worksheet design" by pressing <SHIFT> <↓> <*> at the Model III keyboard). This gives him a quick-reference list of the lesson numbers covered in the worksheet or test. For students who do not do well on the worksheet, he uses the list as an aid in designing

individualized supplementary worksheets that target the concepts missed.

HARDWARE REQUIREMENTS, PRICE, AND AVAILABILITY

The K-8 Math Worksheet Generator program (Cat. No. 26-2162) can be used with a TRS-80 Model III 32K or 48K disk system and any Radio Shack line printer. The suggested retail price of the program is \$99.95, though prices may vary at individual stores and dealers.

Worksheet Generator is available through any Radio Shack store or Computer Center. Also, Radio Shack has Regional Educational Sales Coordinators throughout the country to help schools determine and meet their educational computing needs. For details, call your local Radio Shack store or Computer Center. 


Single Disk FORTRAN

Tim Daneliuk
4927 North Rockwell Street
Chicago, IL 60625

Here is a step-by-step method for running the Radio Shack FORTRAN Compiler with a single disk drive on a TRS-80 Model I (with special thanks to Spencer Lepley of Tallahassee Florida who figured out how to link on a single drive without L80 being on the program diskette):

- 1) Load the disk with EDIT on it and load this program by typing in EDIT. Now remove the disk.
- 2) Place the disk that is to contain your FORTRAN Source file into the drive and EDIT the file as desired. Save the source on this disk. Now remove this disk.
- 3) Insert the disk with F80 on it and load the program by typing F80. Now remove this disk.
- 4) Insert the disk with your source program on it. Compile the program as usual (e.g. TEMP, TEMP = TEMP). This writes the /REL and /LST files to the program disk. Now remove this disk.
- 5) Insert the disk with L80 on it and load the linker by typing in L80. Remove this disk.
- 6) Insert the disk with the /MAC, /REL, and /LST files on it (really all you need is the /REL file, so if have a copy of the source elsewhere and need the disk space, kill the /MAC and /LST files: they are real space hogs!). Now key in the file name you want to link (e.g. TEMP). This loads the relocatable code and displays the undefined globals (like the ones from FORLIB). Remove this disk.
- 7) Insert the disk with FORLIB on it. Note, at this point you are still in L80! To resolve the undefined globals type in FILENAME-S, where FILENAME is the file you want to search to find the global definitions. Usually, this will be FORLIB/REL-S (S is for search) unless you have your own library files. Remove this disk.
- 8) Insert the disk which is to have the final executable /CMD file on it. Key in FILENAME-N to output object code, then -E to write the file and then exit L80. The process is complete.

You can save space on a disk by killing the FORMAT/CMD, BASIC/CMD, BASICR/CMD, and BACKUP/CMD files (they are invisible protected files whose password is their own

name. For example BASIC and BASICR have the password BASIC. FORMAT has the password of FORMAT and so on. 

Color Logo Now Available for the TRS-80 Color Computer

Color LOGO is Radio Shack's new LOGO program for the 32K Disk-based or 16K ROM pack TRS-80 Color Computer. "LOGO" is an educational computer language that can be used to draw pictures on the computer's video display. When simple instructions are entered at the computer keyboard, a shape on the screen (called a "turtle") moves around and leaves a trail of color.

Color LOGO can be a lot of fun for a computer enthusiast or novice of any age, but its main educational use has been with young children.

Color LOGO isn't a language for teaching. You wouldn't use the language to present traditional lesson material. But educators who have been involved in the development and use of LOGO-type programs have been enthusiastic about the language's power to let children learn by exploring. To make the turtle move, children have to plan an action and then describe it to the turtle through commands like "forward," "back" , "right," and "left". This involves them in thinking logically about direction, distance, and angle. They can define a problem such as; "How can I draw a square?" and then break the problem into its parts (in this case, four equal moves forward, with a 90 degree turn between each move). The language gives children vast opportunities to be creative.

HOW TO USE COLOR LOGO

Let's look at how Radio Shack's Color LOGO program can be used and at a few of its many special features.

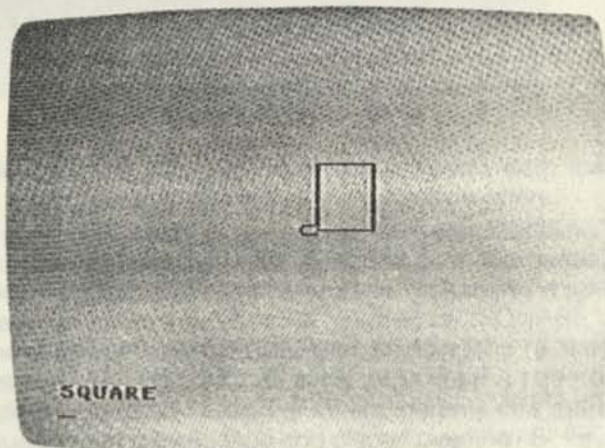
Color LOGO can be loaded into the TRS-80 Color Computer via diskette or ROM pack. User programs written in Color LOGO can be saved onto diskette or cassette.

To move the turtle on the screen, you get into the program's "Run" mode. There you can enter commands such as "FORWARD" (which can be abbreviated as "FD"), "LEFT" (or "LT"), and so on. Along with a command like "FD", you type a number that tells the turtle how far forward to move. With a command like "LEFT", the number indicates how many degrees the turtle should turn in place before receiving the next command to move forward or back. As you enter each command, the turtle makes one move or one turn. By entering commands one at a time, you finally create a picture on the screen.

Once you've mastered the basics of turtle movement, you can write a series of turtle movements and save the series. Such a set of turtle movements is called a "procedure." Color LOGO provides line-oriented editing for procedures, through the program's "Edit" mode. That is, Color LOGO lets you type each command ("FD", "LT", etc.) into the computer and save it as part of the procedure, without having the turtle follow each command as you type it.

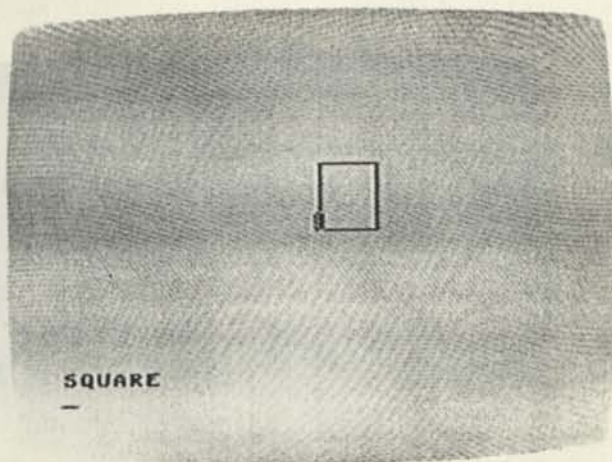
For example, the directions below form a procedure for drawing a square. The prefix "TO" marks the name of the program. "END" signals the end of the program. The information between these two lines consists of ordinary turtle commands. (For example, "FD 30" means "move forward 30 units"; "RT 90" means "turn right 90 degrees."):

```
TO SQUARE
FD 30 RT 90 FD 30 RT 90 FD 30
RT 90 FD 30
END
```



You can use Color LOGO's "REPEAT" command to save yourself some typing while you write the same procedure. The directions below tell the turtle to repeat the action in parentheses four times. (The only difference in this version of "SQUARE" is that the turtle makes an extra right turn after completing the square.)

```
TO SQUARE
REPEAT 4 (FD 30 RT 90)
END
```



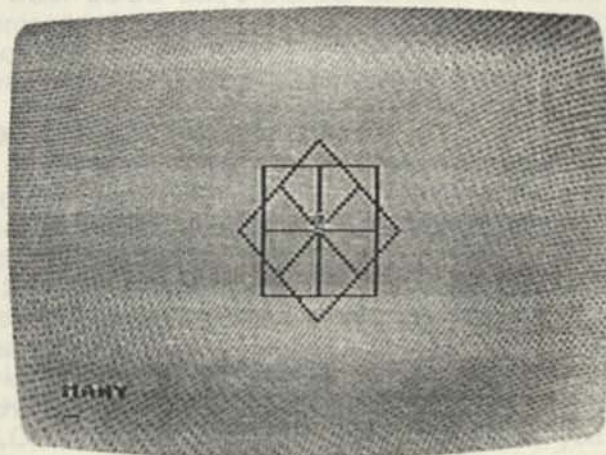
Once you've entered a procedure in memory, you can get into the "Run" mode and use the procedure just like any turtle command. That is, entering "SQUARE" as a command (instead of entering a simple turtle command like "FD 30"), would make the turtle follow all the steps outlined in the procedure "SQUARE".

PROCEDURES WITHIN PROCEDURES

Since a whole procedure can be used just like a simple turtle command, this means that you can include procedures within procedures, just as you include simple turtle commands within procedures. For example, the procedure below tells the turtle to repeat "SQUARE" ten times, making a 45 degree turn to the left before each repetition:

```
TO MANY
REPEAT 10 (SQUARE LT 45)
END
```

Once the procedure "MANY" was in memory, entering the command "MANY" in the Run Mode would produce a more complicated design made with ten squares:



VARIABLES IN PROCEDURES

Now suppose that you want to draw a lot of squares of different sizes. Instead of writing a different square procedure for each different size, you can use variables. Using one or more variables in a procedure allows you to change the procedure every time you run it.

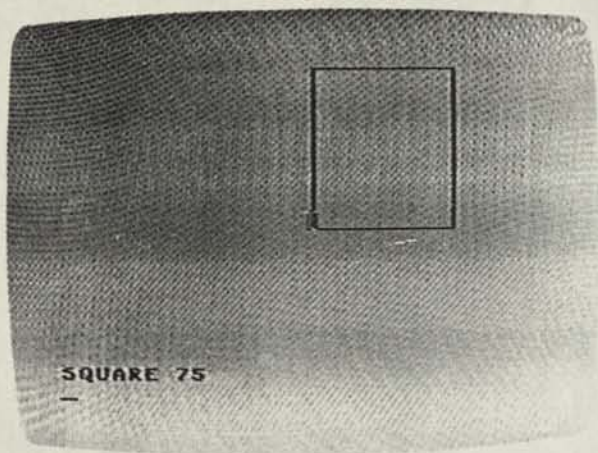
To change the size of a square, you need to change the length of its sides. In the procedure "SQUARE", the sides are drawn by the turtle command "FD 30". To change the length of the sides, we just need to change the number after "FD". With a variable, we can leave the length of the sides open for change each time we run the procedure. For example:

```
TO SQUARE :LENGTH
REPEAT 4 (FD :LENGTH RT 90)
END
```

This way, entering "SQUARE 10" in the "Run" mode will produce a square with sides 10 units long. Entering "SQUARE 50" will make the sides 50 units long, and so on.

SPECIAL FEATURES FOR VERY YOUNG CHILDREN

Color LOGO's "Doodle Mode" is a special feature for children who can't read yet. In Doodle Mode, keys on the top row of the TRS-80 Color Computer correspond to turtle commands like "RT 45", "FD 10", and so on. A plastic overlay for these keys contains graphic representations of each key's meaning. This feature allows children who can't yet read or



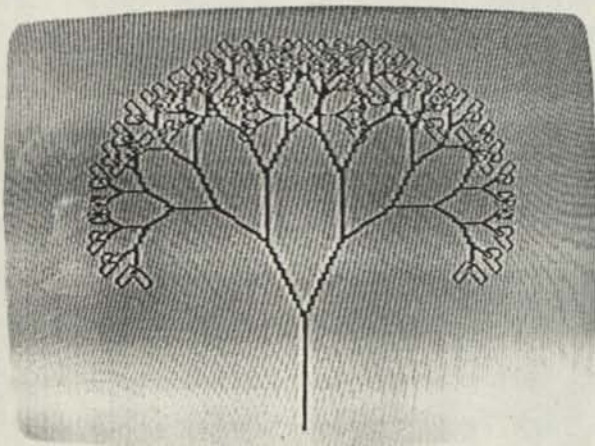
type reliably to use the language and to benefit from the practice in structured thinking that the language offers.

Or, parents or teachers can set up "One-Key Doodling" in the Run mode. The adult writes a procedure that is named by one key. For example, a procedure like "SQUARE" might be written and given the name "S" instead of "SQUARE". When the child works with the program, pressing the "S" key and pressing "ENTER" will make the turtle draw a square. For children who don't recognize letters, parents can cut a small square out of colored tape and place it over the "S" key.

MULTIPLE TURTLES

Unlike most versions of LOGO, Color LOGO allows more than one turtle to be at work on the screen at once. Several parts of a complex drawing can be drawn at once, with the effect that the drawing seems to evolve rather than being drawn piecemeal. For example, you can write a procedure that draws a many-branched tree, with a turtle ending up at the end of each branch.

Multiple turtles allow you to program simple games in Color LOGO. Each player can be assigned one turtle that maintains its position until that player's next turn.



Color LOGO also allows you to create new shapes for turtles. You can use different-shaped turtles as different pieces in a game, or can program simple animation and use turtles as animated figures.

OTHER COLOR LOGO FEATURES

A "SLOW" command lets you vary the speed at which procedures are carried out. Speeds range from "0" (full speed) to "127" (slowest speed).

A "PENUP" command lets you move the turtle without leaving a trail on the screen.

A "HIDETURTLE" command lets you make the turtle invisible.

Arithmetic expressions can be used with variables in Color LOGO procedures.

As mentioned above, limited animation is possible in Color LOGO.

Screen colors can be changed, including the overall colorset, the background color, and the color of the line drawn by the turtle.

The Color LOGO user's manual contains numerous sample procedures, including some at the end of the manual that demonstrate special effects. For example, one of these procedures draws a clock face with moving hands!

HOW YOU CAN USE COLOR LOGO

The Radio Shack Color LOGO program is ideal for classroom or home. The program is available through Radio Shack stores and Computer Centers nationwide. Color LOGO for 32K disk-based TRS-80 Color Computer (catalog number 26-2721) is listed at \$99.00. The suggested retail price of the ROM pack version (Cat. No. 26-2722) for 16K ROM-based TRS-80 Color Computer is \$49.95. Prices may vary at individual stores and dealers.

Much has been written about the educational theory behind LOGO-type programs. For more information, you may wish to consult Seymour Papert's book *Mindstorms: Children, Computers, and Powerful Ideas* (Basic Books, 1980) and the book *Turtle Geometry: Computation as a Medium for Exploring Mathematics* (M.I.T. Press) by H. Abelson and A. DiSessa. Chapter One of *Mindstorms* was condensed in the March 1981 issue of *Creative Computing*, under the title "Computers and Computer Cultures."

MODES IN COLOR LOGO

The Color LOGO system can be in one of 4 "modes" depending upon what the user is doing at the time.

BREAK MODE is entered automatically upon starting Color LOGO, and can be entered from any other mode by pressing the BREAK key at any time. It is signified by the "LOGO:" prompt on the screen. The following single-letter commands may be used:

R enters RUN MODE

E enters EDIT MODE

P prints contents of internal program on the printer connected to the serial port.

Q prints same as P common except the Q sends a line feed after a return character.

L prompts for module designation with a "LOAD:", then reads from the specified source into the internal program area.

S prompts for the module name with a "SAVE:", writes the internal program area to the specified destination.

EDIT MODE is entered from BREAK MODE by pressing E. In EDIT MODE one can edit the currently loaded modules. To start with a blank program area, press SHIFT CLEAR in BREAK MODE before pressing E.

RUN MODE is entered from the BREAK MODE by entering "R." When RUN MODE is entered, the screen is cleared and the turtle appears at the home position. A TEXT WINDOW of three lines exists at the bottom of the screen. The user enters turtle graphics commands or calls Color LOGO procedures that have been entered earlier via the EDIT or DOODLE MODE. The user can enter any of the following commands directly in RUN MODE. However, in RUN MODE no more than one command may be entered on a line. Once the ENTER key is pressed, the command is executed.

CLEAR	HOME
FORWARD	BACK
RIGHT	LEFT
PENUP	PENDOWN
PENCOLOR	SHOWTURTLE
HIDETURTLE	SETX
SETY	SETHEADING
SLOW	COLORSET
BACKGROUND	WRAP
NOWRAP	SEND
PRINT	HATCH
VANISH	

DOODLE MODE is entered from RUN MODE by pressing the "@" key. DOODLE MODE allows the creation of a turtle graphics procedure that will draw a shape without requiring that the user even know how to read. In DOODLE mode the screen displays an "=" sign. The user enters a word (nonsense or otherwise) of at least one letter or number and presses (ENTER). The word is the name of the procedure to be created as a picture is drawn.

The numeric keys (marked by the special keyboard overlay) can be used to enter turtle graphic commands. Each time a key is pressed, the specified command is executed by the turtle. At the same time, a procedure is created in the program area. This procedure can be viewed by entering EDIT MODE.

The DOODLE MODE commands are:

(1) CLEAR	(6) LT 45
(2) HOME	(7) FD 1
(3) PU	(8) FD 10
(4) PD	(9) RT 15
(5) RT 45	(0) LT 15

COLOR LOGO STATEMENTS AND COMMANDS

Control statements include:

IF expr The expression is evaluated. If the value is true (non-zero) the list of statements in parenthesis is executed. If it is false (0) then the list of statements is skipped.

ELSE This statement must follow an IF statement. If the expression value on the IF statement is false, then the list of statements after ELSE is executed. Otherwise it is skipped.

REPEAT expr The expression is evaluated; if it has a value less than or equal to zero, then the list of statements is skipped. Otherwise the list of statements is executed the specified number of times.

WHILE expr The expression is evaluated; if it is false (0) then the list of statements is skipped. If it is true (non-zero) then the list of statements is executed. After the list is executed, control returns to the WHILE and the process is repeated.

STOP This terminates the execution of a procedure. Control is returned to the calling procedure if there is one. If the procedure was called from RUN MODE, then control returns to RUN MODE.

TO procname parmlist This statement defines the start of a Color LOGO procedure. It must start in column 1 of a line and must be the only statement on the line. The "prname" may be any name of one or more letters.

END This is the last statement in a procedure. Execution of an END is equivalent to that of the STOP statement.

VANISH VANISH takes the current turtle out of existence.

MAKE :var expr The value of the expression is assigned to the variable.

PRINT "TEXT"

PRINT expr The literal text of the expression value is displayed at the turtle location. The turtle is not moved.

NOWRAP Execution of the NOWRAP statement takes the screen out of its normal wrap mode. If a turtle then runs off the screen the program will terminate with an "OUT OF BOUNDS" error message.

WRAP Puts the screen back in wrap mode.

HATCH expr procname arglist Creates a new turtle. The turtle will start at the same (X,Y) position as its parent (the turtle which HATCHed it) and will be pointed in the same direction.

SEND expr expr A message is sent to the specified turtle. The first expression value denotes the identification of the turtle to which the message is sent. The value of the second expression is the value sent to the other turtle.

procname arglist This is referred to as a CALL statement, even though it does not contain the word CALL. To CALL any procedure, just code its name followed by any arguments to be passed.

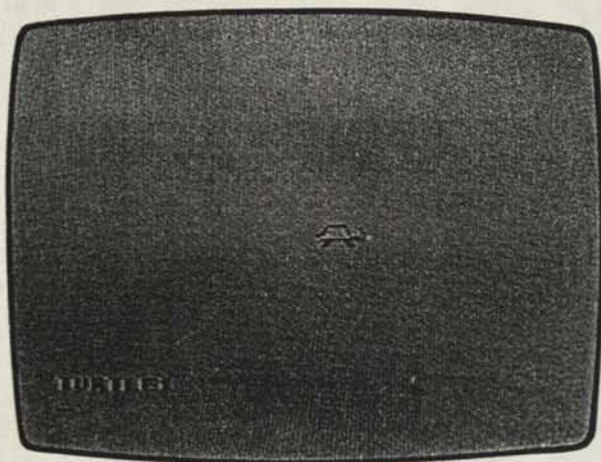
SLOW expr The SLOW statement causes execution to slow down so that it can be watched more closely. The value of the expression denotes how slow to go.

Turtle Graphics Commands include:

BACK expr	PENDOWN
BACKGROUND expr	PENUP
COLORSET expr	RIGHT expr
CLEAR	SETHEADING expr
FORWARD expr	SETX expr
HIDETURTLE	SETY expr
HOME	SHAPE shape list
LEFT expr	SHOWTURTLE
PENCOLOR	

TURTLE SHAPE LIST

The SHAPE statement is used to assign a new shape to the current turtle. The shape of a turtle is made up of a pattern of dots on a grid. The shape list tells Color LOGO how to draw the turtle pattern. The turtle shape is automatically rotated to face in the direction the turtle is headed. Drawing the turtle shape is similar to using normal turtle graphics commands to draw any shape. The difference is that the commands which make up the shape list are a restricted and simplified form of the normal turtle graphics commands. The commands allow a step of one pixel (one square on a piece of graph paper) in any of the 8 possible directions. The 8 directions are: up, down, right, left, and the four diagonal directions.



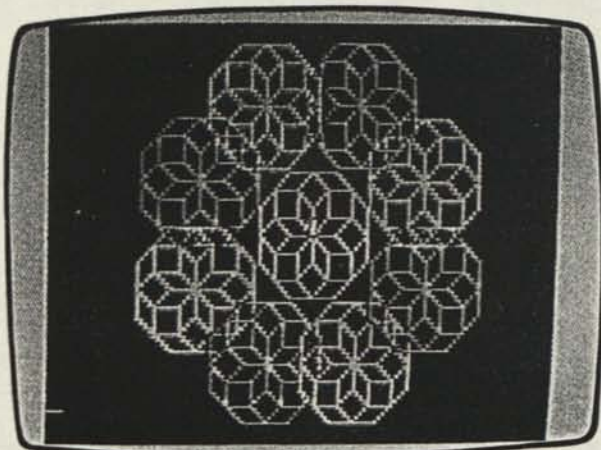
THE CLOCK PROGRAM FROM COLOR LOGO

Copyright © MicroPi

```
TO CLOCK :DELAY :INT
  CLEAR
  CLOCKFACE
  TIME :DELAY :INT
END
```

```
TO CLOCKFACE
  MAKE :NUMBER 12
  SY 180 SX 104 SH 90
  REPEAT 12
    (FD 22 RT 90 FD 5 BK 5
     PU BK 10 PRINT :NUMBER
     FD 10 PD LT 90 FD 22
     RT 30
     MAKE :NUMBER :NUMBER+1
     IF :NUMBER > 12
       (MAKE :NUMBER 1))
  END
```

END



```
TO TIME :DELAY :INTERVAL
  HT
  REPEAT 24
```

```
(MAKE :HR 0
 WHILE :HR<12
 (DIGITAL :HR :MIN
  PC 1 LITTLEHAND :HR :MIN
  PC 2 BIGHAND :MIN
  REPEAT :DELAY ()
  PC 3 LITTLEHAND :HR :MIN
  BIGHAND :MIN
  MAKE :MIN
    :MIN + :INTERVAL)
  MAKE :HR :HR + 1))
```

END

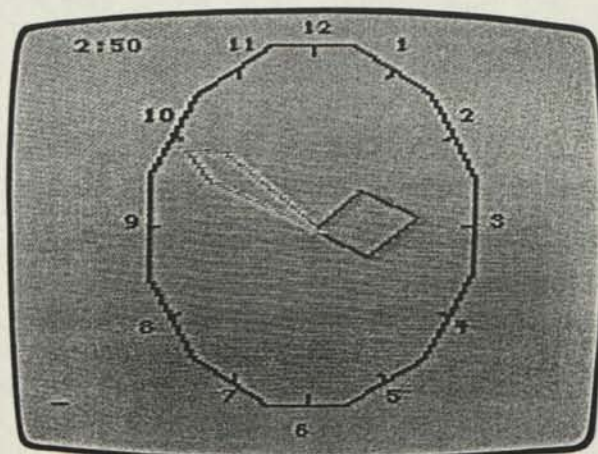
```
TO BIGHAND :MINUTE
  SX 128 SY 96 SH 6* :MINUTE
  LT 8 FD 60 RT 30 FD 18
  RT 130 FD 18 RT 32 FD 60
END
```

```
TO LITTLEHAND :HOUR :MINUTE
  SX 128 SY 96
  SH 30* :HOUR + :MINUTE/2
  LT 32 FD 30 RT 60 FD 30
  RT 120 FD 30 RT 60 FD 30
END
```

```
TO DIGITAL :HOUR :MINUTES
  SX 0 SY 180 PRINT " "
  SX 8*( :HOUR<=9 & :HOUR<>0)
  IF :HOUR (PRINT :HOUR)
  ELSE (PRINT 12)
  SX 16 PRINT ":" SX 24
  IF :MINUTES<10
    (PRINT "0" SX 32)
  PRINT :MINUTES
END
```

Notice that you can set the interval to any number of clock minutes and that you can set the speed with :DELAY. Try running

CLOCK 300 5



TRS-80[®]

Volume 4, Issue 11

DECEMBER 1982

\$1.50

Microcomputer News

Outside Software Support

Model III Business Graphics

1982 Annual Index



Fort Worth Scene

Here we are at the end of our first forty-eight page per issue year. Thumbing through the last year of Newsletters, brings to mind the many changes that have taken place. 1982 saw Bill Barden's series of articles on 6809 assembly language for the Color Computer begin in February as did our two for one media exchange program for user submitted material on tape or disk (not to mention CompuServe).

We have looked at the Pocket Computer II, SuperSCRIPSIT, the Model II Graphics board, the digitizer, hard disks, a whole new line of printers (including dot matrix, daisy wheel, and plotters) for all the computers, terminals, myriad software packages, a new operating system in LDOS, and much, much more. It all leaves us in a state of wonder and breathless anticipation—wonder at what has been accomplished in the Radio Shack line of computer products in this last short year and anticipation at the promise of the many exciting things yet to come.

RING OUT THE OLD, BRING ON THE NEW


While 1982 was a good year for TRS-80s (excellent in fact), 1983 promises to be even better. We cannot tell you everything that may be coming up in the computer line, but we thought you might be interested in the 1983 TRS-80 Microcomputer News topics.

- January—Getting Started
 - Where are you going to put it?
 - Getting it running
 - Selection and care of software
 - What if it doesn't work?
- February—Utilities
 - FORMAT, BACKUP, XFERSYS
 - PATCH, BUILD, DO
 - Personal utilities
 - BASIC utilities
- March—Peripherals
 - What is a peripheral?
 - Storage devices
 - Input devices
 - Output devices
- April—Communications/Databases
 - What is a database?
 - How do I talk to a database?
 - Commercial databases
 - Personal/Business databases
- May—Home
 - Education
 - Management
 - Environmental Control
 - Recreation
- June—Graphics
 - Low resolution
 - High resolution
 - Printers
 - Using graphics
- July—Games
- August—Word Processing
 - What is a word processor?
 - SCRIPSIT
 - Writing your own word processor
 - Getting the most out of your word processor
- September—Education
 - Computer literacy
 - Home education



- Business education
- Education resources
- October—What's New and System Integration
 - What's new for '84?
 - Getting programs to work together
 - Using Dow Jones/CompuServe with BASIC
 - Terminal/SCRIPSIT/BASIC
- November—Operating Systems
 - What is an operating system?
 - TRSDOS
 - LDOS
 - Multi-user
- December—Languages/Modeling/Christmas
 - Why are there different languages?
 - Machine language
 - High level languages
 - Modeling

If you have something related to one of these topics that you want to send us for possible publication, keep in mind that we work three to four months ahead of the month that the News actually goes into print. Don't let that be a limiting factor though, because if you send something we really like, we can run it any time.

Thank you for your contributions and interest in TRS-80 Microcomputer News. Best wishes for the holiday season and a very happy, prosperous 1983 to you and yours from all our staff. 



TRS-80[®] Microcomputer News

TRS-80 Microcomputer News is published monthly by Radio Shack, a division of Tandy Corporation, One Tandy Center, Fort Worth, Texas U.S.A. 76102. Copyright 1982 by Tandy Corporation, One Tandy Center, Fort Worth, Texas U.S.A. 76102. All rights reserved.

Reproduction or use, without express written permission from Tandy Corporation of any portion of the Microcomputer News is prohibited. Permission is specifically granted to individuals to use or reproduce material for their personal, non-commercial use. Reprint permission for all material (other than William Barden's article), with notice of source, is also specifically granted to non-profit clubs, organizations, educational institutions, and newsletters.

TRS-80 Microcomputer News is published monthly by Radio Shack, a division of Tandy Corporation. A single six month subscription is available free to purchasers of new TRS-80 Microcomputer systems with addresses in the United States, Puerto Rico, Canada and APO or FPO addresses. Subscriptions to other addresses are not available.

The subscription rate for renewals and other interested persons with U.S., APO or FPO addresses is twelve dollars (\$12.00) per year, check or money order. Single copies of the Microcomputer News may be purchased from Radio Shack Computer Centers or Computer Departments for \$1.50 suggested retail each.

The subscription rate for renewals and other interested persons with Canadian addresses is Fifteen dollars (\$15.00) per year, check or money order in U.S. funds. All correspondence related to subscriptions should be sent to: Microcomputer News, P.O. Box 2910, Fort Worth, Texas 76113-2910.

Retail Prices in this newsletter may vary at individual stores and dealers. The company cannot be liable for pictorial and typographical inaccuracies.

Back issues of Microcomputer News prior to January, 1981 are available through your local Radio Shack store as stock number 26-2115 (Suggested Retail Price \$4.95 for the set). Back issues of 1981 copies are available as stock number 26-2240 (Suggested Retail Price \$9.95 for the set).

The TRS-80 Newsletter welcomes the receipt of computer programs, or other material which you would like to make available to users of TRS-80 Microcomputer systems. In order for us to reprint your submission, you must specifically request that your material be considered for reprinting in the newsletter and provide no notice that you retain copyrights or other exclusive rights in the material. This assures that our readers may be permitted to recopy and use your material without creating any legal hassles.

Material for publication should be submitted on magnetic media (tape, disk, or CompuServe). If you submit material on tape or disk, and it is accepted for publication, we will send you two cassettes or diskettes for each one you sent us. Cassettes will come from our box of mixed blank cassettes. If you submit material on CompuServe, and we think we may use the material, we will extend your Microcomputer News subscription by six months for each article accepted. If you are submitting material over CompuServe, please include your name and address or your subscription number so we can find you. If the material is very short, send it to us in E-Mail. If you have more than a few lines, you need to place the material in the ACCESS area of CompuServe and then let us know it is there by leaving a message on E-Mail.

Material may be submitted by mail to P.O. Box 2910, Fort Worth, Texas 76113-2910, or through CompuServe. The Microcomputer News' CompuServe user ID number is 70007.535.

Notes to Program Users:

Programs published in the Microcomputer News are provided as is, for your information. While we make reasonable efforts to ensure that the programs we publish here work as specified, Radio Shack can not assume any liability for the accuracy either of the programs themselves or of the results provided by the programs.

Further, while Microcomputer News is a product of Radio Shack, the programs and much of the information published here are not Radio Shack products, and as such can not be supported by our Computer Customer Service group. If you have questions about a program in the Microcomputer News, your first option is to write directly to the author of the program. When possible, we are now including author's addresses to facilitate communications. If the address is not published, or if you are not happy with the response you get, please write us here at Microcomputer News. We will try (given the limited size of our staff) to find an answer to your question and, in many cases, will publish the answer in an up-coming issue of Microcomputer News.

Trademark Credits

CompuServe[™] CompuServe, Inc.
 Dow Jones[™]
 NEWS/RETRIEVAL Service[®] Dow Jones & Co., Inc.
 VisiCalc[®] VisiCorp, Inc.
 Program Pak[™] Tandy Corporation
 SCRIPSIT[™] Tandy Corporation
 TRSDOS[™] Tandy Corporation
 TRS-80[®] Tandy Corporation

Contents:

1982 Cumulative Index	21
Color Computer	
Color Computer EDTASM+ Tips	29
Bugs, Errors, and Fixes	
Color Disk Drive Kit (26-3022)	6
Programs	
Christmas Card by Franklin Pigoulias	43
Christmas Eve by Todd Day	39
Concentric Circles on the CoCo by Donald Whitelock	43
Text Editor Revisited by Ashok Basargekar	41
Computer Clubs	40
Computer Customer Service	
Holiday Fun with Your TRS-80	16
Data Bases	
CompuServe	7
TRS-80 Christmas Story—Billy and Santa	
Profile	18
Profile III+: Setting Up a User Menu	
Standard and Poor's	9
Now You Too Can Become a "Wall Street Wizard"	
VisiCalc/SATAN	35
Personal Time Savers Revisited	
Education	
Chemistry Simulations for Secondary Students	33
Feature Story	
Outside Software Support by Bruce Elliott	5
Fort Worth Scene	2
General Interest	
Microcomputers in Education	32
Reprints Are Now Available	34
Model I/III	
Business Graphics	11
LDOS Hints and Tips	30
Model III Port Assignments	15
Model II	
Bugs, Errors, and Fixes	
BEDIT (26-4705) with TRSDOS 4.1	32
BISYNC (26-4715)	32
Model II/16 TRSDOS (26-4150) Version 4.1	31
Payroll (26-4503)	32
Profile II (26-4512) Version 1.0	32
Programs	
Model II Twinkling Christmas Tree by Robert P. Graham	31
Notes on Previous Newsletters	25
November, 1981	
Model III Directory Menu	
February, 1982	
BASICally Speaking	
March, 1982	
Relocating Machine Language Programs	
April, 1982	
Linefeed with a Carriage Return	
June, 1982	
Concentric Circles	
Graphs for the PC-2	
July/August, 1982	
Asteroids	
Color SCRIPSIT Bug	
Graphic Keys	
More about SCRIPSIT	
SCRIPSIT and the LP VIII	
Sieve of Eratosthenes	
Peripherals	
Modem II Programming Techniques	37
Pocket Computer	
Availability of 16K RAM Modules for the PC-2	42
Programs	
A Clock/Scheduler for the Busy Executive by John Titus	28
PC-1 Twelve Days of Christmas by Robert K. Phelps	42
View from the 7th Floor by Jon Shirley	4

Prices shown in TRS-80 Microcomputer News are in U.S. Funds.

View From the Seventh Floor

by Jon Shirley
Vice President
Radio Shack Computer Merchandising

It is hard to believe that it is Christmas time already and another year is almost gone. Back in September we had our fifth anniversary in the microcomputer business but we hardly had time to notice it, as we surged into our sixth year with a great many new ideas for new products. 1983 will be our most exciting year, yet 1982 was certainly not too bad.

Christmas, among its other connotations, does mean gifts, and we have numerous suggestions for the TRS-80 owner. Our number one family gift is the Color Computer, specially priced for giving. If you already have a CoCo check out all the new software like MegaBug, Microbes, Poltergeist, Clowns and Balloons, and Tennis. A Pocket Computer is a neat idea for the man who has everything. Videotext software and a modem are welcome additions to any TRS-80. I don't want to run an ad here, but if you are stuck for gifts, check out the local Shack; we have gifts for boys and girls from 8 to 80.

One gift any owner of a computer would like is a subscription to a magazine. Here are a few ideas for you, based on what we read. Please understand that we are not involved in these publications but merely want you to know they exist.

80-US is a rapidly growing publication that supports all our Models from the Pocket Computer to the Model 16. They review products and publish programs. They would like to increase their readership, so here is a special offer for Microcomputer News readers. Write to them and say you are interested, and they will send you one free issue to check out. No risk or obligation. If you like it, subscribe. If you don't like the publication, give it to the Boy Scouts paper drive. Write to them at: 80-US Journal, 3838 South Warner St., Tacoma, Washington 98409, and request a sample copy.

The Rainbow is exclusively for the Color Computer and it has been growing in size recently. They also publish and review software, and it is a good source for third party programs. They can be contacted at: The Rainbow, 5803 Timber Ridge Dr., Prospect, KY 40059.

Also for the Color Computer is an interesting publication, without ads, called the Color Computer Index. It is an index to articles, programs, reviews, and news releases about the Color Computer. The one I saw had 25 pages of items listed by title, author, and subject from 6 different magazines and the TRS-80 Microcomputer News. They do not publish the actual articles, just a bibliographic citation and brief annotation. Of course, most of the cited articles can be found at a local library. For more information write to: American Library and Information Services, 3705 Mary Ellen NE, Albuquerque, NM 87111.

Yet another new magazine is called two/sixteen and is obviously for Model II and 16 owners. It is published bi-monthly, but might go monthly if they get enough subscriptions. It is a nice looking magazine and the issue I saw had a


great deal of useful information. Write to: two/sixteen, 131 East Orange St., Lancaster, PA 17602.

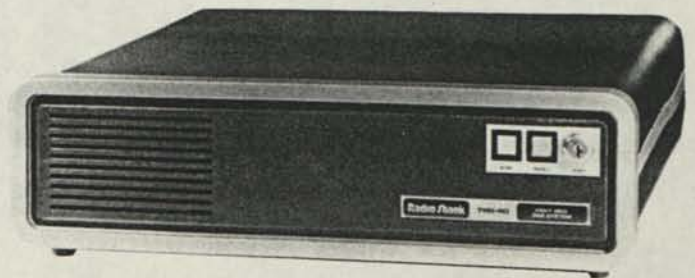
HARD DISK

Off the Christmas theme for a minute, I would like to talk about the hard disk. I use one, and I now have a lot of very valuable information on it. To insure the safety of that information, I do backups. Hard disks do fail, although not very often, and it is easy to forget the backups and assume it will run forever. Do that and it is certain to fail. Someone's law. Our manual leaves something to be desired and is being redone, but here is the procedure I follow and recommend that you follow, if you keep large amounts of data updated frequently on the disk. Once a month backup the entire disk. Do that by invoking `SAVE :4 TO 0 {SYS,ALL}`. This will save all user files and programs and system items like runtimes that are not on the INIT disk.

Each day (or even twice a day if the disk is used all day long) back up those files modified since the last backup. To do that: `SAVE !:4 TO 0 {DM>MMDDYY}` where MMDDYY is the date of your last full month end backup. This will SAVE only those files or programs that were changed since the last full backup. Should your disk fail, all you would need are two sets of diskettes to recapture everything up to the end of the previous day. The procedure would be `RESTORE 0 TO 4 {SYS}`, using the month end set, then `RESTORE 0 TO 4` again using the month end set. Then take the last day set and do `RESTORE 0 TO 4 {ABS}`. This last step will write over those files that have been modified and bring them up to date. If you are a very heavy user of the hard disk, you may want to do the big save more often, twice a month or even weekly, to make the daily saves faster.

And no matter which machine you have or what media you use, please make backups if you value your data. There is no such thing as a failsafe computer.

Thanks for reading the Microcomputer News this year and for your support of the TRS-80. Have a happy holiday and a great new year. 



Outside Software Support

Radio Shack is launching an innovative new Outside Software Support Program. This new program offers a variety of support and assistance services to third-party software vendors who are developing software to run on any of Radio Shack's TRS-80 computers. Radio Shack Vice President for Computer Merchandising, Jon Shirley, acknowledges the program as "A major effort to aggressively support the needs of independent software developers who are themselves interested in supporting the needs of TRS-80 computer owners and users."

This new Outside Software Support Program is available to virtually all professional software developers with an interest in creating and marketing programs to run on Radio Shack TRS-80 computers. "This is most assuredly a third-party plan in every sense," comments Shirley. "Unlike our earlier policy, the software being developed with the support of our new program does not have to be intended for eventual sale through Radio Shack stores and dealers. We're putting this program in place specifically to help develop new options for people who own our hardware."

Technical assistance is one of the key services provided by the new Outside Software Support Program; others include customer referrals through a software review program and a directory of outside software, and product line consideration, if desired, by Radio Shack.

The Technical Assistance Program will provide information to aid the development of software for TRS-80 computers. Licensing of Radio Shack TRSDOS (disk operating systems) and BASIC, sale of TRSDOS source codes and bulk sale of RUNTIME packages will be made available. Information updates on new operating system releases, new patches and new hardware will also be offered.

SOFTWARE SUPPORT PROGRAM

The Radio Shack Software Support Program was established to assist third party software firms in the development and/or conversion of reliable, specialized software for our TRS-80 line of computers.

If you or your company wish to develop specialized software packages for the TRS-80 computer line under TRSDOS and are interested in a relationship with Radio Shack, write for an Application (or pick one up at your local Radio Shack Computer Center—Form # FC-3013), and mail it to Radio Shack.

If your application meets the criteria for the Radio Shack Software Support Program we will send you an information packet containing the information you requested and an application for the Radio Shack Software Review Program.

SOFTWARE SUPPORT BENEFITS

If Radio Shack accepts your application, you are eligible for many outstanding benefits. The list of these benefits include:

- 1 - TECHNICAL INFORMATION GUIDES pertaining to the piece of hardware on which you are developing systems. This information will include material not generally released. You may also request technical information (upon your acceptance in the support program) associated with the hardware you are working with, that is not contained in the Technical Information Guides. If you have experienced problems and have determined their solutions, we will incorporate any information from you in the Technical Information Guides.
- 2 - SOFTWARE DEVELOPMENT GUIDES that are used by Radio Shack in developing our packages. This guideline illustrates procedures used in creating thorough documentation and functional smooth-flowing programs. The Software Development Guide outlines concepts that will assist you in making your product ready to market.
- 3 - INTERNAL LIBRARY ROUTINES used by the Radio Shack Software Development Group for standardization and high productivity of packages. Included in this category are:
 - the INKEY routines for Interpreter BASIC & Compiler BASIC
 - an assembler input routine that allows the programmer full control of the keyboard and graphics in COBOL on the Model II & Model III
 - source copies that are copied by our COBOL source programs when developing software
 - more advanced joystick routine for the Color Computer
 - an artifacting example routine for more sophisticated graphics on the Color Computer
 - assembler routines that are callable on the Pocket Computer 2
- 4 - TRSDOS LICENSING will be available to approved Software Support Program participants. This will allow you to duplicate & distribute TRSDOS and BASIC Interpreter system software on diskettes with your application software.
- 5 - PURCHASE TRSDOS SOURCE code copy for customizing utilities and making changes for your specific requirements. Any modification to TRSDOS is the responsibility of the individual who changes TRSDOS and releases the responsibility of support from Radio Shack. Calls to TRSDOS are preferred to ensure compatibility with future releases of TRSDOS.
- 6 - COMMERCIAL QUANTITIES of Compiler BASIC and COBOL Runtime packages at reduced rates will be made available. Along with a TRSDOS Duplication license, RUNTIME quantity purchases will allow you to sell completed packages at a reduced rate.
- 7 - ROM PAK COMPONENT information listing the materials and their availability from Radio Shack for assembling your own cartridges.
- 8 - MISCELLANEOUS INFORMATION which includes price listings for quantity purchases of computer hardware,

software, and peripherals. Also, a listing of programming aids and software design books that are felt to be above average is available.

- 9 - INFORMATION DISTRIBUTION LIST which will make available to you any information on patches to Radio Shack software, including TRSDOS. Also, you will be one of the first to know about any new operating systems or new versions of the operating systems. Under certain circumstances you may receive information on new hardware, so you may begin developing software before the new hardware is available in stores.

SOFTWARE REVIEW PROGRAM

The Radio Shack Software Review Program is designed to allow third party software firms, who are members of the Software Support Program, an opportunity to receive product exposure through a reference guide for reviewed software and in Computer Center product brochures.

SOFTWARE REVIEW BENEFITS

If Radio Shack approves your software package, it will be listed in a reference guide for reviewed software.

The reference guide for reviewed software will contain our evaluation on your package and the location where the package may be purchased. Separated by Application Area specified by you, the evaluations will contain the information specified on the application form, our positive opinion of the package, and any positive reactions from your user group.

On selected software packages that are extremely well developed and address specialized vertical markets which we cannot or do not currently address, Radio Shack will, at its expense and subject to your approval, prepare and print software brochures which will be provided to Computer Centers for distribution to customers.

BASIC RULES

- 1 - We will only consider software currently running on any TRS-80 CPU including: Model I, Model II, Model III, Model 16, Color Computer, Pocket Computer, and Pocket Computer 2.
- 2 - No Games will be evaluated in the Software Review Program.
- 3 - You must be an accepted member of the Software Support Program.
- 4 - Software and documentation flow with program must be tested. You will be asked to sign a Confirmation Form.
- 5 - You must be able to support any user problems with the package. A report containing the names of the packages approved, along with your phone number will be distributed to the Radio Shack Customer Service Group so they can refer customers to you.
- 6 - We recommend (but do not require) that your company establish an "800" number for support purposes, if you have not already done so. This number would also serve as a direct line to you for potential customer inquiries.
- 7 - Follow all documentation and software guidelines when submitting packages.
- 8 - Try not to have too many calls to the operating system, so that your system will be more compatible with future DOS's.

- 9 - Do not contact Radio Shack about your application or evaluation, this will only slow the response process.
- 10 - DO NOT submit ANY software or documentation until you are requested to do so by Radio Shack.
- 11 - Only submit disk software running under TRSDOS, until further notice, or cassette tape software that can be loaded by the appropriate TRS-80 computers.
- 12 - Software or documentation will not be returned.
- 13 - We maintain the right to refuse to review any program. In particular, we are not interested in reviewing products that are directly competitive to Radio Shack's offerings. We are most interested in vertical market applications and personal productivity programs.

NOTICE

The Radio Shack Software Support and Review Programs are provided for the assistance and convenience of those developing software for use on TRS-80 computers. Radio Shack shall have the right, at its sole discretion, to determine those persons and companies it will accept in the Program and to eliminate any person or company from the Program at any time. Radio Shack shall also have the right to discontinue distribution of any information it so desires and to discontinue the Programs, in part or in their entirety at its discretion. Radio Shack's acceptance of any application shall not be regarded as creating any contracted relationship between Radio Shack and the Applicant.

Software authors and vendors who want further information on Radio Shack's new Outside Software Support program are urged to write:

Radio Shack
Software Support/Review Program
1300 One Tandy Center
Fort Worth, TX 76102.

Color Computer Bugs, Errors, and Fixes

COLOR DISK DRIVE KIT (26-3022)

The end of the cable that connects the disk drive(s) to the disk controller is stamped with the word TOP in white letters. The TOP end of the cable is connected to the disk controller with TOP up. The two connectors which connect to the disk drives may or may not be similarly marked. Some of these cables are keyed so that they cannot be inserted upside down while others are multi-striped and not keyed. For those that are not keyed, the pin numbers (from 1 to 34) are visible on the edge of each connector. When connecting the cable, place the lowest pin numbers closest to the bottom of the drive.

When the disk drive cable is connected and the power is turned on, the drives should not start running. If they do then the cable is connected incorrectly. Before hooking or unhooking disk drives, make sure that all diskettes have been removed. They should also be removed before powering down the system.

TRIS-80 Christmas Story—Billy and Santa

by Jacque Farthing

Editor's Note: The CompuServe Information Service is one of the largest information and entertainment services available to owners of personal computers and computer terminals. With each issue of TRIS-80 Microcomputer News, various features of CompuServe will be discussed. The CompuServe Information Service is sold at Radio Shack stores nationwide and in Canada.

Little Billy laid out cookies and milk on the small table next to the towering Christmas tree, twinkling and reverent in its corner of the dark living room.

Eight-year-old Billy sank into the folds of the family easy chair, nearly burying himself, and snuggled up with the eider-down to observe his effect on the already cozy Christmas scene.

As Billy huddled in the warmth of the atmosphere, he gazed around the room lit only by the lights of the magnificent Christmas tree. The tree was positioned in the far corner and shone brightly of green, red, blue, and other flashes of light illuminating the vast array of ornaments dangling between branches and tinsel.

The presents were heaped one on top of the other and were strewn as far as the fireplace, whose ashes were still smoldering from the evening's fire. Billy glanced up at the silhouetted stockings lying in wait at the mantle.

Sitting back with a satisfied air, Billy's eyes rested on the familiar TRIS-80 Model III, now dark after a full night's use by various family members. As he stared at the blank screen Billy dreamed he was a top computer programmer, keying equations and computations for the space program and calculating major defense plans for the approval of the president and his defense committee.

As Billy imagined himself being congratulated by the president, he was interrupted by a sudden noise over near the tree. He raised his sleepy eyelids just enough to make out a figure by the fireplace, bent over with the cumbersome weight of an object resting on his back. The figure heaved the huge bundle onto the hearth and straightened himself out, dusting himself off and chuckling.

As Billy gazed across the dimly lit room, he rubbed his eyes and squinted to get a better look at what now occupied a large space in front of the fireplace. As Billy stared at the spot on the hearth, it turned just enough for him to see that it was a man engulfed in a long white beard and long white hair, pink cheeks and bright, glistening eyes. The man was dressed all in red with furry white trim on his collar, cuffs, and jacket front, which seemed to almost pop with his heaviness. He wore a stocking cap on his head with a little white ball on its end, and his pants were tucked into a high pair of jet black boots which

matched the massive black belt stretched tightly around his middle.

Billy leapt from his perch, no longer mildly observing the actions on the hearth, as Santa peered into his huge bundle and chuckled quietly to himself.

"Santa! What did you bring me, Santa?" Billy cried as he made his way across the room to get a closer look at Santa Claus.

"My, how you startled me, Billy. What are you doing up at this hour?"

"I'm waiting for you, Santa. I wanted to be here to help you with the gifts and give you milk and cookies. What did you bring me this year, Santa?"

As Billy eagerly awaited Santa's reply, Santa laughed and then pointed to the solemn computer and with a grin said, "Why Billy, I've brought you a world of information."

Billy gazed at Santa with puzzlement as he watched him rummage through the bulk of his huge bundle until he drew out a bright green light and set it in the palm of his hand.

"What's that?" Billy asked as he backed away from the light.

Billy could just catch a glimpse of Santa winking at him through the light as he turned toward the computer. As Billy moved closer he realized that the light was no longer coming from Santa but from the screen of the now bright computer.

"Sit down next to me, Billy, and I'll show you the growing world of videotex," Santa said.

As Billy gazed up at the terminal, Santa typed in his user ID number and secret password to the CompuServe Information Service.

"How would you like to talk to people all across the country on your TRIS-80 Model III?" Santa asked.

Billy continued to stare at the screen and slowly shook his head yes while Santa typed Go HOM-21 at the "!" CIS prompt.

"This is CompuServe's version of the CB, Billy. By using CompuServe you can talk through your computer to people all over the U. S. and Canada," Santa said.

"Once you enter with your ID number and password, you select "Communications." This automatically logs you on CB where you'll see many different conversations by people like Tootsie Roll, Cupcake or the Cookie Monster. These are their handles, just like those used by people on the radio version CB. Your terminal is your microphone and everything you type, followed by an <ENTER> is seen by other CBers on one of 40 channels," Santa said.

"But if I'm on channel 19 how do I change to another channel? Or how do I know if anyone's listening to me," Billy inquired.

"There are a few commands you give the computer to help you get around in CB," Santa said. "To change your channel, just type in the command /TUNE and you can send and receive messages on a different channel. /STATUS lets you see how many are "listening" to you, and the /MONITOR command lets you follow other conversations. You can even have your own private conversation with the /SCRAMBLE command," Santa told Billy.

"What else does CompuServe have, Santa? Can I play games on it?"

"CompuServe offers lots of games, Billy, plus shopping at home and Special Interest Groups."

"What's a Special Interest Group," Billy asked as he munched on a cookie from Santa's plate.

"A Special Interest Group is for people who share a common interest in something and want to swap information. CompuServe has groups for photography and music (Photo-80/Orch-80), groups for people who like ham radios (HAMNETS), groups for people who like to cook (the Cook's Underground) and even a group for the people who like to talk on CB. It's called CBIG. These groups share their information through an electronic bulletin board and some have a Feedback section where you can converse with hardware manufacturers about equipment, problems, or suggestions."

"Could I become a member of a special interest group?" Billy wondered.

"Of course you can. Most SIGs add you as a member when you log on and access them. Some have a fee, while others have certain equipment requirements."

"What kinds of things would be there for people like my Mom and Dad and brothers and sisters?"

"How about the ability to shop from your own home without getting up out of your chair? You can do that with Comp-U-Store, the electronic shop-at-home service. They have prices 10 to 40 percent lower than other places and can be accessed at CIS rates of \$5 for standard service and \$22.50 for prime service. Just access the main menu of Home Service and then the shop-at-home section on the subsequent menu. You can read information on products and prices and can order from a selection of over 30,000 brand name products including cameras, musical instruments, and stereo equipment."

"How do you do that?" Billy asked from the side of the now empty milk glass.

"With Comp-U-Store you can comparison shop among all the items. When you've decided on what you're going to buy, you go to the ordering section where you give information like your address, name, and method of payment."

"But how do they get the money to them if they're not in a store?" asked Billy.

"They can pay by credit card including the new no fee VISA which Comp-U-Store is offering to its members. With this feature, members don't have to pay a fee to use their VISA card. Comp-U-Store also has a Databasement where other specific items are further discounted, plus a weekly auction where items are bid on by members. As in any other auction, whoever gives the highest price gets the item."

"What kinds of games can I play," Billy asked, through a large yawn and bleary eyes.

"Oh, there are lots of games on CompuServe," Santa told Billy. "There are games like football and strategic games like chess. There're games of chance like roulette, blackjack, and craps, and popular games like golf. There's the Adven-

ture game where you battle snakes and dwarves in Colossal Cave in search of lost treasure. There's a series of adventure-type games such as Adventureland, Pirates Adventure, Mission Impossible, Voodoo Castle, Ghost Town, Pyramid of Doom, and Strange Odyssey. There's Megawars which is a real-time space battle where up to 10 people can play at a time. You're either a Colonist or a Kryon, and the object of the game is to attack your opponents' space ships and capture planets."

Santa chuckled and looked down to find Billy fast asleep by his lap with a large smile and a dried "milk mustache." He gently lifted Billy up into the chair in front of the computer and removed the empty plate of cookies and glass of milk. Santa quietly typed in a short message on the computer, gathered his huge bundle, and stepped onto the hearth. With the smoke still swirling around his windburned cheeks, he bad Billy a Merry Christmas and good night and disappeared up the chimney without a sound.

In the morning, Billy's family found him fast asleep in front of a black screen. Billy awakened and excitedly told them about his visit from Santa and about all the things he had shown him on the computer. Billy's mother wiped the dried mustache from Billy's lip and directed Billy to the pile of presents in front of the tree. Billy's brothers and sisters remarked how they used to sit and wait for Santa on Christmas Eve and told Billy that they used to dream that they actually saw Santa, too.

"But I did see Santa," Billy protested. "He showed me the games, and shopping at home, and special interest groups, and . . ."

But everyone just continued to open their presents amidst Billy's indignant protests. After they opened all the presents, Billy wandered over to the computer and switched it on. To everyone's surprise there was a "main menu" displayed on the screen and a little message at the bottom which read, "You have EMAIL waiting."

Remembering the instructions Santa had given him, Billy pushed the keys to read "Go EMA," and a short message distinctly appeared on the screen as Billy's family gathered around. The message read . . .

```
CompuServe          Page EMA-3

1 SANTA/MERRY CHRISTMAS

Last menu page. Key digit
or M for previous menu.

!1

CompuServe          Page EMA-5
25-DEC-82 01:00 Fr [#####]
Hope you enjoyed the world of information, Billy.

Thanks for the milk and cookies.

Merry Christmas to all.

SANTA
```

Questions and comments about the CompuServe Information Service can be sent to Richard A. Baker, Editor, Director, or Jacqueline A. Farthing, Assistant Editor, CompuServe Information Service, 5000 Arlington Centre Blvd., Box 20212, Columbus, Ohio 43220, or through Feedback main menu item 5, CompuServe User Information.

Now You Too Can Be a "Wall Street Wizard" . . .

George Maniscalco
Vice President, Standard & Poor's Corporation

I know a lot of people who dream about making their fortune in the stock market. After all, history shows that many people have already done so, and the opportunities for future success certainly seem to be plentiful enough. In fact, the only thing one really needs to know are the names of some stocks that will be going up (or down) in price. Then one could invest accordingly.

Obviously, and unfortunately, this knowledge is hard to come by. So, a great deal of effort and energy is expended in the fascinating arena of investment analysis. The basic idea, of course, is to find a way to forecast what is likely to happen to the price of a company's stock. At present there are a great many systems which are designed to help investors make decisions about whether and what to buy, sell, or hold. Essentially, these systems fall into one of two basic approaches: Fundamental or Technical analysis. Let us briefly look at these two approaches.

FUNDAMENTAL ANALYSIS . . .

In the area of fundamental analysis, the investor is typically looking for a situation which reflects a good buying (or selling) opportunity, usually based on some computation of current data in the context of a historical perspective or comparison. The information used here almost always contains some combination of fundamental financial data; that is, basic earnings data, dividend data, growth rates, sales or revenue data, balance sheet data such as assets or debt, and so on. The idea behind this approach is that, in the long run, the shares of companies with superior performance will outperform the shares of companies with less than superior performance.

Therefore, if one can find a company whose data shows movement towards superior performance, an increase in the price of that company's stock probably is not far behind. (Of course, there are other considerations that may affect how long a period one will have to wait. These include the overall economic situation, competitive considerations, and so forth.) The source for this fundamental information is often the company itself, frequently through the data it presents in its annual report. In addition, there are a number of financial publishing companies (e.g. Standard and Poor's) which collect, process, and re-distribute information about these companies. This information is made available to individual investors, and to the "professionals" at brokerage houses, investment banking firms, investment-advisory companies, and the investment departments of a large number of other companies.

TECHNICAL ANALYSIS . . .

On the other hand, technical analysis systems typically disregard all fundamental information. The concept behind these systems is that the rise or fall in the price of a company's stock is most simply expressed by the balance (or lack of it) between willing buyers and sellers of the stock. Further, this balance is considered to be best reflected in the "picture" (usually expressed as a chart or graph) of the stock's trading activity. A wide variety of displays exist for these technical systems. For example, there are bar charts, line charts, point and figure charts, ratio charts, moving average and volume indicators, logarithmic charts, etc., etc. Some of the buzzwords used to describe these pictures or formations may sound familiar: "head and shoulders", "flag", "pennant", "upside breakout", and so forth. In most cases, the incredible array of displays and visualizations comes from only four pieces of daily information: the high, low, and last price, and the trading volume.

Very often, the type of system that an investor uses will provide insights into his investment objectives and his investment time-frame as well. Patient investors will tend to use fundamental approaches. They'll focus more on long-term growth opportunities and avoid the temptation to try for a quick profit. Less patient investors are perhaps better described as "traders". Typically, they'll look for short-term situations and are usually less interested in long-range opportunities. Technical approaches tend to serve their needs better than fundamental approaches.

Actually, neither approach is inherently good or bad, right or wrong; and sometimes they can even work very well together. What is most important to the individual investor is to select the approach that makes him, or her, most comfortable. After all, the challenge itself is difficult enough: no need to make it worse by using an approach that does not match your style. Of course, some people use a third approach, sometimes referred to as the "dart-board" system. Since this approach is totally without discipline, we will not deal with it here.

Given all the different systems and approaches and types of data, it is easy to understand why many investors feel confused about the investment approach they should follow. To make matters worse, increasing amounts of basic data have become available to feed into this growing number of investment systems. For the average investor, dealing with all of this can often be a complex, time-consuming, and sometimes even frustrating job.

ENTER, AN APPROACH DESIGNED TO OFFER SOME HELP...

The advent of the microcomputer has made the task of investment analysis much easier to deal with. Although micros can be useful for a wide variety of applications, their use in this area is a particularly good fit. The main reason for this has to do with the fact that investment analysis systems usually require a lot of computational processing, and the process gets repeated on a regular basis as new data becomes available.

Two years ago, Standard & Poor's and Tandy combined forces to introduce a powerful, yet easy-to-use, investment software system. We called it Stockpak. This system was designed to help investors in two distinct and important ways. First, the Stockpak System includes a Portfolio Management capability to help you manage your investments. Second, Stockpak contains a unique and powerful Screening feature that helps you locate companies that meet your own personal investment criteria. Interestingly, both parts of the Stockpak System work against a disk-database which is updated each month.

Each of these capabilities is basic enough and important enough to be worthy of "stand-alone" status. Taken together, the combination of their features makes the Stockpak System extremely powerful, versatile, and useful. Let's take a closer look...

STOCKPAK'S PORTFOLIO MANAGEMENT CAPABILITY

For starters, we will make the assumption that anyone with investments (in anything) has a basic need for some way to keep track of those investments. After all, if for no other reason, you must at least have accurate records of your investment transactions for the Internal Revenue Service. A true portfolio management system is a well-defined database management system.

The Stockpak System includes a Portfolio Management capability that keeps track of stock, bond, and option purchases and sales. These functions are a minimum requirement for any portfolio management capability. In addition to these record-keeping functions, Stockpak handles the calculations to determine such items as Net Present Position, and Realized/Unrealized Gains or Losses. Stockpak also calculates the Rates of Return of each item in the portfolio against the S & P 500 Index and tells you when an issue will go from 'short-term' to 'long-term' status for income tax purposes, a feature which can be quite valuable. In addition, Stockpak sorts the data and delivers a report to either the screen or a printer. A very convenient feature allows the Portfolio diskette to accept information from the monthly database diskettes, about which we will have more to say later. As impressive as these features may be, they are only a small part of Stockpak's real power...

STOCKPAK'S SCREEN AND SELECT SYSTEM

By far, Stockpak's most impressive feature lies in its Screen & Select capability. This function permits users to create their own investment screens, then apply those screens to a database of companies. This part of the system works in conjunction with a database diskette (updated each month) which contains 30 financial facts on 900 actively traded companies. The 900 companies in the database include the 500 companies in the S & P 500 Stock Index, plus

400 other New York, American, and Over-the-Counter companies. Through the use of a unique Data Dictionary feature, users can create and maintain an additional 200 financial facts about these companies. Therefore, well over 200 items can be accessed in the process of creating a screen.

For example, you might tell the system to "find" only large companies, or only small ones... only companies with high earnings growth, or low P/E's, or high yields... only companies in specific industries or on certain exchanges... only companies with high sales or low debt... a high S&P Earnings and Dividend ranking. In fact, you can draw from over 200 information items to create a single screen and there is no limit to the number of screens you can apply. The system then goes to work to sort through all of the data on the database disk to find only those companies that exactly meet the criteria you specified. If you need to, you can easily re-define a screen if the previous criteria were either too loose or too tight. It is not hard to imagine how difficult, if not impossible, performing a task like this would be for someone without a computer.



Photo 1. Database Updates

Once you have reduced the 900 companies to a small group of your prime candidates, you can create a separate Spin-Off file to follow them more closely in the future. To round out the functions, the system also provides extensive reporting and writing facilities (both customized and pre-defined) for temporary or permanent retention of the results.

Until recently, these capabilities were only available to professional investors using time-sharing or large in-house computer facilities. Now, with Stockpak's software and database, individual investors stand a much better chance of finding those companies of special interest, based solely on the factors that they consider to be most important.

TOWARDS BETTER INVESTMENTS

The Stockpak System is currently available for both Model III and the Model I TRS-80's, and it comes in two parts. The first part includes all of the software described above (Portfolio Management, Screen & Select, and Report Writer), a sample database diskette, and complete documentation. The second part of the system is an annual subscription to updates of the database. These updates are produced once each month and mailed after the last trading day of the month. Also, included in part two is a subscription to Stockpak Strategist, a monthly newsletter that helps subscribers get even more value from using the system.

Part one is available from all Radio Shack stores and sells for \$49.95 (26-1507). Part two is available from Standard & Poor's and sells for \$200 per year. For more information, contact your local Radio Shack dealer or write to Standard & Poor's, c/o Stockpak, 25 Broadway, New York, N.Y. 100

Business Graphics—Analysis Pak

Effective business graphics made easy—that's the promise of a new software package called the Business Graphics-Analysis Pak. We have obtained this package from TimeWare Corporation, a Palo Alto software firm with many years experience in the field of business graphics. The package is a real bargain, offering functions similar to those of mainframe products costing thousands of dollars.

With the Business Graphics-Analysis Pak, you can display your data pictorially—in the form of line charts, bar charts, and scatter charts. The package is designed to run on the TRS-80 Model III microcomputer and requires two disk drives and 48K RAM. There is no expensive hardware to buy—charts can be produced on the printer you probably are using right now (dot matrix line printer V, VI, or VIII, or the daisy-wheel printer II) or on the TRS-80 multi-pen plotter.

Designed for business users at all levels, the Business Graphics-Analysis Pak is completely menu-driven—you can put it to work for you right away, without having to learn complicated operating procedures.

While the chart is displayed on the screen, you can edit the text by inserting, changing, deleting, or moving labels or captions. You can actually move text freely about the screen with the arrow keys—placing the text exactly where you want it, in or around the chart.



Photo 2

When you are satisfied with the chart on the screen, you can then produce a hard copy on the printer or pen plotter or save the chart in a file for future use.

If you want, you can assume more control over the chart format:

- Modify the size and shape of the chart—up to 11 inches square
- Adjust the range of the numeric scale and the format of the scale labels
- Provide a starting time label (such as a month or a quarter) that will be incremented automatically for each plot point or bar
- Modify the format of the curves, plot points, bars, or slices
- Request shading under curves or within bars, bar segments, or pie slices
- Add a frame to the chart

You can even design and save a customized chart format—this is handy for charts that are produced on a regular basis.

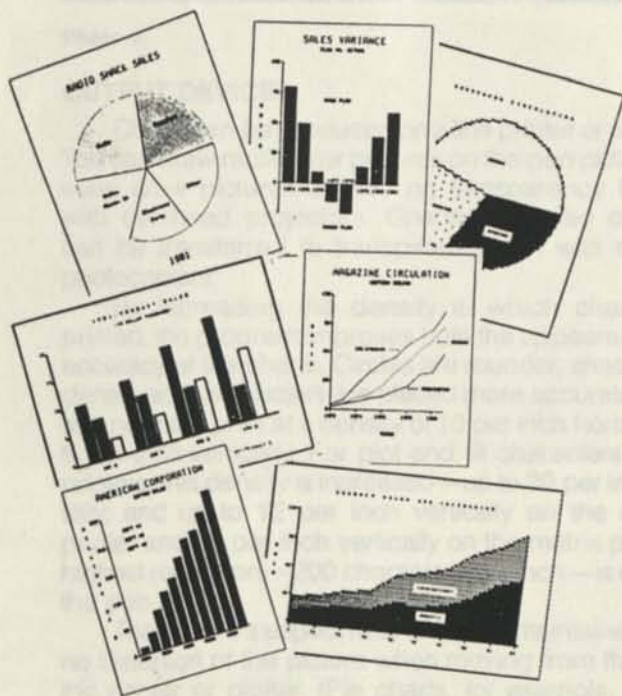


Photo 1

The Business Graphics-Analysis Pak sets up an attractive chart for you—automatically scaled, labeled, and laid out to suit your particular output device. Just select a chart type, supply your data, provide a title or two, and ask to see your chart displayed on the screen. After previewing the chart, direct it to your printer or pen plotter. It's as simple as that!

CHART DATA

Data for the chart can be entered at the terminal, brought in from a disk file, or generated by the program. Virtually any type of file can be used, including VisiCalc, SCRIPSIT, BASIC, and FORTRAN files. (Some types of files must first be converted to another format, using a utility that comes with the package.)

The Business Graphics-Analysis Pak provides extensive data handling capabilities. You can generate an arithmetic or geometric series of values. You can edit data, or manipulate it with various mathematical functions. You can project trends, compute moving averages, consolidate data, etc. You can display the data on the screen, print it on the printer, or save it in a file.



Photo 3

OUTPUT DEVICES

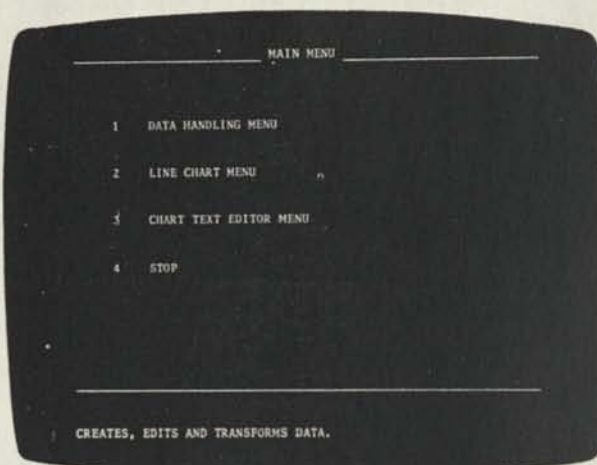
Charts can be produced on a line printer or a pen plotter. You can draw multi-color pictures on the pen plotter, and can even draw pictures directly on transparency film for use with overhead projectors. Charts produced on a printer can be transferred to transparency film with many office photocopiers.

By increasing the density at which characters are printed, the program improves both the appearance and the accuracy of the charts. Circles are rounder, shading is more dense, and data points are placed more accurately. Characters normally print at a density of 10 per inch horizontally and 6 per inch vertically. For plot and fill characters on the line printers, this density is increased—up to 20 per inch horizontally, and up to 12 per inch vertically on the daisy-wheel printer and 24 per inch vertically on the matrix printers. The highest resolution—200 characters per inch—is obtained on the pen plotter.

The picture's aspect ratio is always maintained—there is no distortion of the picture when moving from the screen to the printer or plotter. (Pie charts, for example, are always round, not egg-shaped!)

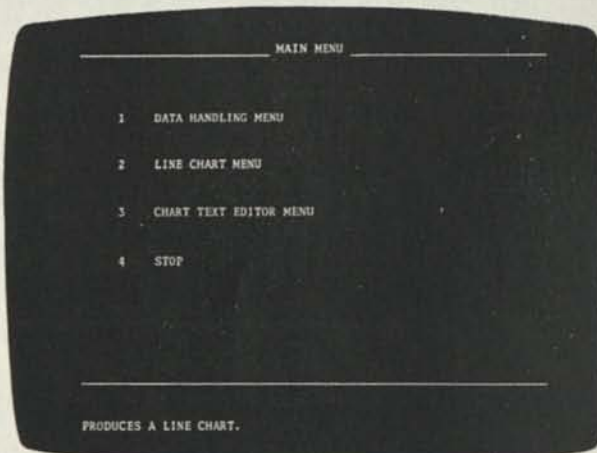
LEARN-AS-YOU-GO OPERATION

Because the Business Graphics-Analysis Pak is completely menu-driven, you can learn as you use it. Each time a menu appears on the screen, one menu item is blinking off and on; an explanatory message about that item appears at the bottom of the screen. If you press the <ENTER> key, the flashing item is selected.



Screen 1

If you point to another menu item (either by pressing corresponding number key or by moving to the item with arrow keys), that item starts to blink, and a new explanatory message appears at the bottom of the screen. You can select the menu item, if you wish, by pressing the <ENTER>



Screen 2

When you first start using Business Graphics-Analysis Pak, you can browse through the menus, reading all explanatory comments before deciding on your selection. As you get more familiar with the program, you can make selections immediately by pressing the number key <ENTER>, pausing to read the explanatory message when you need to refresh your memory.

STEP-BY-STEP EXAMPLE

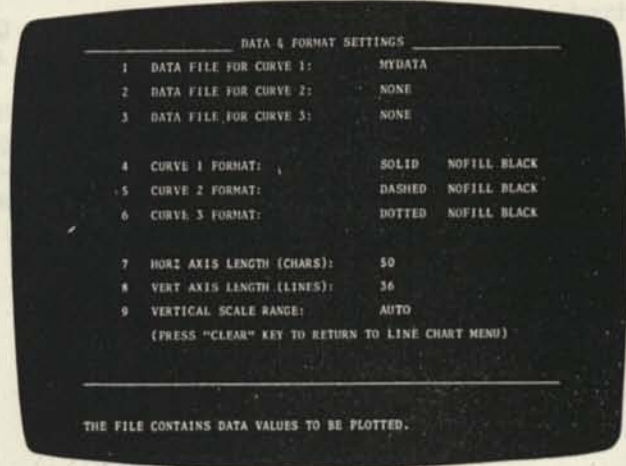
The following example—creation of a time plot—shows just how easy it is to use Business Graphics.

After loading the TRSDOS system disk and the line chart disk, you type in "TRSCHART" and press the <ENTER> to access the Main Menu. Pressing <ENTER> a second time transfers you to the Data Handling Menu.



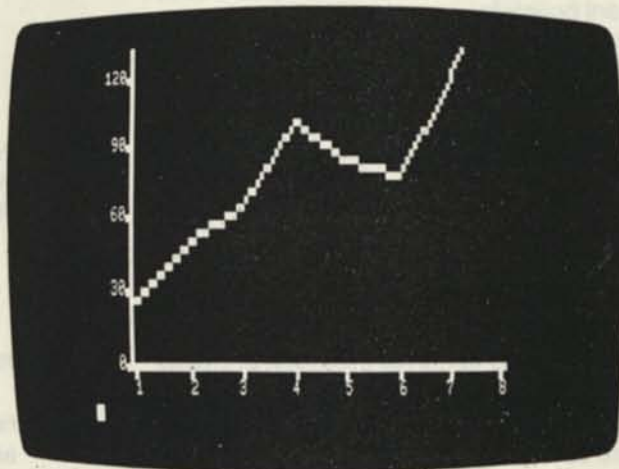
Screen 3

Since menu item 1 (ENTER DATA FROM KEYBOARD) is flashing, pressing the <ENTER> key brings up a new screen; you are then prompted to enter your data, one value at a time. Each value you enter is displayed on the screen. Data entry is terminated with the <CLEAR> key, which returns you to the Data Handling Menu. You select item 8 (SAVE DATA) and enter a file name; the program writes your data values to disk under that name. You then return to the Main Menu and select menu item 2, the Line Chart Menu.



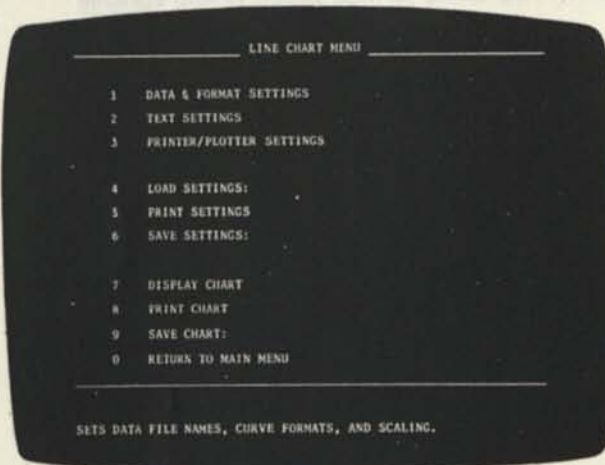
Screen 5

Press <CLEAR> to return to the Line Chart menu and select menu item 7 (DISPLAY CHART) to display your chart on the screen.



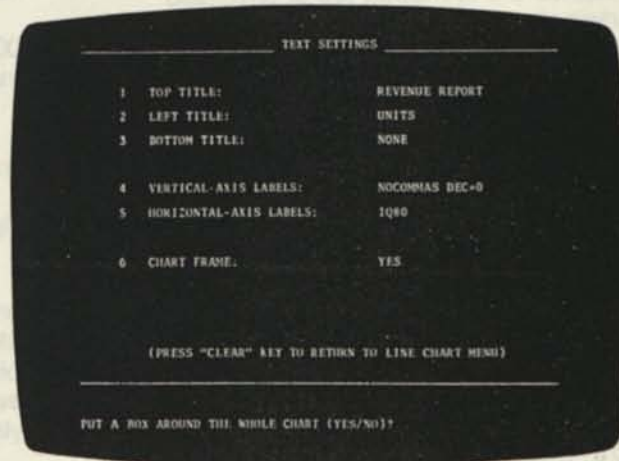
Screen 6

Pressing the <CLEAR> key restores the Line Chart Menu, and you select menu item 2 (TEXT SETTINGS). The Text Settings Menu is then displayed.



Screen 4

Since menu item 1 (DATA & FORMAT SETTINGS) is flashing, you need only press the <ENTER> key to be transferred to the Data and Format Settings Menu, where you enter the name of your data file for curve 1.



Screen 7

Here you enter a top title, a left title, and labels starting with 1Q80 (the first quarter of 1980) for the horizontal axis. You also request a frame for the chart.

The **<CLEAR>** key returns you to the Line Chart Menu. If you want, you can preview the chart on the screen. Selecting menu item 8 (**PRINT CHART**) causes the chart to be produced on the printer as shown below. (A daisy-wheel printer was used in this example.)

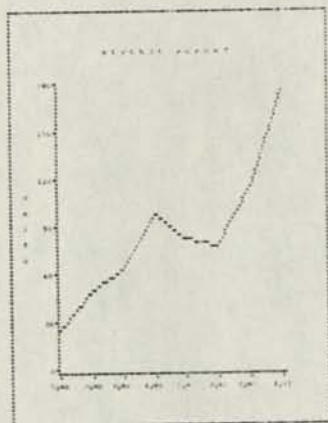


Photo 4

You can save the chart in a disk file (using menu item 9, **SAVE CHART**) for future use. Or you can save the chart settings (using menu item 6, **SAVE SETTINGS**) to be used with new data for another chart.

SYSTEM CAPABILITIES

Chart Types. Line charts, bar charts, pie charts, and scatter charts are available.

- Line charts may contain up to 3 curves, each based on up to 100 data points. Curves can be shaded from the curve line down to the bottom grid.
- Bar charts may contain up to 100 bars. Bars can be grouped in clusters of up to 3 bars or stacked (cumulated) with up to 3 segments. (Stacking or grouping reduces the maximum number of bars.) Bars or bar segments can be shaded.
- Pie charts may contain up to 12 slices, any of which can be shaded.
- Scatter charts (X-Y plots) can be based on up to 100 pairs of data values. By default, the data points are unconnected; you can request a connecting line, if desired.

Chart Style. You can let the program format the chart, or you can control any part of it directly.

- Chart width can range from 2 to 11 inches, chart height from 3 1/3 to 11 inches.
- Charts can be framed.
- Solid, dashed, or dotted lines can be used for curves, and any character can be specified for the data points.
- On a pen plotter, color can be specified for curves, plot points, bar or slice outlines, shading, and text entries.
- Automatic scaling is based on the data; alternatively, you can specify the range to be used.

Text. Screen editing is available for all text entries.

- The program provides for three titles, one each cen-

tered at the top, bottom, and left of the chart. Top and bottom titles can be enlarged.

- Automatic labeling is provided by default for line charts, bar charts, and scatter charts. Scale labels are based on the data; time periods are numbered from 1 to 4. You can control the format of the scale labels, and you can specify a starting time label (week, month, quarter, or year) for the time periods or bars, or suppress the labels altogether.
- In chart edit mode, the chart is displayed on the screen. You can then add, change, delete, or modify any text, supply special labels, etc. You can place labels anywhere in or around the chart and see exactly how they will look before you produce the picture.

Data Handling. Extensive data handling capabilities are provided.

- Data can be entered at the keyboard or retrieved from a disk file.
- An arithmetic or geometric series can be generated from a starting value and an increment or factor.
- Any set of data can be transformed by adding or subtracting a constant or multiplying or dividing each value by a constant; by consolidating the data (such as 12 months into quarters); smoothing the data with a moving average; or computing the logarithm of each value.
- Growth projections, either arithmetic or geometric, can be computed or a linear, quadratic, or exponential trend curve can be fitted to the data and projected for a number of periods.
- Data can be edited on the screen—values can be changed, inserted, or deleted, and the results are played immediately.


Program Output. Output can be handled several ways:

- Charts can be displayed on the screen, printed on a matrix or daisy-wheel printer, or drawn on the multiplotter. Charts can also be stored in a disk file for future use.
- Data values and chart settings can be displayed on the screen, printed on the printer, or stored in a disk file.

Program Documentation. Documentation is provided in the program itself and in a separate manual.

- Menus printed on the screen list the available choices at each stage of operation.
- For the current item, some additional information is displayed at the bottom of the screen.
- Diagnostic messages are provided when an error occurs. Error and warning messages appear at the bottom of the screen and flash off and on; the message must be acknowledged before the program continues.
- The *Business Graphics Reference Manual* provides an overview of the program, a tutorial for the new user, and detailed reference materials. A comprehensive index is included.

SEE IT IN ACTION!

Stop by your local Radio Shack and ask to see an automated demonstration of the Business Graphics—Business Pak. This will give you a better idea of how the program actually works. See how easy it is to design a chart, to produce and edit it right on the screen, and to produce the final chart on the printer or plotter. 

Model III Port Assignments

Robert E. Brown
2 Christina Drive East
Schenectady, NY 12303

(Text on Page 17)

MOD 3 PORT	FUNCTION	MOD 1 EQUIVALENT	MOD 3 PORT	FUNCTION	MOD 1 EQUIVALENT
E0H (224)	Maskable Interrupt Latch Bit reset = interrupt request (directs jumps to interrupt routines) Bit 0,1 - cassette interrupts Bit 2 - clock interrupt Bit 3 - I/O bus interrupt Bit 4 - RS232 transmit inter. Bit 5 - RS232 receive inter. Bit 6 - RS232 error interrupt Bit 7 - unused interrupt #7	37E0H (14304) Bit 0,1 - unused Bit 2 - communications interrupt Bit 3 - unused Bit 4 - unused Bit 5 - unused Bit 6 - disk controller interrupt Bit 7 - clock interrupt	F0H (240)	FUNCTION FDC Command/Status Register OUT: see Western Digital FD1771/3 manual - various bit patterns handle all disk commands including: Restore, seek, step, step in, step out, read data, write / data, read track, read address, write track, force interrupt IN: Bit 0 - set = busy Bit 1 - DRQ (set = data register full)* Bit 2 - lost data/missing address* Bit 3 - set = CRC error* Bit 4 - set = seek error/record not found* Bit 5 - set = head engaged* Bit 6 - set = write protected disk* Bit 7 - not ready *various status conditions are provided, depending upon the disk operation in progress	37ECH (14316) MOD 1 EQUIVALENT F0H Model 3 port
E4H (228)	Non-Maskable Interrupt Latch Bit reset = interrupt request Bit 5 - front panel reset inter. Bit 6 - motor time-out interrupt Bit 7 - disk controller interrupt	—No Direct Equivalent—	F1H (241)	Disk Track Update Register IN: current track NOTE: track selection is done by loading the data register (F3H) with the desired track and then issuing a SEEK command through port F0H	37EDH (14317) F1H Model 3 port
E8H (232)	RS232/Modem Status Register OUT: any byte resets the interface IN: Bits 0,1,2,3 - unused Bit 4 - RI (ring indicator) Bit 5 - CD (carrier detect) Bit 6 - DSR (data set ready) Bit 7 - CTS (clear to send)	—Same as Mod 3 (port E8H)—	F2H (242)	Disk Sector Select Register IN or OUT: 8-bit parallel register for storing/reading desired sector	37EEH (14318) F2H Model 3 port
E9H (233)	RS232 Baud Rate Select and Switch Sensor OUT: Bits 0-3 - select receiver baud rate Bits 4-7 - select transmit baud rate IN: Bits 0,1,2 - ignore Bit 3 - parity (set = enabled) Bit 4 - stop bits (set = 2, reset = 1 bit) Bits 5,6 - word length (00 = 5, 01 = 6, 10 = 7, 11 = 8) Bit 7 - parity (set = even, reset = odd)	—Same as Mod 3 (port E9H)—	F3H (243)	Disk Data IN and OUT: 8-bit parallel port	37EFH (14319) F3H Model 3 port
EAH (234)	UART Control/Status Register OUT: Bit 0 - DTR (data terminal ready)* Bit 1 - RTS (request to send)* Bit 2 - Break (disable transmit data) Bits 3-7 - see IN assignments, port E9H IN: Bits 0-2 - unused Bit 3 - set = parity error Bit 4 - set = framing error Bit 5 - set = overrun Bit 6 - set = data sent (register empty) Bit 7 - set = data received (register full) *The Radio Shack Mod 3 service manual lists these two-bit assignments reversed from the arrangement shown here.	—Same as Mod 3 (port EAH)—	F4H (244)	Disk Drive Select OUT: Bit 0 - set = drive #0 Bit 1 - set = drive #1 Bit 2 - set = drive #2 Bit 3 - set = drive #3 Bit 7 - set = double density IN: Bits 0-3,7 - ignore Bit 4 - reset = side 0, set = side 1 Bit 5 - set = write precompensation engaged Bit 6 - set = generate waits	37E1H (14305) F4H Model 3 port IF in Double-Density operation (using write precompensation)
EBH (235)	RS232 Data Register OUT: 8-bit parallel transmit data IN: 8-bit parallel receive data	—same as Mod 3 (port EBH)—	F8H (248) or 37E8H (14312)	Line Printer Status/Data Register OUT: 8-bit parallel data port IN: Bits 0-3 unused Bit 4 - set = printer fault Bit 5 - set = device selected Bit 6 - set = out of paper Bit 7 - set = busy	37E8H (14312) F8H Model 3 port
ECH (236)	Miscellaneous Controls Bit 1 - cassette motor (set = on) Bit 2 - 32 CPL mode (set = 32 CPL) Bit 3 - Kana characters (reset = Kana) Bit 4 - Enable I/O bus (set = enabled) Bit 5 - Enable video waits (set = enabled) Bits 0,6,7 - Unused	No direct equivalent, See FFH (255) described later . . .	FFH (255)	Cassette Status/Command Register OUT: Bits 0,1 - output signal 00 = .85v, 01 = .46v, 10 = 0.0v Bits 2-7 unused IN: Bit 0 - set = 1500 baud Bit 1 - set = motor on Bits 2-5 - see same bits port 0ECH Bit 6 - unused Bit 7 - set = 500 baud —No Direct Equivalent—	Port FFH (255) OUT: Bits 0,1 - output signal Bit 2 - set = motor on Bit 3 - set = 32 CPL display Bits 4-7 - ignore IN: Bit 0 - ignore Bit 1 - ignore Bits 2-5 - ignore Bit 6 - video display status Bit 7 - cassette input 37E4H (14308) OUT: 01 = Select Cassette #1 02 = Select Cassette #2

(Continued on Page 17)

Holiday Fun with Your TRS-80

Well, they tell me that it's time for Christmas once again. It's a bit hard to get into the Holiday Spirit when it's still 102 degrees outside. (In case you haven't picked up on it yet, these articles must be written several months in advance. This one is being written in late August/early September.) Anyway, we're going to try to give you a few new ideas for having fun with your equipment during the Holiday Season.

The first program we'd like to share with you is for the PC2. It includes a printer routine which may be omitted if you haven't gotten your printer yet. I'm not going to tell you what will happen; I'll just let you type it in.

```

2 GOSUB 100
6 WAIT 100
7 PRINT "SEASONS GREETINGS !!!"
8 CLS
9 WAIT 0
10 GCURSOR 1
20 GPRINT 32; 48; 120; 48; 32; 0; 0;
21 GPRINT 16; 24; 28; 30; 127; 30; 28; 24; 16; 0;
0
22 GPRINT 32; 48; 120; 48; 32; 0; 0; 0; 0; 0;
0; 0; 0; 0;
30 GPRINT 63; 127; 127; 112; 48; 0; 0; 0; 0; 0;
0; 0; 0; 0; 0;
40 GPRINT 124; 124; 125; 126; 125; 124; 124; 0;
0; 0; 0; 0; 0; 0; 0;
50 GPRINT 95; 81; 113; 82; 84; 84; 116; 88; 80;
88; 0; 0; 0; 0;
60 GPRINT 68; 56; 88; 24; 88; 61; 70; 1; 0; 0; 0;
0; 0; 0; 0; 0; 0;
70 GPRINT 124; 126; 6; 6; 12; 60; 0; 0; 0; 0; 0;
0; 0; 0; 0;
80 GPRINT 64; 96; 127; 127; 96; 68; 36; 24; 0; 0;
0; 0; 0; 0; 0; 0; 0;
90 GPRINT 32; 48; 120; 48; 32; 0; 0;
91 GPRINT 16; 24; 28; 30; 127; 30; 28; 24; 16; 0;
0;
92 WAIT 100
93 GPRINT 32; 48; 120; 48; 32
94 WAIT
95 GOTO 500
100 GRAPH
105 COLOR 2
110 GLCURSOR (100,-500)
: SORGN
120 LINE (0,500) - (-70,450) - (-20,460) -
(-80,410) - (-25, 420) - (-90,370) -
(-30,380)
130 LINE - (-100,330) - (-15,340) - (-15,300) -
(15,300) - (15,340) - (100,330)
140 LINE - (30,380) - (90,370) - (25,420) -
(80,410) - (20, 460)
150 LINE - (70,450) - (0,500)
180 FOR Z = 1 TO 9
185 COLOR RND(3)
190 READ A, B, C, D, E, F, G, H, I, J, K, L, M,
N, O, P, Q, R
200 LINE (A,B) - (C,D) - (E,F) - (G,H) - (I,J)
210 LINE - (K,L) - (M,N) - (O,P) - (Q,R)
220 NEXT Z
230 DATA -34, 460, -36, 458, -38, 456, -36, 454,
-34, 452, -32, 454, -30, 456, -32, 458,
-34, 460
240 DATA 20, 360, 18, 358, 16, 356, 18, 354, 20,
352, 22, 354, 24, 356, 22, 358, 20, 360
250 DATA 0, 480, -2, 478, -4, 476, -2, 474, 0,
472, 2, 474, 4, 476, 2, 478, 0, 480
260 DATA -50, 420, -52, 418, -54, 416, -52, 414,
-50, 412, -48, 414, -46, 416, -48, 418,
-50, 420
270 DATA 24, 440, 22, 438, 20, 436, 22, 434, 24,
432, 26, 434, 28, 436, 26, 438, 24, 440
280 DATA -20, 400, -22, 398, -24, 396, -22, 394,
-20, 392, -18, 394, -16, 396, -18, 398,
-20, 400
290 DATA 64, 412, 62, 410, 60, 408, 62, 406, 64,
404, 66, 406, 68, 408, 66, 410, 64, 412
300 DATA -70, 344, -72, 342, -74, 340, -72, 338,
-70, 336, -68, 338, -66, 340, -68, 342, -70,
344
310 DATA 36, 400, 34, 398, 32, 396, 34, 394, 36,
392, 38, 394, 40, 396, 38, 398, 36, 400
320 GLCURSOR (-100,200)
: SORGN
330 TEXT
335 CSIZE 3
340 LPRINT "MERRY CHRISTMAS!"
345 LF +3
346 LPRINT "SEE DISPLAY . . ."
347 LF +4
350 RETURN
500 READ A, B
510 IF A = 0 THEN FOR D = 1 TO 6
: NEXT D
520 IF A = 255 THEN END
530 BEEP 1, A, B
540 GOTO 500
550 DATA 70, 450, 50, 300, 50, 300, 50, 660, 45,
600, 38, 400, 38, 400, 38, 1100, 0, 0, 38,
400
560 DATA 44, 375, 38, 400, 36, 750, 53, 450, 44,
400, 50, 300, 50, 700, 0, 0, 70, 450, 50,
300, 50, 300
570 DATA 50, 550, 45, 650, 38, 400, 38, 400, 38,
1100, 0, 0, 38, 400, 44, 375, 38, 400, 36,
750
580 DATA 53, 450, 44, 400, 50, 300, 50, 700, 0,
0, 31, 500, 31, 450, 38, 400, 27, 900, 31,
500, 31, 400
590 DATA 36,-450, 36, 800, 0, 0, 36, 500, 36,
400, 44, 400, 31, 800, 36, 500, 36, 400, 38,
500, 38, 900
600 DATA 0, 0, 70, 450, 50, 300, 50, 300, 50,
660, 45, 600, 38, 400, 38, 400, 38, 1100, 0,
0, 38, 400
610 DATA 44, 375, 38, 400, 36, 750, 53, 450, 44,
400, 50, 300, 50, 1000, 255, 0

```

If you do not have the Plotter/Printer-Cassette Interface you should omit lines 2, 95, and 100-350. Now I hear you say "But I have a Plotter/Printer for my Model I/II/III/16/Color Computer. What do you have for me?" To that I give you the following program for the CGP-115 Color Graphics Printer.

```

10 CLS
20 LPRINT CHR$(18)
30 LPRINT "M240,-700"
   : LPRINT "I"
40 LPRINT "C3"
   : LPRINT "M-8,498"
50 LPRINT "D0,519,8,498,-10,510,10,510,-8,498"
60 LPRINT "M0,500"
70 LPRINT "C2"
80 LPRINT "D-140,400,-40,420,-160,320,-50,
   340,-180,240,-60,260"
90 LPRINT "D-200,160,-30,150,-30,40,30,40,30,
   150,200,160"
100 LPRINT "D60,260,180,240,50,340,160,320,
   40,420"
110 LPRINT "D140,400,0,500"
120 C=RND(4)-1
   : IF C = 0 THEN GOTO 120 'NO BLACK
130 LPRINT "C"; C
140 READ A, B
   : IF A = 0 THEN GOTO 220
150 LPRINT "M"; A; ", "; B
160 LPRINT "D"; A; ", "; B; ", "; A - 4; ", "; B -
   4; ", "; A; ", "; B - 8
170 LPRINT "D"; A + 4; ", "; B - 4; ", "; A; ", "; B
180 LPRINT "D"; A - 3; ", "; B - 5; ", "; A + 3;
   ", "; B - 1; ", "; A - 3; ", "; B - 1
190 LPRINT "D"; A + 3; ", "; B - 5; ", "; A - 3;
   ", "; B - 5
200 GOTO 120
210 DATA -40, 460, 1, 350, -100, 280, 100, 270,
   -50, 240, -150, 180, 80, 190, 170, 170, 0,
   0
220 READ C
   : LPRINT "C"; C
230 READ A, B
   : IF A = 0 THEN GOTO 290
240 LPRINT "M"; A; ", "; B
250 LPRINT "D"; A; ", "; B; ", "; A + 80; ", "; B;
   ", "; A + 80; ", "; B - 50
260 LPRINT "D"; A; ", "; B - 50; ", "; A; ", "; B
270 GOTO 220
280 DATA 2, -190, 100, 3, 140, 100, 0, 0, 0
290 LPRINT "M-190,75"
   : LPRINT "C3"
300 LPRINT "D-190,75,-110,75"
310 LPRINT "M-150,50"
   : LPRINT "D-150,50,-150,100"
320 LPRINT "C2"
   : LPRINT "M160,50"
330 LPRINT "D160,50,160,100"
340 LPRINT "M140,60"
   : LPRINT "D140,60,220,60"
350 LPRINT "M-240,-10"
360 LPRINT "S4"
   : LPRINT "Merry Christmas!"
370 LPRINT "A"
380 LPRINT " "

```

For users of the Color Computer, change all occurrences of LPRINT to PRINT #-2. No further changes should be required. If you use a Model II, you should use the error trapping routine given in the printer manual.



Computer Customer Services Address and Phone Number

8AM to 5PM Central Time
Computer Customer Services
400 Atrium, One Tandy Center
Fort Worth, Texas 76102

Model I/III Business Group(817) 390-3939
Model II/16 Business Group(817) 390-3935
Languages and Compilers Group(817) 390-3946
Color and Pocket Computer Group(817) 390-3944
Hardware and Communications Group ... (817) 390-2140
Educational Software(817) 390-3302
Games, Books, and News Product Info ... (817) 390-2133
Newsletter Subscription Problems(817) 870-0407

Model III Ports (From Page 15)

My disk head cleaner program, which appeared here a few issues ago, resulted in a flood of correspondence from some very nice people who wanted to know more about Model III ports and their use. To answer the questions and to improve my own understanding, I decided to do some research, and the results are presented in the following charts.

One common misconception, evident in the letters, was a tendency to equate INP with PEEK and OUT with POKE. Ports are not memory locations and are massaged and reset by the central processor at very high speeds. It is very unlikely that one could OUT a value to a port and find the same value with INP. In fact, with only a very slight delay, a port can be loaded with data via the OUT command and then read immediately using INP, providing entirely different information having to do with a device's status. Perhaps the confusion arises because the Model I was "memory-mapped" and communication was with what seemed like memory addresses (actually, they were not RAM addresses but 'addressed devices'). Early Model I computers used only one port—FFH (255) for the cassette. Later, additional ports were claimed by the RS-232 interface, and these same ports are used in the Model III. In fact, the Model III did away with memory-mapping altogether except for the display which is truly memory-mapped in RAM.

The Z-80 processor is capable of communicating with 256 'ports' which are usually assigned to peripheral devices. It establishes the link by using the address bus and the data bus, sending out 8-bits of a port address and a signal saying, "I have an I/O address here." It is the responsibility of the peripheral device to recognize its unique I/O address and to receive or transmit data. The charts which follow list the ports known (by me) to be assigned in the Model III and their approximate memory-mapped counterparts in the Model I.

The ports assigned to disk operations must be carefully addressed as regards timing and sequence. So, with the exception of simple select, restore, and seek operations (as were used in my head cleaning program), they are not of much use to the BASIC programmer. The ports assigned to the RS232, line printer, and cassette recorder, on the other hand, are very amenable to use from within BASIC.

Profile III + : Setting Up a User Menu

Profile
The small Computer Company

Written for the users of PROFILE II, PROFILE+ and PROFILE III+

PROFILE Editor
P.O. Box 2910
Fort Worth, TX 76113-2910
By Ivan Sygoda, Director, Pentacle

Once you've set up your PROFILE III+ data base, your last step is to define your user menu. Creating a user menu is optional but well worth the effort because of the increased speed and ease with which you can manipulate your files. The manual which came with your copy of the program tells you how to set up the user menu, and you should refer to the manual for general instructions. In this column, we'll explain some of the finer points of menu creation, especially substituting the user menu for the regular Runtime menu.

MENU NAMES

Setting up a menu is easier than it looks at first. The trickiest part is the menu itself. Some of the confusion over names stems from the fact that there are three independent entities that can be construed as menu names: 1) the heading for your menu, which is actually only the text that serves as a title at the top of the menu screen, 2) the six-character entry you can use to code or identify the menu, which appears only during menu creation, never to be seen or used by the program again, and 3) the real menu name, which is listed in the DIRectory and which is the file called by PROFILE III+ when you ask for the user menu.

More confusion stems from the name changes you must make to substitute the user menu for the Runtime menu supplied by Radio Shack. (Why substitute? If you do, your user menu, rather than the regular Runtime menu, appears whenever you reset the computer or finish an operation.) The original Runtime menu is RM/CMD (CM/CMD is the Creation menu). The reason menu names have to be juggled is that PROFILE III+ knows only how to call for something named RM/CMD when it wants a menu. Consequently, to substitute a user menu for the Runtime menu, you must take these steps:

- 1) At definition time, the user menu-to-be must originally be called anything but RM/CMD. We suggest MI/CMD, although any permissible TRSDOS file name will do.
- 2) When the user menu has been completed and checked, the Runtime menu (RM/CMD) must be renamed (with the RENAME command). Again, any name but RM/CMD is appropriate. The manual's suggestion, HOLD/CMD, is as good as any other and easy to remember.
- 3) Finally, since there must be something called RM/

CMD for PROFILE III+ to call, RENAME your user menu RM/CMD.

IF ONE IS GOOD, MORE IS BETTER

You'll appreciate the ease with which your user menu takes you right into your data base. You'll also soon realize, especially if you have formatted a variety of screens, reports, and labels, that the eight user menu choices allowed may not be enough. Indeed, you will probably have only seven choices, since it's a good idea to use one of the eight for calling the PROFILE III+ Runtime menu (now called HOLD/CMD, remember?).

The solution is simple: make a multi-page user menu! The first page (or screen) of your menu is the new RM/CMD. Define another user menu—select "Define User Menus" from the Creation menu—giving it the file name "M2/CMD" or something similar. Then add M2/CMD to RM/CMD as a menu entry. You should also add an entry on the page 2 menu, M2/CMD, for the page 1 menu, RM/CMD.

Another option is creating a three (or more) page menu, each page called from another menu or even from a menu of menus. You could have one page for inquiry options, one for report options, and one for label options. This would give you enough room to have separate choices, for instance, for "Report 1, Normal Selection" and "Report 1, Extended Selection."

Each user menu occupies six granules of disk space but can be placed on any available drive.

DON'T PASS OUT: PASS PARAMETERS

The PROFILE III+ manual discusses ways to have the computer enter keystrokes automatically when a certain menu selection is made. For instance, something like "EFC8 A (YOURFILE, 1, PRINT REPORT FROM FORMAT ONE)" will take you all the way to the sorting prompts.

It's possible to get even further into your program by using a DO file to pass additional keystrokes to PROFILE III+. To continue with the example above, if you consistently use report format 1 to make zip-coded sorts, you could construct the following "DO" file using the TRSDOS "BUILD" command:

Instructions:	Comments:
BUILD ZIPSORT	"ZIPSORT" is the name of the DO file.
<ENTER>	
EFC8 A(YOURFILE,	This passes the file name, format and

1,PRINT REPORT heading to the DO file.
FROM FORMAT ONE)

<ENTER>

3 <ENTER>

Your zip code field number is 3; press
<ENTER> to record it.

<ENTER>

To use the entire field length for
sorting.

<ENTER>

To select all records.

<BREAK>

To record the DO file; "TRSDOS
Ready" returns.

Then replace "EFC8 A (YOURFILE,1, PRINT REPORT
FROM FORMAT ONE)" with "DO ZIPSORT" on your user
menu:

1 Print Report 1 in Zip Code Order DO ZIPSORT

Pressing "1" on the menu now prints the entire report
without any user intervention.

BUILD A STAIRWAY TO THE STARS

There is almost no limit to the operations you can run
automatically with such a DO file. Suppose the above report
format were a 132-character format, but your dot matrix
printer has a default setting of 80 characters. Your printer can
be initialized automatically for 132-character reports (assuming
it is capable of 132 column operations) by adding the
following commands to the beginning of the ZIPSORT file
(Please note that this program applies to a Radio Shack Line
Printer IV. Consult your own printer's user manual for equivalent
commands.)

Instructions:

Comments:

BUILD ZIPSORT

You can rewrite the DO file to include
the new instructions.

<ENTER>

PAUSE Printer Ready,

on Line? <ENTER>

This prompts the operator to check the
printer; the message "Press
<ENTER> to continue" will follow
"Printer Ready ...".

BASIC <ENTER>

Go into BASIC.

<ENTER>

Press <ENTER>—you don't need to
open files.

<ENTER>

Press <ENTER> again to skip the
memory size prompt.

LPRINT

CHR\$(27);CHR\$(20)

<ENTER>

This is the code to change the default
to 132 characters.

<ENTER>

Press <ENTER> so that BASIC knows
to continue.

CMD"S" <ENTER>

This command returns you to
TRSDOS.

EFC8

A(YOUR FILE ...

Continue as before.

The final step in automating your menu is to add an "AUTO
RM" command to your Runtime diskette. When you boot up,
page 1 of your user menu will automatically appear on your
screen after the data and time prompts are answered. It is
then possible to print an entire 30-page report with only two
keystrokes—the "1" to start the operation from the menu,
and <ENTER> when the printer is ready.

MAKE YOUR DATA BASE USER-FRIENDLY

You'll soon become adept at making PROFILE III+ do
what you want it to do. The next level of sophistication is
getting the people in your office who use your program to use

it the way you had in mind. A well-thought-out user menu
system will make the program more transparent to your
colleagues, which in turn will increase efficiency. Here are a
couple of pointers and ideas.

Group functions logically, in chronological order or ac-
cording to priority. For instance, don't place "Add Records"
between "Print Report 1" and "Prepare Mailing Labels".

If different operators perform different functions, give
each one his or her own menu page, even at the cost of
duplicating selections.

PROFILE III+ is all about improving work flow. Keep
this foremost in mind as you design your user menus.

*Profile Editor's Note: This is Mr. Sygoda's third selection
in a series of 'how-to' PROFILE III+ articles. Other articles in
the series will be published over the next few issues in this
column. We hope that you enjoy this new feature, and we look
forward to your comments and questions on PROFILE III+.*

*Pentacle is a New York City-based non-profit service
organization specializing in administrative services for per-
forming art groups.*

PROFILE PLUS: WHAT TO DO WHEN YOU GET AN ERROR

PROFILE PLUS is a nearly fool-proof program. Nonethe-
less, our readers tell us that sometimes they do get error
messages and would like help in figuring them out. So, in
response to your queries, part two of this PROFILE column
will be our suggested PROFILE PLUS troubleshooting
procedures.

We recommend, first of all, that you keep a log with a
section on problems and troubleshooting. A troubleshooting
log is very helpful reference material for anyone using your
PROFILE PLUS system.

This log should have five columns of entries: "What
Program Were You Using?" when you saw the error; "Name
of File and Extension"; "Error Number"; "Any Special Cir-
cumstances?"; and "Date."

ERROR TYPES

In general, there are two kinds of errors: operating errors
and system errors.

When the program doesn't do what you expected it to
do, that's an "operating" error. For example, you thought you
indexed a data base on field 1, "Last Name." When you go to
update records, however, you find that the field shown on the
index screen is field 2, "First Name." You've made a mis-
take—what is it?

A "system" error gives you an error message and drops
you into system mode—the line "TRSDOS READY" appears
beneath the message. For help with these errors, look in the
following section, "System Errors and Solutions."

HOW TO HANDLE ERRORS

Your first response to an error message should be to
decide if you understand why you got it and if you can fix it
without help.

If not, you should write the error message down in your
troubleshooting log noting:

- 1) the PROFILE PLUS operation,
- 2) the name and extension of the file you were using,
- 3) the error number, and
- 4) any unusual circumstances surrounding the error
condition—a brown-out, for instance.

Then look for the reason you got the error. Once you get an explanation, you should put it in the log also.

OPERATING ERRORS AND SOLUTIONS

To find out why you're seeing operating errors, turn to the part of the PROFILE PLUS manual that describes the operation you're in, and look through the instructions. Watch especially for notes—your problem may be explained there. If not, check that you did each step of the operation correctly and used the proper syntax.

SYSTEM ERRORS AND SOLUTIONS

There are two ways to find explanations for system errors. One is through the computer itself. The other is through reference materials like your manual or this column.

COMPUTER

If the program only shows you the number of the error, type in the word error and the error number when you see the message "TRSDOS READY" and press <ENTER>. For instance, type:

ERROR 4

Press <ENTER>. The computer responds:
CRC ERROR DURING DISK I/O OPERATION
(See below for an explanation.)

The error messages in PROFILE PLUS programs always include descriptions of the problem and the name of the file in which the error occurred. For instance, in the PROFILE PLUS Inquire, Update, Add program you might see this error:

DISK ERROR CODE 28 OCCURRED ON FILE INVOICEX/
IY1 ATTEMPT TO READ PAST EOF


You probably tried to use an index with no records in it. This would happen if you had started to index the INVOICEX data base earlier, then changed your mind and broke out of the index operation before the computer could write any records into the index file.

REFERENCE

A list of operating system error messages is available in the Disk Operating System Reference section of your TRS-80 Model II Owner's Manual, under "Technical Information".

Some system errors are more common than others. Following is a list of errors, possible reasons for their occurrence, and what you can do about them.

- 4 CRC ERROR DURING DISK I/O OPERATION
The computer, in attempting to read data off a diskette, discovers that it cannot verify the data. An error 4 may indicate a magnetically—damaged diskette. Error 4s are sometimes transient. Try the question a few times. If you continue to get the error, go into TRSDOS, open the drive door, take the diskette out, and put it back in to reseal it. You might also try moving the diskette to another drive. If any of these procedures work, immediately back up the problem diskette onto a fresh diskette. Watch for error messages—if there are no errors, then the information was recovered and your new backup diskette can be used. Do not use the original diskette. Label this diskette as damaged and remove it from use. If errors **do** appear during the backup operation, neither the original diskette nor the new backup diskette is usable. You will have to go to an earlier set of backups—from this point on, use these diskettes.

- 5 DISK SECTOR NOT FOUND
The computer tried to read a specific sector on a diskette and could not. Possibly a mechanical problem; more likely, the sector was magnetically damaged or erased. Follow the error 4 instructions.
- 6 ATTEMPT TO OPEN A FILE WHICH HAS NOT BEEN CLOSED
This probably means that the diskettes were removed from the system or the computer was reset without exiting PROFILE PLUS first. Simply try the operation again—in the process of giving you the error message, PROFILE PLUS closes the files.
- 8 DISK DRIVE NOT READY
Check to see if the drive door is open, if the specified drive has a diskette in it, or if the diskette is in backwards.
- 15 DISK IS WRITE PROTECTED
The silver tab over the notch on the diskette has come off. Also check that you have the right diskette in the drive—a CREATION or master diskette is write-protected for a reason.
- 19 IMPROPER FILE NAME (FILESPEC)
You may have mis-typed a file name or its extension, or some other specification—drive number, password, etc.
- 24 FILE NOT FOUND
You have misspelled the file name, have the wrong diskettes in the system, or the data base or parts of the data base (MAP, KEY, DAT, DA2, DA3, etc.) have been erased. PROFILE PLUS tells you the name of the missing data base. Also check that you typed the file name in all upper-case letters, or upper-and lower-case letters, as you defined it.
- 25 FILE ACCESS DENIED DUE TO PASSWORD PROTECTION
The file, format, or screen you asked for is protected by a password, or you have misspelled the password.
- 28 ATTEMPT TO READ PAST EOF
EOF is End Of File. This error usually occurs if a format or index was started but not finished—there are no records in the file. For instance, you opened the "Define Math Formulas" operation, but defined no formulas. You must either return and finish the operation or format, or kill the incomplete file.
- 31 PROGRAM NOT FOUND
You have the wrong diskettes in the system, or the program has been erased.
- 40 SEEK ERROR
Errors 4, 5 and 40 may all mean that the diskette has been magnetically damaged. Error 4 means an error on a sector; error 5 means the computer couldn't find the sector; error 40 means it couldn't find the track. To correct or to find the nature of the error, try your operation again, try a different drive, and if worse comes to worst, use your backups. Notice if any of these errors occur during backup. If you try to use the diskettes with these errors, you'll have unreliable results.
- 42 PRINTER NOT READY
Check to see if the printer is turned off, off-line, disconnected, if there is no paper in the printer, or if the ribbon or ink has run out. 

1982 Cumulative Index

Assembly Language Programming

by William Barden, Jr.

6809 Divides and Conquers	OCT,5
Assembly Language Graphics	MAR,6
Color Computer Assembly Language Programming	FEB,7
Corrections to June Barden Article	OCT,9
Machine Language Sort	MAY,7
Machine Language Sort, Part II	JUN,13
Moving Dots	APR,8
Sieve of Eratosthenes	JUL/AUG,8



Color Computer

Color Computer EDTASM + Tips	DEC,29
Bugs, Errors, and Fixes	
Color Disk Drive Kit (26-3022)	DEC,6
Color Scripsit (26-3105)	JUL/AUG,46
Color Spectaculator (26-3104)	JUL/AUG,46
Product Line Manager's Page	
Answers to Interesting Questions	MAR,45
BASICally Speaking	FEB,37
Color Computer Notes by R. G. Neill	MAY,20
Extended Color BASIC Graphics and Color BASIC Sorts	MAY,46
More Serious Side of the CC	JAN,42
Questions and Answers	APR,41
Programs	
3-D Color Graphics by M. Granger	FEB,39
Asteroids by R. Zepp	JUL/AUG,46
Auto Repeat, Revisited by D. Gangwisch	JAN,43
Biorhythms by K. A. Mowen	FEB,44
Bouncing Box by W. T. Aprmadoc	FEB,44
Christmas Card by F. Pigounias	DEC,43
Christmas Eve by T. Day	DEC,39
Christmas Tree Letterhead by A. Voisard	NOV,37
Color Editors	JUL/AUG,43
Color Scripsit and the LP VIII by G. McGowan	JUL/AUG,45
Concentric Circles on the CoCo by D. Whitelock	DEC,43
Core Editor by J. Jamieson	JAN,44
Deltasin by D. Sligar	MAR,10
Draw for 4K by D. Shapiro	JUN,47
Firewood Calculator by G. Davis	OCT,24
Fireworks by F. Pigounias	JUL/AUG,47
Graphic Height Correction by R. A. Esquivel	JUN,38
Hex to Decimal by C. Thonen	OCT,10
Hi-Res Character Generator by R. Van Dyke	APR,46
High Resolution Graphics Character Generation by G. Fraser	JUN,41
High Resolution Graphics with 4K by M. Ayen	MAR,46
Joystick Draw Routine by J. Ebbert	FEB,38
Landline by R. Hallam	OCT,10
Linefeed with a Carriage Return by E. Hamilton	APR,45
Loadcopy and Loaddata by G. Fraser	OCT,44
Mailing Labels on the CoCo by D. Banaszak	OCT,46
Merge Cassette Programs by J. Mir	APR,42
Navigation with a Color Computer by L. McNeil	NOV,31

Perpetual Calendar by K. Heberle	APR,43
Planetarium by J. W. Hurley	SEP,45
Print to the Screen and Wraparound	OCT,10
Print Videotex Material	FEB,33
Printer Calendar by H. Stern	APR,43
Rotational Grafx by J. Ennis	JUN,45
Shoot-Em Again by J. W. Myers	JAN,43
Space Alert by J. W. Myers	FEB,41
String Packing	FEB,14
Supertext by M. McGovern	JUL/AUG,43
Text Editor by B. Lieallen	JUL/AUG,43
Text Editor Revisited by A. Basargekar	DEC,41
Texture by G. Fraser	JUN,44
Revisions, Enhancements, Etc.	JAN,43;APR,42
Computer Clubs	FEB,14;APR,2; JUL/AUG,15;SEP,38;OCT,17;DEC,40
Computer Customer Service	
Computers in Education	SEP,8
Data Base Management Systems	OCT,11
Graphic Basics	JUN,18
Holiday Fun with Your TRS-80	DEC,16
More about Scripsit 2.0	JUL/AUG,12
Notes From Computer Customer Services	APR,14
Phone Number Changes, Part Two	MAY,14
Questions We Ask You	FEB,16
Radio Shack Languages	MAY,12
Relocating Machine Language Programs	MAR,11
SCRIPSIT™ Ideas	APR,13
What is an Operating System?	NOV,5
Data Bases	
CompuServe	
Confessions of a Computing Family	FEB,12
Languages and More	MAY,17
Megawars	NOV,7
Multiple Choice Puts You to the Test	SEP,13
Network Expansion	APR,15
New E-Mail Service	JUN,20
New Menu System	JUL/AUG,14
New Services	JAN,9
Notes Off the CompuServe Wire	FEB,14
Racing Information Service	MAR,15
Shop at Home Now Available on CompuServe™	MAR,14
TRS-80 Christmas Story—	
Billy and Santa	DEC,7
Welcome to Someday	OCT,13
Dow Jones	
Academic American Encyclopedia—	
A to Z Electronically	OCT,18
Disclosure II	MAY,19
Dow Jones Sports Data Base	APR,17
Free Text Search	JAN,11
More Quotes and a Whole World of News	NOV,9
Movie Reviews	JUL/AUG,16
Weather Information on Dow Jones	MAR,16
Window on Wall Street	SEP,14
You Can Quote Us	JUN,22
Profile	
Merging Profile Files with Scripsit	JUL/AUG,19
Profile Plus	MAY,23
Profile III +: Introducing the Flexible Data Base Management System	OCT,25
Profile III +: Optimizing Storage Capacity	NOV,11
Profile III +: Setting Up a User Menu	DEC,18
Transferring Data From Long Records Into Shorter Records	SEP,43
Variable Length Segments	JAN,15
Standard and Poor's	
Now You Too Can Become a "Wall Street Wizard"	DEC,9
VisiCalc®, SATN	
Additional Use of "@LOOKUP" in VisiCalc by L. Scanlon	JUL/AUG,23

Decision Making	JAN,17
Divying Up	MAY,25
Introducing SATN, The Journal for VisiCalc® Users	NOV,14
New Commands and Functions for VisiCalc®	MAR,18
Personal Time Savers	SEP,19
Personal Time Savers Revisited	DEC,35

Education

Chemistry Simulations for Secondary Students	DEC,33
Color LOGO Now Available for the TRS-80 Color Computer	NOV,43
Compress Graphics Using Model III DEBUG by D. Tanner	JUL/AUG,34
Computer Literacy	JUN,30
Computers in the Bunkhouse	JAN,34
Education Graphics	JUN,29
Educational Courseware for Graph Analysis in High School Physics	APR,29
Essential Math Programs	FEB,29
Figure It Out With Alice and Mac	SEP,12
In This System, The Computer Future Is Now	JUN,31
Interpreting Graphs in Physics	APR,31
Introducing "Number Theory"	OCT,32
Introducing Radio Shack's C.A.R.D. 1: Sentences	MAY,33
Microcomputer-Based Data Management for Small Libraries	OCT,32
Microcomputers for Neophytes	JUL/AUG,15
microResearch Problems	JUL/AUG,17; SEP,15;OCT,27;NOV,10
Neighborhood Computer Literacy	JAN,35
Network 3—New Capabilities for the Classroom	JUL/AUG,32
News From the Education Division	SEP,30
Racing Information	JUL/AUG,15
reading•is•fun	JAN,34
Secondary Problem-Solving Programs	MAR,29
SIGI® and the TRS-80 Model II	JAN,33
Some Straight Talk About the Radio Shack Network Systems	SEP,29
Southern New Jersey Computer Awareness Project	APR,32
TRS-80 Author System	MAY,31
TRS-80® Educational Software Sourcebook	JAN,35
Using Your Printer to Make Math Worksheets for Grades K-8	NOV,41

Feature Story

Football for Summer	JUN,10
John Hopkins Search	MAY,5
Non-Printer Peripherals	APR,5
Outside Software Support by B. Elliott	DEC,5
Radio Shack Network in the Classroom	SEP,5
Random Access Files	FEB,5
Sequential Files	JAN,5
Word Processing with the Keiths	JUL/AUG,5
Compound Interest and Savings by M. Riggs	APR,20
Computer Camp for Diabetic Children	MAR,22
Figuring an I.R.A. Account by E. Hoover	SEP,7
Heart Health Programming Contest	OCT,24
Microcomputers in Education	DEC,32
National Crisis Center for the Deaf by S. Thompson	JAN,10
New Journal Calls For Papers	SEP,44
Ode to Micro Idiom by J. Carr	OCT,43
Radio Shack Agricultural Sourcebook	SEP,44
Reprints Are Now Available	DEC,34
Scripsit Ideas by W. Ashworth	OCT,21
Search and Rescue and TRS-80 by J. Wellman	SEP,7
Ultra Precision Multiplication by G. Richard	MAY,16
Verifying Programs and Data Files by the Detroit Computer Support Group	May,20
Wind-Chill Factor by W. C. Clough, Jr.	APR,7

Fort Worth Scene

	JAN,14;FEB,11;MAR,2;
	APR,2;MAY,2;JUN,2;JUL/AUG,2;
	SEP,2;OCT,2;NOV,2;DEC,2

General Interest

&, &H and VAL by B. Dickson	FEB,24
Binding Fan Fold Paper by E. Kooi	NOV,10

"Clear"	MAF
Coded Message by D. Snyder	MAF
Coded Message by P. Vogel	OC

International News

Computing in the Netherlands	
Radiation Suppression in a Model I by J. Noordanus	JAN
Computer Programs Sent by Shortwave	JAN

Level I

Level I Calendar by D. Goldman	MAF
Using Data Tapes by H. Morrisette	MAI



Model I/III

Baker's Dozen of Helpful Hints by J. Rogers	OC
Business Graphics	DE
LDOS Hints and Tips	DE
Model I Double Density by B. Elliott	OC
Model III Port Assignments	DE
Bugs, Errors, and Fixes	
Accounts Payable (26-1554)	AP
	MAY,29;JUN,33;JUL/AUG,30;S
Accounts Receivable (26-1555)	JA
	FEB,22;MAR,25;APR,21;M/
	JUN,33;JUL/AUG,30;S
Advanced Statistical Analysis (26-1705)	JUL/AU
Business Mailing List (26-1558)	M
Cassette Portfolio (26-1506)	JAN,20;JUL/AUG,29;S
Cobol Compiler (26-2203)	FE
	MAY,30;JUL/AUG,31;SEP,28;O
Disk Payroll (26-1556)	FEB,22;MA
	JUL/AUG,30;S
General Ledger (26-1552)	JAN,20;M/
Inventory Control (26-1553)	JL
Manufacturing Inventory Control (26-1559)	APR,21;JUL/AU
Medical Office System (26-1568)	AF
	MAY,30;JUL/AUG,30;NOV,26;S
Micro Movie (26-1903)	M
Microfiles (26-1565)	JAN,20;M
Model III Disk Course (26-2014)	MAR,27;N
Model III Disk System Owner's Manual (26-2111)	MAR,27;C
Model III Operation and BASIC Language Reference Manual (26-2112)	M
Model III TRSDOS (26-312)	FEB,21;APR,21;JL
Stockpak (26-1507)	JUL/AUG,30;S
Profile (26-1562)	F
Real Estate (26-1571)	M
Real Estate Vol. III (26-1573)	MAY,30;S
Versaflex (26-1604)	J
Product Line Manager's Page	
Desktop/Plan-80	M
More PRINT USING with articles by J. N. Phillips, O. G. Leslie, E. R. Kool, J. Bond	F
Personnel Manager	A
SuperSCRIPSIT™—An Overview	M
SuperSCRIPSIT and SCRIPSIT 2.0—A Comparison	JUL/A
Three Dimensional Graphics Package by C. E. Burton	J
Time Manager, Revisited	J

Products

Business Checkwriter (26-1585) NOV,15
 LDOS—The New Operating Systems for
 Your TRS-80 by Logical Systems, Inc NOV,23
 Note to SuperSCRIPSIT Users with the DMP-200 NOV,20

Programs

Arithmetic Practice by P. Fried SEP,24
 Array Input Routine by L. Self APR,24
 Arrays by L. Miller APR,23
 Blinking Prompt and INKEY\$ by S. Vescera MAR,17
 Budget Management and the LP VIII
 by R. Shackelford NOV,6
 Cassette Telephone File by S. Walker JAN,28
 Client Activity Information Array
 by K. E. Robinson APR,25
 Concentric Circles by S. Havens JUN,22
 Crush—A Mathematical Game by T. Myerson SEP,32
 Customary to Metric by T. Pruzan SEP,19
 Directory Algorithm by T. Kelanic SEP,20
 Double Dump to Tape by R. Pollock MAR,5
 Educational Programs SEP,19
 Extending Sequential Files by J. Whinery JAN,29
 FORTRAN/BASIC Data Files
 by J. L. Montgomery FEB,24
 Graphic Keys by R. Bennett JUN,17
 Guess the Winning Number by R. Huttenbrauck MAR,5
 Instant Recall by D. Dager FEB,25
 MERGEIT/BAS by D.B. Hatch JAN,22
 Microfile Hints by W. Smith FEB,15
 Model I Head Cleaner Program,
 Version 2 by L. Johnson APR,16
 Model I/III Head Cleaner by R. Brown JUN,17
 Model I/III Disk Head Cleaner Program
 by M. Sooke JAN,14
 Model III Disk Inventory—Version Two
 by D. Henao JAN,30
 Modification to Sorted Directory Program
 by S. Bailey JAN,29
 Pascal Division Program by J. Wood NOV,26
 Physical Science Program by A. Freeman SEP,19
 Poor Man's Editor by D. Powell JUL/AUG,35
 Prime Number Generator #1 by G. Greer SEP,23
 Prime Number Generator #2 by R. Euliano SEP,23
 PRINT to LPRINT Improved A. Reudisueli
 and J. C. Miller FEB,23
 Quick Printer Graph Routine by G. Patek NOV,36
 RSBASIC Lister Program by W. J. Schauert JAN,21
 Single Disk FORTRAN by T. Daneliuk NOV,43
 Software Driver for Model I Voice Synthesizer
 by S. D. Griffis NOV,22
 Special Character Mode by R. Butts JUN,33
 String Formatting and the Window Game
 by D. Fuger SEP,20
 Stunt Racer by Y. Maksik FEB,27
 Tape Mailing List and Two Across Labels
 by R. Blinn NOV,10
 Revisions, Enhancements, Etc JAN,21;APR,22

Model II

Model II Scripsit 2.0, L P VIII and the New Printers
 Bugs, Errors, and Fixes NOV,30
 Accounts Payable (26-4505) MAR,34;
 APR,36;MAY,36;JUL/AUG,38;SEP,37
 Accounts Receivable (26-4554) JUN,37;SEP,38
 Accounts Receivable (26-4604) JAN,39
 BEDIT (26-4705) with TRSDOS 4.1 DEC,32
 BISYNC 3270 (26-4715) JAN,40;FEB,35;
 MAR,34;MAY,37;SEP,38;DEC,32
 BISYNC 3780 (26-4716) FEB,35;MAR,34;MAY,37
 Business Mailing List (26-4506) MAY,36
 General Ledger (26-4501) JUN,36;SEP,37
 Hard Disk Interfacing Applications MAY,36
 Inventory Management (26-4502) JAN,39;
 JUN,36;JUL/AUG,38
 Mailing List (26-4507) SEP,37
 Payroll (26-4503) JUN,36;SEP,37;DEC,32

Profile II (26-4512) JAN,40;FEB,35;DEC,32
 RSBASIC Compiler (26-4705) MAR,34;MAY,37
 RSCOBOL (26-4703) APR,36;MAY,37
 RSCOBOL Generator (26-4707) SEP,38
 RUNCOBOL (26-4704) APR,36
 SCRIPSIT 1.0 (26-4530) FEB,35
 SCRIPSIT 2.0 (26-4531) FEB,35;MAY,37;
 JUN,37;JUL/AUG,38;OCT,37
 Statistical Analysis (26-4540) JUL/AUG,39;SEP,38
 TRSDOS (26-4910) JAN,40;MAR,35;
 JUN,37;JUL/AUG,39
 TRSDOS-HD (26-4150) JUN,38;
 JUL/AUG,38;DEC,31
 Versafile (26-4510) JAN,39
 Product Line Manager's Page
 High Resolution Business Graphics for
 Models II and 16 JUN,34
 Model II Hard Disk System FEB,34
 Model II SCRIPSIT 1.0 vs. Model II SCRIPSIT 2.0 APR,33
 New TRS-80 Model 16 MAR,32
 Power Up Sequence for the Model II MAY,35
 Scripsit Ideas JUL/AUG,36



Programs

ADDSPACE/BAS by H. C. Brown MAY,40
 Auto Field Routine for Model II by M. Lupul SEP,33
 Calculator by A. Banick, Jr SEP,34
 Chaining Subroutine for FORTRAN
 by J. Montgomery MAY,38
 Changing Print Size on the Line
 Printers V and VI by R. Graham NOV,28
 Clean Text by A. Spencer JAN,40
 Firmament by W. Boyd OCT,38
 Garbage Collection by W. L. Pierce FEB,36
 General Electric ASCII Sequential Files
 by R. Halloran SEP,35
 Histogram by K. Willoughby JUN,40
 Model II BACKIT by C. Rizzio MAR,35
 Model II Scripsit 1.0 Underline by A. McCoy JAN,8
 Model II Twinkling Christmas Tree
 by R. P. Graham DEC,31
 More Scripsit Ideas by C. M. Mendenhall JUL/AUG,37
 PATCHWRT Writing Program by R. Burke OCT,35
 Programming Techniques SEP,33
 Read Screen by A. Pacheco SEP,34
 RSCOBOL Screen Dump Utility by D. Ray JAN,39
 Statistical Analysis by W. W. Schumacher APR,35
 String Machine Language Programs
 by C. Hamkins SEP,33
 Typewriter by D. F. Salisbury FEB,36
 Revisions, Enhancements, Etc JAN,40;APR,36

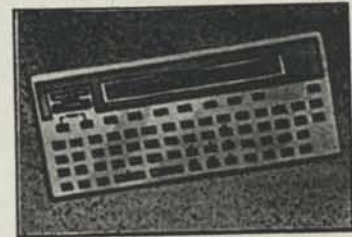
Notes on Previous Newsletters

June, 1981	
PCLEAR	JUN,38
SAFE	OCT,4
August, 1981	
Checkerboard	MAR,27
November, 1981	
Color Disassembler	MAR,27
Model III Directory Menu	DEC,25
December, 1981	
Model I/III Calendar Program	
Corrections	APR,28
PATCH and Spelling Checker	OCT,4
January, 1982	
Cassette Telephone File	NOV,21
Lines	OCT,4
One-Armed Dime Bandit	OCT,4
Shoot 'Em Again	OCT,4
VisiCalc	APR,28
February, 1982	
BASICally Speaking	DEC,25
Bouncing Box	JUN,39
Mailing List Files	NOV,21
No Gaps in Names	JUN,39
Random Files	NOV,21
Typewriter	JUN,40;NOV,21
March, 1982	
Blinking Prompt and INKEY\$	NOV,21
Clear	JUL/AUG,22
Deltasin	JUL/AUG,22
Double Dump to Tape	JUL/AUG,22
Model II Bugs, Errors, & Fixes	JUL/AUG,22
Pocket Computer Programs	JUL/AUG,22
Relocating Machine Language Program	JUL/AUG,22;DEC,25
Sub Destroyer	JUN,40;JUL/AUG,22
William Barden Jr.'s Series	NOV,21
April, 1982	
Linefeed with a Carriage Return	DEC,25
May, 1982	
SuperSCRIPSIT—an Overview	NOV,22
June, 1982	
Concentric Circles	DEC,26
Disk Head Cleaner	NOV,22
Graphs for the PC-2	DEC,26
July/August, 1982	
Asteroids	DEC,26
Color SCRIPSIT Bug	DEC,27
Graphic Keys	DEC,27
More About SCRIPSIT	DEC,27
SCRIPSIT and the LP VIII	DEC,26
Sieve of Eratosthenes	DEC,27

Peripherals

Color Graphic Printer	OCT,22
Have We Got Printers for You! by L. Miller	NOV,16
Modem II Programming Techniques	DEC,37
Modem II Questions and Answers by P. Schreiber	SEP,16
Bugs, Errors, and Fixes	
Line Printer VII (26-1167)	APR,7
Product Line Manager's Page	
Direct-Connect Modem II	FEB,19
Envelope Feeder for Daisy Wheel II	MAY,21
Graphics Subroutines	JUN,23
Plug 'n Power™ Program Changes	MAY,22
Printer Codes Revisited	MAR,20
TRS-80 Disk Head Cleaning Kits	JAN,13
Using the Digitizer to Calculate Area	
Measurement and Curve Integration	JUL/AUG,18
Programs	
Data File Recording Program by A. Barclay	OCT,39
Descenders on the LP VII by J. Diel	NOV,29
Line Printer Set Up by W. Allen	NOV,27
Line Printer VI Control Program	
by H. R. Priesmeyer	FEB,33

LP VI Underline/Boldface by D. Wood	FEB
More TAB and USING in a Total Inventory	
Program by R. Perkins	NOV
Printing Titles by B. Christ	NOV
Quick Label by R. W. MacLean	FEB
Revisions, Enhancements, Etc.	AP



Pocket Computer

Availability of 16K RAM Modules for the PC-2	DE
Pocket Computer II Start-Up Instructions	SE
Word Processing on the Pocket Computer	
by D. Hergert	NC
Bugs, Errors, and Fixes	
Calendar (26-3529)	OC
Electrical Engineering (26-3520)	MA
Personal Finance (26-3518)	JUN,9;JUL/AU
Surveying (26-3512)	JUN,9;JUL/AU
Product Line Manager's Page	
Pocket Computer Model PC-2	M
Printing Features of the PC-2	J
Retirement Income Projection	JA
Programs	
Circle Subroutine for PC-2	NI
Clock/Scheduler for the Busy Executive by J. Titus	DI
Compound Miters by B. Taub	AI
Convert Integer to Binary by M. E. Wilborne	M
Convert 35MM to 16MM by R. Edwards	JUL/AU
Cost Comparison by V. Scally	JUL/AU
Cub Scout Song by J. L. Schafer	AI
Data File Recording Program by A. Barclay	C
Factorials by T. Cox	JUL/AU
Homework by B. Cutler	S
Measurement Conversion by F. Bishop	JUL/AU
Mileage Calculations by A. Leiwant	JUL/AU
MPG on the PC by B. Taub	JUL/AU
Multiple Linear Regression by G. Bryant	A
One Armed Dime Bandit by D. W. Bond	J
PC-1 Twelve Days of Christmas by R. K. Phelps	D
Pearson (R) Correlation by G. Currah	C
Pocket Computer Programs by F. Nachbaur	M
Printer Plot by H. G. Dorsey III	F
Projectile Trajectory by L. Mermelstein	JUL/AU
Pythagoras' Theorem by J. Strolin	F
Resistor Values by R. Edwards	JUL/AU
Reverse Polish Notation on the Pocket	
Computer by J. Morrison	C
Star Trek Fight by R. Saccone	I
Step Saving Ideas by L. Levine	F
Use of the TRS-80 Pocket Computer in	
Celestial Navigation by D. Parson	JUL/AU
You Want to Calculate What?	JUL/AU
Revisions, Enhancements, Etc.	JAN,46;

View from the 7th Floor

by Jon Shirley	JAN,4;FEB,4;
	APR,4;MAY,4;JUN,4;JUL/AUG,4;SEP,4;NOV,4

Notes Previous

NOVEMBER 1981

Model III Directory Menu

Richard Ellers
426 Central Parkway SE
Warren, OH 44483

Here is what may be the world's shortest, practical BASIC program:

```
1 NEW
SAVE "NEW"
```

I devised it to use with James Ditucci's terrific auto loading Model III directory menu from the November 1981 TRS-80 Microcomputer News.

Ditucci devised his auto-loaded directory so a user would not have to know about and go through booting.

"NEW" adds convenience for the programmer by providing direct entry into program-writing BASIC mode without having to suppress the auto load on BOOT or having to load a program from the menu, breaking it, and starting NEW.

In the same fashion, another short program gets you to TRSDOS directly from Ditucci's menu.

```
1 CMD"S"
SAVE "CMD/S"
```

FEBRUARY 1982

BASICally Speaking

Bruce Lewis
Box 395
Osceola, MO 64776

In the February 1982 issue on the product line manager's page, there was a program that used RND(TIMER) to scramble the Color Computer's random sequence. This doesn't seem to work on my Extended BASIC version. The assembly language listing below is the ultimate way to scramble the sequence. It uses the timer to scramble the memory holding the position of the random sequence.

To use the code below, you may have to refer to the article on assembly language programming in the February 1982 issue of Microcomputer News.

This may be used as a subroutine in your BASIC program. Pressing <BREAK> while this subroutine is working will not stop your BASIC program. The computer will strobe the keyboard again to see if <BREAK> is pressed, and since it doesn't use auto repeat, it will recognize no keys.

I hope to see other assembly language programs in future issues of this magazine. It should save other programmers from insomnia.

```

0FE2          00100      ORG   SFE2
0FE2 FC 0FE2   00110  START  LDD   SFE2  GET SEED FROM TIMER
0FE5 F3 0116   00120      ADD D $116  \
0FE8 FD 0116   00130      STD   $116  I
0FEB F3 0117   00140      ADD D $117  MIX WITH RANDOM
0FEE FD 0117   00150      STD   $117  SEQUENCE MEMORY
0FF1 F3 0118   00160      ADD D $118  I
0FF4 FD 0118   00170      STD   $118  /
0FF7 AD 9F A000 00180      JSR   [9A000] POLL FOR KEY
0FFB 81 03      00190      CMP A #03  <BREAK> PRESSED?
0FFD 26 E3      00200      BNE  START  IF NO, CONTINUE
0FFF 39        00210      RTS     RETURN TO BASIC
          00220      END
00000 TOTAL ERRORS
END 0FFF
START 0FE2
```

MARCH 1982

Relocating Machine Language Programs

David D. Bailey
3962 Petite Dr. W.
Jacksonville Beach, FL 32250

In regards to the March article about Relocating Machine Language Programs, there is another problem that can occur with any program that requires information to be stored from the program. Although you point out that the "SAVE" command will still store data on cassette tape, you failed to tell the Model III user that they must make one more change to allow this to occur.

The original programs, such as "Cassette Scripsit", were designed to operate at Low Baud. When they were transferred to disk, the computer operated in High Baud. This will not allow programs already stored to be LOADED, nor will the program "SAVE" or "LOAD."

This can be corrected using the following sequence.

(1) Patch to prompt for the Baud Rate:

```
PATCH BASIC/CMD (ADD=5202, FIND=00, CHG=FF) <ENTER>
```

You must follow the sequence below each time you re-boot.

- (2) a. Start TRSDOS
- b. Answer Date Prompt
- c. Answer Time Prompt
- d. BASIC <ENTER> ' Go to Disk BASIC
- e. CASS? L
- f. <ENTER>
- g. <ENTER>
- h. CMD"S" ' Returns to TRSDOS
- i. SCRIPS/CMD ' Loads Program

I realize this is an involved start-up sequence, but by using this method you can "SAVE" your programs to tape.

APRIL 1982

Linefeed with a Carriage Return

David E. Pitts
16011 Stonehaven Drive
Houston, TX 77059

I am currently using a Dot Matrix printer with my 32K Color Computer. Since this printer has had the automatic linefeed circuit disabled, I was very interested in Ed Hamilton's article "Linefeed with a Carriage Return." Since I write many of my BASIC programs with maximum packing (up to 255 characters per line), I needed Hamilton's routine to give a carriage return and linefeed after each N characters have been printed. The program is currently set up for N=78 characters (change line 165 to have other line widths). This not only allowed me to print my long lines with abandon, but also to print narrow columns if I should so desire. Below is the BASIC program. My compliments to Mr. Hamilton for an excellent program.

```

10 REM O/S PATCH FOR LF ON CR
20 REM MODIFIED BY DAVID PITTS FROM ED HAMILTON
   TRS-80 MICROCOMPUTER NEWS APRIL 1982 PAGE 45
30 REM MODIFIED TO GIVE LF AND CR AFTER 78
   CHARACTERS
100 REM PUT CODE AT MEMTOP -41 (CODE IS AT 32726,
   32725=WORKING)
110 MT=PEEK(116) * 256 + PEEK(117)
112 MT=MT-40
120 FOR I = 0 TO 39
122 READ T
124 POKE MT + I, T
126 NEXT
```

```

130 REM SAVE O/S CODE
140 REM PUT JMP ON O/S HANDLER
142 AH = INT(MT/256)
144 AL = MT - AH * 256
146 POKE 359, 126
148 POKE 360, AH
    : POKE 361, AL.
150 REM MACHINE - LANGUAGE CODE
151 DATA 52, 4, 214, 111, 193, 254
155 DATA 38, 27, 124, 127, 213, 129
160 DATA 13, 39, 10, 246, 127, 213
165 DATA 193, 78, 38, 13, 189, 162
166 REM CHANGE 78 IN ABOVE LINE TO BE LINE WIDTH
    DESIRED.
170 DATA 133, 127, 127, 213, 134, 10
175 DATA 189, 162, 133, 134, 13, 53, 4
180 DATA 126, 203, 74
200 REM RESERVE MEMORY
210 CLEAR 200, MT-1
220 CLS
230 NEW
999 END

```

JUNE 1982

Concentric Circles

Donan C. McGinley
Keystone Engineering Corp.
845 N. 19th Street
Philadelphia, PA 19130

I just wanted to make a few additions to Stephen Haven's Concentric Circles program. To begin with, an LPRINT inserted at line 85 gives the program a little animation. All sorts of variations are possible. Making the value of the resolution greater slows the animation while making it smaller speeds it up. The value for the number of spaces determines how quickly the spaces on the screen vary. Changing the constant in line 70 (2.6667) to a smaller value also causes interesting changes.

The program is even better when an extra line or circle is plotted. Below are two additions that can be made:

```

71 G = 63 + Y * (2.6667 * SIN(A)): H = 23 - Y *
    COS(A)
81 SET (G,H)
85 LPRINT
OR
81 SET (N,N)
85 LPRINT

```

By experimenting with various sizes, resolutions, and spaces, all sorts of amusing things appear. The possibilities are endless.

Andrew Pensky
105-25 67 Ave.
Forrest Hills, NY 11373

On page 32 of the June 1982 edition, you printed a program called "Concentric Circles." While running the program through the computer, I noticed two mistakes in line 90.

Line 90 read:

```

90 NEXT A
  : Y = Y - 1 - Z
  : IF Y < 0 THEN 160 ELSE 90

```

Line 90 should read:

```

90 NEXT A
  : Y = Y - 1 - Z
  : IF Y < 0 THEN 120 ELSE 60

```

The first mistake was that there was no line 160 for the computer to go to. And the ELSE function in that line sends you back to a NEXT A statement which causes a NEXT without FOR error. To see the program work as stated, input Y=23:X=23:Z=6 when prompted by the program.

Graphs for the PC-2

The following are known errors in the Graphs for the PC-2 program which appeared in the June 1982 issue.

Line 944 should be entered as line 955. There is no line 944.

In line 1045 the semicolon (;) preceding the final USIN statement in the line should be changed to a colon (:).

In line 1115 where the statement ... LET L=1 ... appears, the letter l should be changed to the number 1.

Our thanks to Mack R. Rhea of Nashville, Tennessee and Robert C. Taylor of Montreal for these changes. Mr. Taylor sent the the following changes.

Line 1225 becomes:

```
IF L>3 LET L=1
```

Line 1226 becomes:

```
LINE (V,P) - (S,P),0,L
```

Line 1260 becomes:

```
IF L>3 LET L=1
```

Line 1261 becomes:

```
LINE (P,W) - (P,T),0,L
```

JULY/AUGUST 1982

Asteroids

Joseph Prisco
RD #7 Box 80
Oswego, NY 13126

I have just reached a score of 3998 in Richard Zer's "Asteroids" with screen number 118. This tops the ship high score to date. Also, I changed the color of the "Ship" is easier to see in black and white and added two lines to make the high score tally along with the player's score (the high). They are as follows:

```

295 IF S > HS THEN HS = S
311 IF S > HS THEN HS = S

```

SCRIPSIT and the LP VIII

H.U. Kohler, CDP
P.O. Box 628
Proctor, VT 05765

The July/August 1982 issue addresses SCRIPSIT and the LP VIII by Gilbert Keith. To add to the list of possibilities, my SCRIPSIT disk, when RESET, goes automatically to the BASIC program "PRINTER" which displays the appropriate menu. After the desired typestyle is entered the program automatically goes back to TRSDOS with the CMD"S" statement.

Needless to say, the printer should always be reset to running "PRINTER."

```

2 *****
3 *
4 * PROGRAM TO SET UP PRINTER TYPE AND SPACING
5 *
6 * BY: H.U. KOHLER, CDP
7 * P.O. BOX 628
8 * PROCTOR, VT 05765
9 *
10 *****
11 CLS
    : PRINT@202,"-----"
    : PRINT@202,"MENU"
    : PRINT@202,"-----"
12 PRINT@266,"1 - PRINT EXPANDED REGULAR
    CHARACTERS"
13 PRINT@330,"2 - PRINT PROPORTIONATE CHARACTERS"
14 PRINT@394,"3 - PRINT PROPORTIONATE EXPANDED
    CHARACTERS"
15 PRINT@458,"4 - PRINT CONDENSED CHARACTERS, 6
    LINES PER INCH
16 PRINT@522,"5 - PRINT CONDENSED EXPANDED
    CHARACTERS, 6 LPI"

```

```

17 PRINT#586,"6 - PRINT CONDENSED CHARACTERS, 3/4
    VERTICAL SPACING"
18 PRINT#714,"ENTER FUNCTION CODE:";
19 INPUT I
20 ON I GOTO 22, 24, 26, 28, 30, 32
21 GOTO 12
22 LPRINT CHR$(27);CHR$(14)
23 CMD"S"
24 LPRINT CHR$(27);CHR$(17)
25 CMD"S"
26 LPRINT CHR$(27);CHR$(14);CHR$(27);CHR$(17)
27 CMD"S"
28 LPRINT CHR$(27);CHR$(20)
29 CMD"S"
30 LPRINT CHR$(27);CHR$(20);CHR$(27);CHR$(14)
31 CMD"S"
32 LPRINT CHR$(27);CHR$(20);CHR$(27);CHR$(56)
33 CMD"S"

```

More about SCRIPSIT

Carol V. English, Ph.D.
9920 West 34th Drive
Wheat Ridge, CO 80033

Since our office acquired the Model II, my secretary complains that I leave her only the routine things to type up! I'm enjoying so much of the software myself.

In the July/August issue on page 13 of "More about SCRIPSIT," there is a suggestion on how to address an envelope. I think my method is a bit less cumbersome. Possibly, it just seems that way to me because I created the method.

After I type the letter, I define the addressee information from the letter and duplicate it on the page I create for this purpose which follows the last page of the letter. (CTRL) (R) sets up the address. After I print the letter (set up on the print menu to stop after the last page of the letter), I go back to the print menu for the document and program it to print only the address page. Since the company stationery envelopes have our return address on them, I need only to program the print menu for the column position that fits the size of the regular envelope. (In our case, 40)

For a large manila envelope, I execute a user key defined for my company's return address, press (ENTER) eighteen times, change the format line to conform to the larger envelope (TABS here would require a line-by-line entry or rearrangement for the block form), and do a (CTRL) (R) to bring up the duplicated address from the letter.

I hope Model II users find this as easy a method as I do.

Andrew J. Wheeler, Ph.D.
Department of Human Resources
State of North Carolina
Murdoch Center
Butner, NC 27509

I enjoyed your July/August issue featuring word processing and thought that you might be interested in our adaptation of your method for addressing envelopes described on page 13. Our office has a 48K, two disk Model III with Daisy Wheel II Printer and Scripsit. We have put the letter and envelope on separate pages of a single document. Since Scripsit does not have user-defined keys, the addressee block at the beginning of the letter is defined as Block A. Then on the second page (the envelope) of the document, the return address is typed to appear on the upper left corner of the envelope, the margins are reset for the address (LM = 40 works for a standard business envelope), and a number of line feeds are entered for centering the address. At the place where the address should go, a comment line is entered, "Insert Block A Here." When typing form letters, simply type in the name and address at the beginning

of the letter, and then move the cursor down to the beginning of the comment line. Insert Block A, then insert stationery into the printer and give the print-with-pause-between-pages command. Insert envelope for second page, and you get the letter and envelope after having typed name and address only once.

I hope this adaptation will be helpful for Model III and Scripsit users.

Sieve of Eratosthenes

Joel M. Jacobson
5 Sudbrook Court
Pikesville, MD 21208

The article by Wm. Barden, Jr., "Sieve of Eratosthenes" was especially interesting to me since I have been using the simple BASIC program below to determine prime numbers. While this should work between any limits, the time becomes excessive for very large numbers.

```

10 PRINT "ENTER RANGE IN WHICH PRIME NUMBERS ARE
    DESIRED."
20 INPUT "LOWER LIMIT, UPPER LIMIT"; L,U
30 N = L
40 FOR M = 2 TO N - 1
50 X = N/M
60 IF X-INT(X)=0 THEN 100
70 NEXT M
80 PRINT N;
100 N = N + 1: IF N > U THEN END
110 GOTO 40

```

Color Scripsit Bug

David Bressan
2815 Ritchie Rd.
Forestville, MD 20747

I read your July/August issue and found others have encountered the bug in Color Scripsit concerning TAB ((SHIFT) (RIGHT ARROW)) when in the Insert Mode. To avoid this, set a tab stop at the right end of the line when you start up or load from tape. You will then only insert spaces to the line end and not to memory end.

Graphic Keys

Robby Bennett
2702 S. Banker
Effingham, IL 62401

On page 17 of the newsletter the program Graphic Keys contains an error. Change 7 LPRINT CHR\$(21); to 7 PRINT CHR\$(21);



A Clock/Scheduler for the Busy Executive

John Titus
P. O. Box 153
Descanso, CA 92016

Our hypothetical Mr. X is a busy executive with many demands on his time. Unfortunately, he also has many interruptions and is somewhat absentminded, so he needs frequent reminders of what he has to do next. For this he doesn't particularly want the time. What he wants is to know how long it will be until the next thing he has to do.

Mr. X also has a number of clients throughout the world (New York, London, Pago-Pago) and sometimes needs to call them.

Because of its flexibility and special features such as TIME, the PC-2 is an ideal way to handle these functions. (Remember how to set the TIME? According to your PC-2 manual, TIME can be set or reset by typing: TIME =MMDDHR.MMSS<ENTER>. That's two digits each for month, day, hour, a decimal point, and two digits each for minutes and seconds.)

The program operates in 4 modes:

1. F3:GOTO 75@

This entry clears the previous schedule and sets up a new one. The time for each appointment is entered as HR.MN(P), using Hours (2 digits), Minutes (2 digits), with a decimal point in between. For times after Noon, P must be added. (A for AM is optional.) EVENT is a short note which may be added as a reminder of why the time is being set aside. A printed record is produced.

Note: This entry must be entered first, since it sets up the DIM Statements for the variables. Here is a sample:

SCHEDULE

```
# 1 10.25 CALL J
# 2 12.10P LUNCH W
  JTH M
# 3 4.20P MEET WIT
  H A&C
# 4 4.40P HENRY G
```

2. F4:GOTO 80@

This entry adds items to the schedule without disturbing existing items.

3. F1:GOTO 10@

This item presents the local time and a graphic display (with 10-minute interval markers) plus a small cross for each item on the schedule occurring within the next 60 minutes.



In the above example, the time is 3:57P, and the items displayed are the two between 3:57 and 4:47P (the items at 4:20P and 4:40P).

We should add a detail here about the program's sorting capabilities. All events which are scheduled to occur within a given hour will be displayed. However, their order of display will depend on how they were originally input. The two displays which follow were input in chronological order (see display screens, #3 and #4). If their input had been switched, their order of appearance, within that given hour, would also have been switched.

Push <ENTER> after the BUSY signal goes out, and the time before the first item will be displayed.



Push <ENTER> again and the next one will be displayed.



4. F2:GOTO 200@

This entry gives times in various cities. City names and time differences are stored in DATA. The display will show:



The first two digits entered must correspond to the first two digits stored in the city's name in DATA. Thus for London time, these could be LO, LON, etc.



Push <ENTER> and the time in London is displayed:



And with all this, Mr. X still manages to miss an appointment now and then. But everybody admits he is a lot better than he used to be!

If no printer is available, the LP and LF instructions in Statements 75 and 95 should be removed.

```

10 REM DISPLAY SCHEDULE
15 GOSUB 100
   :D$=STR$ D
   :C$=STR$C
   :B$=""
   :IF D<10 LET D$=B$+D$
20 CLS
   :WAIT 0
   :FOR I=39 TO 147 STEP 18
25 GPCURSOR I=1
   :GPRINT 65
   :GCURSOR I
   :GPRINT 34
   :GCURSOR I+1
   :GPRINT 65
   :NEXT I
30 CURSOR 0
   :PRINT C$;" ";D$;A$
35 FOR I=1 TO 10
   :GOSUB 180
   :GOTO 35+X
40 V=108*U+40
   :GCURSOR V-1
   :GPRINT 8
   :GCURSOR V
   :GPRINT 28
   :GCURSOR V+1
   :GPRINT 8
45 NEXT I
50 WAIT
   :CURSOR 0
   :PRINT ""
55 FOR I=1 TO 10
   :GOSUB 180
   :GOTO 55+X
60 PRINT INT (U*60+1);" min til ";E$(I)
65 NEXT I
   :PRINT "THATS ALL FOR NOW!"
   :STOP
70 REM SET UP SCHEDULE
75 CLEAR
   :DIM H(10),E$(10)
   :K=1
   :LPRINT "SCHEDULE"
   :LF 1
80 INPUT "HR.MN(P) ";U$
   :H(K)=DEG VAL U$
85 IF RIGHT$(U$,1)="P"AND H(K)<12 LET
   H(K)=H(K)+12
90 INPUT "EVENT?";E$(K)
95 LPRINT "#";K;" ";U$;" ";E$(K)
   :K=K+1
   :GOTO 80
100 REM LOOK UP TIME
105 E=TIME /100
   :FOR I=2 TO 4
110 @(I)=INT E
   :E=(E-@(I))*100
   :NEXT I
115 H=C+D/60+E/3600
   :GOSUB 150
   :RETURN
150 REM H<24, C<13, AM/PM
155 IF H>24 LET H=H-24
160 A$="P"
   :IF H<12 LET A$="A"
165 IF C>=13 LET C=C-12
   :GOTO 165
170 RETURN

```

```

180 U=H(I)-H
   :X=5
   :IF U>1 OR U<0 LET X=10
185 RETURN
200 REM TIME IN OTHER CITIES
205 GOSUB 100
   :WAIT 0
   :PRINT "At "
210 CURSOR 12
   :PRINT "The Time is"
   :CURSOR 5
   :INPUT L$
220 RESTORE
   :FOR I+1 TO 6
   :READ J$,DL
   :IF LEFT$(L$,2)=LEFT$(J$,2) LET H=H+DL
   :C=C+DL
   :GOSUB 150
   :GOTO 230
225 NEXT I
230 CLS
   :WAIT
   :PRINT C;" hrs ";D;" min ";A$;"M ";J$
   :GOTO 200
300 DATA "LONDON",8,"BONN",9,"NY",3,"CHICAGO",
   2,"LA",0,"PAGO-PAGO",21
320 END

```

Color Computer EDTASM + Tips

George Yefchak
4112 Silver Oak St.
Dayton, OH 45424

I have been very happy with EDTASM+, the Color Computer Editor-Assembler (Stock Number 26-3250). I bought one of the first ones available in my area. There is an error in the manual on page 26. In the section on passing parameters to and from assembly routines, the method shown for calling the floating-point routines INTCNV and GIVABF is wrong. The starting addresses of these routines are not STORED IN \$B3ED and \$B4F4; they ARE \$B3ED and \$B4F4, respectively. In other words, these two routines should be called with direct addressing, not indirect addressing. Note that the other ROM routines listed in the manual (JOYIN, CHROUT, etc.) are to be called indirectly, as the manual indicates.

One more note. Although INTCNV must be called with either the JSR or LBSR instructions, as the manual says, GIVABF can be called with the simpler JMP or LBRA instructions. It will then return directly to BASIC. I suspect that JMP would be the most efficient.



LDOS Hints and Tips

The following information was excerpted from *THE LDOS QUARTERLY* Volume 1, Number 2, October 1, 1981 with permission from Logical Systems, Incorporated.

BACKUP-BY-CLASS

The Backup-by-class feature of LDOS is a very powerful feature, but it is also very complex internally. There are a couple of things that are done when SYS0 is moved to the destination disk by a Backup-by-class that are not readily apparent.

SYS0 is the heart of LDOS. It can also be referred to as the resident system. It is like a traffic control system for the entire operation of LDOS. When it is moved, certain other sectors of the source disk are also placed on the destination disk. These are sectors 2, 3, and 4 of track 0.

One reason these sectors are moved is that they contain special information storage used by the system, sectors 0 and 1 are not moved to assure that the proper type of boot will be on the destination disk. Another reason to move these (sectors 2 - 4) is that the state of the SYSTEM (SYSGEN), On or Off, is stored within them. So, if the source disk is sysgened then the destination will also be, and if the source disk is not sysgened, the destination disk will NOT be sysgened, even if it was before the backup began.

This is a good time to mention the use of JCL to configure your system. Because you have to type in all the setup you desire for your system, why not type it into a JCL file so that it can be reused!! To do this, refer to BUILD in your LDOS manual (a suggested name for the file would be SYSGEN/JCL). After you have entered the statement to turn on or set up the features and options you want, simply execute the JCL file with the LDOS DO command. This will execute all the statements you typed in as though you were entering them as commands from the keyboard. When the DO is completed, type SYSTEM (SYSGEN) and the present setup will be saved on drive 0 as that system disk's configuration. Do not place "SYSTEM (SYSGEN)" in the JCL file!

CREATING A MINIMUM CONFIGURATION DISK

All files except certain /SYS files may be removed from your run time drive 0 disk.

For operation, SYS files 1, 2, 3, 4, 8, and 10 should remain on drive 0. SYS 11 must be present only if any JCL files will be used. Both libraries (SYS 6 and SYS 7) may be removed if no library command will be used. SYS 5 and SYS 9 may be removed if the system DEBUGger is not used. SYS 0 may be removed from any disk not used for BOOTing.

When using LBASIC, BASIC/OVN and /OVX may be removed if no renumbering or cross-referencing will be done.

The presence of any Utility, Driver, or Filter program is totally dependent upon the user's individual needs. Most of the LDOS features can be saved in a configuration file with the SYSTEM (SYSGEN) command, so the Driver and Filter programs won't be needed in run time applications.

The passwords for LDOS files are as follows:

System files (/SYS)	Update Password = WOLVES
Filter files (/FLT)	Update Password = GSLTD
Driver files (/DVR)	Update Password = GSLTD
Utility files (/FLT)	Update Password = RRW3
LBASIC	Update Password = BASIC
BASIC overlays(/OV\$)	Update Password = BASIC
CONFIG/SYS	Update Password = CCC
MASTER PASSWORD	= RSOLTOFF

ITEMS OF GENERAL INTEREST

Model III owners wishing to move programs from LDOS to Model III TRSDOS should put the files on a 35 track, single density diskette and then use the Model III TRSDOS Convert command to move the files. Be sure the disk is formatted for 35 tracks, as the TRSDOS Convert expects the directory to be on track 17.

We have found that many direct connect modems need to have the DTR (Data Terminal Ready) line high before the modem will remain off hook. Therefore, when using one of the LDOS RS-232 driver programs, it may be necessary to specify (DTR) as a parameter when setting the RS-232 driver.

FEATURES OF LDOS 5.1

- Any LDOS command which requires parentheses around the parameters may be entered without the closing parenthesis.
- The ATTRIB command includes the functions of PROT.
- The COPY command contains an (X) parameter to allow the transferring of files when dealing with non-system diskettes. COPY also contains the ECHO parameter to echo characters to the screen during the copy.
- The CONV command will allow you to transfer files created on the Model III under TRSDOS 1.2 or 1.3 to an LDOS diskette. Note that Model I owners who wish to utilize CONV must have two drives and be capable of reading double density with their system.
- The Job Log (*JL) is established via the SET command.
- The Keyboard driver (*KI) activates the use of the <CLEAR> key and must be established if advanced keyboard features are to be utilized (KSM, MiniDOS, etc.). In addition, the type-ahead and screen print functions are enabled using the Keyboard driver.
- A SuperVisory Call (SVC) table can be loaded into high memory for use by assembly language programmers. It contains most documented system entry points and routines.
- The MiniDOS filter allows the keyboard driver to intercept certain keyboard inputs and immediately act on them. MiniDOS commands are issued by depressing <CLEAR> <SHIFT> and an alphabetic key.

Model II Twinkling Christmas Tree

by Robert Graham

This program is a modification of the Twinkling Christmas Tree program listed in the December 1981 Microcomputer News. Now everyone with a Model II can enjoy the fun. The size of the tree has been expanded from 10 lines to 15 lines.

Instead of PEEK-ing into memory to see what was displayed at a given location, it uses the variable T(Y) to keep track of what was last printed there. And instead of POKE-ing something into display memory, the program uses the PRINT@ command in the graphics mode. Also pressing "Q" during the operation returns you to TRSDOS READY.

```

10 DIM S(225)
   :DIM T(225)
   :CLS
   :X=-3
20 FOR L=1 TO 15
   :X=X+2
30 FOR W=40+(80*L)-((X/2)+.5) TO 40+(80*L)+((X/2)+.5)
40 PRINT@W,CHR$(42)
   :Y=Y+1
   :S(Y)=W
   :T(Y)=1
50 NEXT W
   :NEXT L
60 X=-1
   :FOR Y = 1318 TO 1322
70 PRINT@Y,CHR$(158)
   :NEXT Y
80 PRINT@1466,"M E R R Y C H R I S T M A S"
90 Y=RND(224)+1
   :PRINTCHR$(02)
100 IF T(Y)=1 THEN PRINT@ S(Y),CHR$(43)
   :T(Y)=-1
   :GOTO110
105 PRINT@S(Y),CHR(42)
   :T(Y)=1
110 X=X*-1
   :IF X=1 THEN PRINT@S(1),CHR$(42)ELSE PRINT@
   S(1),CHR$(43)
120 B$=INKEY$
   :IF B$="Q" THEN SYSTEM ELSE 90

```

Model II Bugs, Errors, and Fixes

NOTE TO USERS:

The following program changes and corrections are provided for your information. If you have an applications program which is working correctly, you should probably NOT make any changes to it. If you feel that the changes should be made, but you do not feel qualified to make the changes yourself, please contact your local Radio Shack Computer Center or Expanded Computer Department for assistance. If you do not have access to one of these stores,

then you may want to call Computer Customer Services in Fort Worth for assistance.

PATCHES

PATCHes are entered from TRSDOS READY and are used to make corrections to files stored on the disk.

1. Before making a PATCH, back up the diskette that requires modification and make the PATCHES to the backup copy of the diskette.
2. Apply PATCHES according to the information given in your TRSDOS manual.

MODEL II/16 TRSDOS (26-4150) Version 4.1

These patches are used to permanently change the operation of the TERMINAL program to use channel B in place of channel A. Do not apply these patches unless you have a special situation which requires the use of channel B only. Please note that a serial printer (which also uses channel B) may not be used concurrently with TERMINAL once these patches are applied.

```

PATCH TERMINAL {R=5 B=172 F=3E61 C=3E63}
PATCH TERMINAL {R=5 B=248 F=3E60 C=3E62}
PATCH TERMINAL {R=6 B=164 F=3E64 C=3E65}
PATCH TERMINAL {R=6 B=173 F=3E64 C=3E65}
PATCH TERMINAL {R=6 B=183 F=3E64 C=3E65}
PATCH TERMINAL {R=6 B=193 F=3E61 C=3E63}
PATCH TERMINAL {R=6 B=251 F=3E60 C=3E62}
PATCH TERMINAL {R=8 B=37 F=3E60 C=3E62}
PATCH TERMINAL {R=8 B=86 F=3E60 C=3E62}

```

When an "INIT dr#" is performed on a floppy disk drive, not all records from the previously inserted diskette are cleared from the system. This problem will manifest itself when a "DIR" is done on one diskette containing two to eight files and then an "INIT" is done followed by the insertion of another diskette into the drive. A "DIR" on the new diskette will now show not only its files, but also some files from the first diskette.

The following "PATCH" sequence will remedy this situation.

```

PATCH SYSRES/SYS {R=15 B=88 F=203BC5 C=C52039}
PATCH SYSRES/SYS {R=15 B=99 F=29 C=2A}
PATCH SYSRES/SYS {R=15 B=108 F=0911980E
C=03C3981D}
PATCH SYSRES/SYS {R=30 B=37 F=0000000000000000
C=D511980EEDA0EDA8}
PATCH SYSRES/SYS {R=30 B=45 F=0000000000000000
C=4E23462BC37A0E}
PATCH SYSRES/SYS {R=15 B=116 F=1814EDA0EDA8
C=4E23462BD500}
PATCH SYSRES/SYS {R=15 B=141 F=E5 C=D1}
PATCH SYSRES/SYS {R=15 B=145 F=36FFE1 C=C3A71D}
PATCH SYSRES/SYS {R=30 B=52 F=0000000000000000
C=36FF6069C38A0E}
PATCH SYSRES/SYS {R=15 B=148 F=C1545D7E23666F
C=545D7E23666FC1}

```

PAYROLL (26-4503)

The following is a new ADDENDUM FOR PAYROLL 26-4503 — CONVERTING PAYROLL TO 2.0a

Conversion of prior versions of Payroll to the 2.0a operating system requires a Multi-drive system. Make a Backup of this disk prior to converting.

Follow this procedure for converting your older version diskette only after completing your accounting cycle.

1. Reset the computer and insert the old version of Payroll. Use this version to make at least one Backup of the disk. Keep this Backup in a safe place until the conversion is complete and confirmed by use. This is very important!
 2. Reset the computer and insert a Backup of the new 2.0a Payroll disk in Drive 0. Insert the old Payroll disk in Drive 1.
 3. Type in the command:
XFERSYS :1 DATA <ENTER>
The screen will show:
TRS-80 Model II XFERSYS Utility Ver 2.0
Mount DESTINATION diskette in Drive 1
Press ANY key to continue
Now press any key.
The screen will then show:
Phase 1 now starting
XFERSYS will purge the disk in Drive 1 of all system files, leaving only the Payroll programs and data.
 4. At TRSDOS READY, type in the command:
COPY PR4X:1 TO PR4X:0 <ENTER>
You will see the message:
Existing file. Copy over it (Y/Q)
Press <Y> <ENTER> in response.
 5. AT TRSDOS READY, type in the command:
MOVE */DAT:1 TO :0 <ENTER>
This will move the data files from your old diskette to the new diskette.
The screen will show:
SOURCE disk ready?
Press <Y> <ENTER>
To the right of the data file names displayed on the screen, you will see the message:
Existing file. Copy over it? (Y/Q)
Answer <Y> to PR4CO/DAT
 <N> to PR4COID/DAT This is VERY IMPORTANT
 <Y> to PR4EMPLE/DAT
 <Y> to PR4EARN/DAT
 6. If there have been no error warning messages displayed during the conversion, you can now enter the following data as necessary:
 - Choose option 1 from the Company Setup menu; then choose number 1, Company Name; enter your company information and the State U.C. code (if applicable).
 - From the Company Setup menu, choose option 6, Other Rates and Limits, and enter the basis of supplemental benefits.
 - Reenter the State Tax Tables (if applicable).
- It is recommended that you make a Backup of the new Payroll disk and store your original converted version in a safe place.
Your Payroll on TRSDOS 2.0a is now ready to be used.

PROFILE II (26-4512) Version 1.0

This program requires modifications as described below before it can be FCOPYed for hard disk use.

At TRSDOS READY, apply the following patches to correct the above problem.

```
PATCH CREATE/EFC A=51AE F=34 C=38
```

BEDIT (26-4705) with TRSDOS 4.1

BEDIT does not properly store source files when running under Model II TRSDOS 4.1. The result is garbled source files.

Make the following PATCH to BEDIT:

```
PATCH BEDIT (A=4184 F=3E30CF C=000000)
```

This patch is required for running under TRSDOS 4.1. It is not required for TRSDOS 2.0 or 4.0. It should not be done on TRSDOS 1.2.

BISYNC (26-4715)

The PATCH printed in the September 1982 *Microcomputer News* on page 38 is not valid. Version 3.4 of the BISYNC 3270 package solved the problem. The 3.4 version is now in the process of being released and will be available as an upgrade, at no extra cost.

Microcomputers in Education

Forward to the 3 C's Communicating, Calculating, and Computing

The College of Education at Arizona State University is hosting the 3rd Annual Microcomputers in Education Conference emphasizing the potential of computers to revolutionize the learning process; computers are changing the nature of content in all subject areas, the nature of teaching, and the nature of what it means to be well-educated. A variety of workshops, demonstrations and presentations are designed for anyone interested in creative, innovative microcomputer applications in education.

This two and a half day conference will be held on the Arizona State University campus on March 17-19, 1983. University credit will be available.

A call for conference presenters is extended. Individuals who wish to make a presentation at the 1983 conference should write or call for a proposal form as soon as possible.

Vendors interested in securing a booth are invited to contact the Conference Co-Director. Since exhibit space is limited, reservations will be made on a first come first served basis.

For further information or registration materials write or phone:

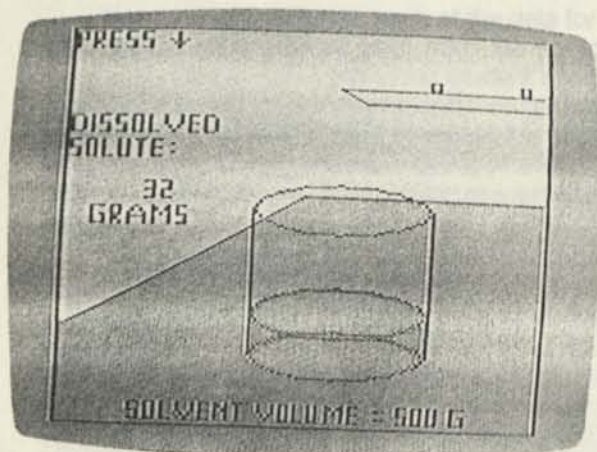
Marilyn Sue Ford
B-47 Payne Hall, College of Education
Arizona State University
Tempe, Arizona 85287
(602) 965-7363 or 965-3322

Chemistry Simulations for Secondary Students

A new Radio Shack courseware program lets secondary students use the TRS-80 computer to simulate chemistry experiments. TRS-80 Chemistry Lab, Volume I is now available for the Model III or Model I 32K disk system or 16K tape system, or for the 16K Color Computer tape system. TRS-80 Chemistry Lab, Volume I supplements classroom instruction for secondary and college chemistry classes. Six different computer programs are included to allow for simulations in six different topics: Kinetic Theory, Charles' Law, Boyle's Law, Conductivity, Solubility, and Titration.

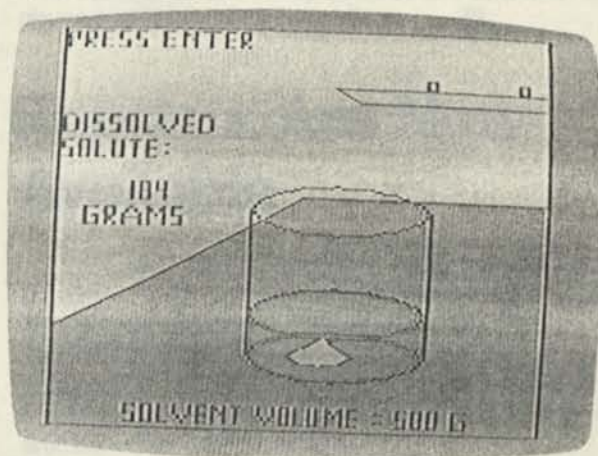
The program design allows students to repeat and vary the simulations many different times, collecting different data each time. For example, the Kinetic Theory, Charles' Law, and Boyle's Law programs let the student choose the volume, and the pressure(s) or temperature(s) of gas for which data will be collected. Thus, numerous different experiments can be performed. Programs are based on equations derived from actual laboratory experiments.

Both the Model III/I version and the Color Computer version of TRS-80 Chemistry Lab, Volume I, lead the student through simulations with helpful on-screen explanations. Not only experimental results, but also correct experimental procedures are realistically simulated. Extensive use of graphics provides a realistic representation of each experiment. For example, the Solubility program shows the student what occurs when you add solute to a solution before the solution has become saturated:

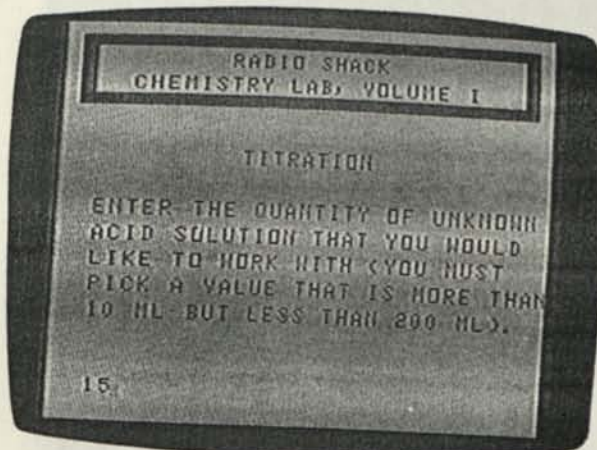


If the student continues to add solute once the solution has become saturated, the screen shows the solute piling up

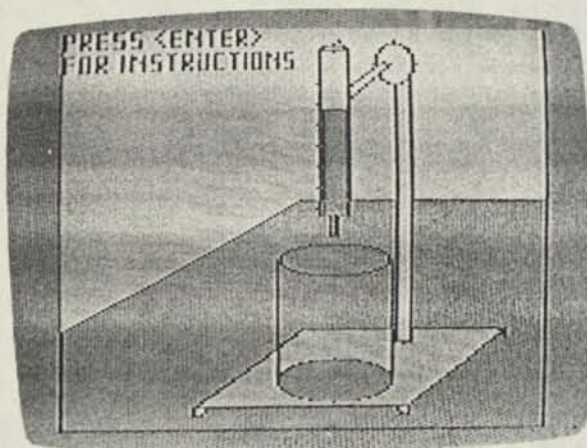
on the bottom of the beaker rather than dissolving:



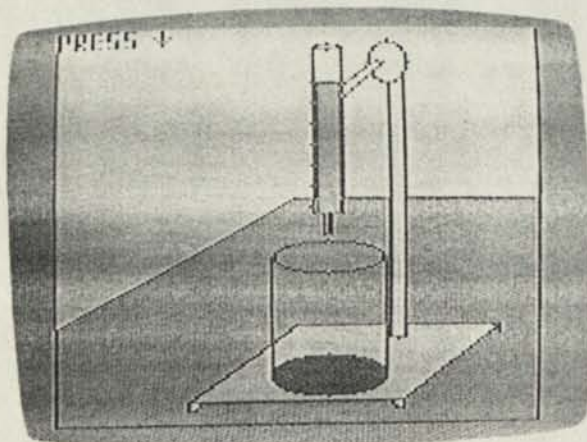
These simulations are not intended to replace experience in a real laboratory, but they do provide an easy, safe way for students to see the results of an experiment again and again, and to collect, graph, and analyze the data produced by these experiments, with the aid of the computer. The Titration program provides a good example of how the simulations reinforce correct experimental procedures. The Titration simulation leads the student through several steps to collect and refine the data that he or she needs. First, the student chooses a base solution and then chooses the quantity of unknown acid solution to work with:



Then the beaker is filled, and the student fills the buret:



Then the student uses the down-arrow key to add drops of the base to the acid. When the solution has been titrated, it changes colors—just as it would in a real experiment if litmus dye were used:



The computer has kept track of the data for the titration. The student then records data from the computer screen onto a worksheet (provided in the student experiment book that accompanies the program), and then presses <ENTER> to continue. The next screen presents the student with the equation used to calculate the molarity of acid for this titration. The student fills in the appropriate amounts from the data worksheet and solves for the molarity of the acid.

TRS-80 Chemistry Lab, Volume I includes an instructor manual and twenty-five student experiment books. The student experiment books include background information and experimental procedures for each topic, a short summary quiz over the background information, data worksheets for recording the results of the experiments, and a set of questions over the data acquired in each topic. Step-by-step instructions for using the program on the TRS-80 computer are also included. The teacher's manual includes all of this information, plus instructional objectives for each topic, answers to the summary quizzes and final questions, and suggestions for areas of emphasis and methods of presentation.

Radio Shack's TRS-80 Chemistry Lab, Volume I, is available at Radio Shack stores and Computer Centers. The Color Computer version of the program (Cat. No. 26-2626) can be used with a 16K Color Computer tape system with Extended Color BASIC. The Model III/I version (Cat. No. 26-2609) is designed for use with the Model III or Model I 32K disk system or 16K tape system. This Model III/I version includes a Model III diskette and a 1500 Baud cassette tape for use with the Model III tape system. Replacement software for Model I users can be obtained FREE from Radio Shack. (The teacher's manual for the package tells you how.) The suggested retail price of either package (26-2609 or 26-2626) is \$199.00. (Prices may vary at individual stores and dealers.)

TRS-80 Chemistry Lab, Volume I, is the first of several simulation programs to be released by Radio Shack. A second volume of chemistry simulations and two volumes of physics simulations are to follow.

Reprints Are Now Available

The 1981 reprint set (Catalog No. 26-2240, \$9.95) is now available. This second set includes all 1981 issues of the TRS-80 Microcomputer News plus the 1981 index. "Stop That Out-of-Sorts Feeling", a two dimensional machine language sort for the Models I/II, and "Printer Codes—the Inside Story" are only two of the many useful and informative articles in the 1981 Reprints.

The response to the first TRS-80 Microcomputer News Reprints (Catalog No. 26-2115, \$4.95) has been even better than expected. For those who may not be familiar with it, this set of reprints includes all issues of TRS-80 Microcomputer News from the first-ever-issue through December 1980.

While a reprint of the 1982 TRS-80 Microcomputer News is planned, it will not be available until late summer or early fall of 1983.



Personal Time Savers Revisited

OOPS!!!

Calls from frustrated readers regarding the September VisiCalc Personal Time Savers column have resulted in this follow up article. The ideas presented are in fact valid, but there is additional information which was not included in the previous article. We apologize for any time loss or confusion that may have resulted from this oversight.

THE GENERAL IDEA IS . . .

Keyboard commands like /IR, >A5, /F\$, and formulas like +A5+A6 can be entered into a VisiCalc sheet as labels by using the preceding " (quote mark), stored in Print File format, and read back into another VisiCalc sheet as a command file. While immediate applications for this capability may not spring to mind, playing around with the examples a little may give you additional insight into several practical applications of this capability.

STORING COMMANDS AND FORMULAS IN A VISICALC PRF FILE

The coordinates, which are entered into the VisiCalc sheet as labels by using a quotation mark ("), indicate the coordinate where the values (10) or formula (+A5+A6) will actually be stored when the file is loaded back into another VisiCalc screen. The first example is a sample command file which stores the value 10 at coordinates A5 and A6 and the formula +A5+A6 at coordinate A7. The following key-strokes setup A5, the first screen coordinate of the file, and the value to be stored at that position.

```
">A5:10 <ENTER>
```

The " is for a label; >A5 indicates that the formula or value will be stored at coordinate A5; the : must follow the coordi-

nate in this format or this procedure will not work; and 10, in this example, is the value that is stored at A5.

A value or command may be stored at the coordinate by entering it:

1. Right after the colon (A5:10)
2. In the next coordinate position.

For example, the format /F\$ can be stored at coordinate A1 by typing ">A1:/F\$" or by typing ">A1: at coordinate A1 and "/F\$" at coordinate A2.

To create the first file, enter the information shown in Screen 1. Be sure that all commands, labels, or formulas are entered as labels. Numeric items entered in a cell by themselves are entered as numbers, not labels.

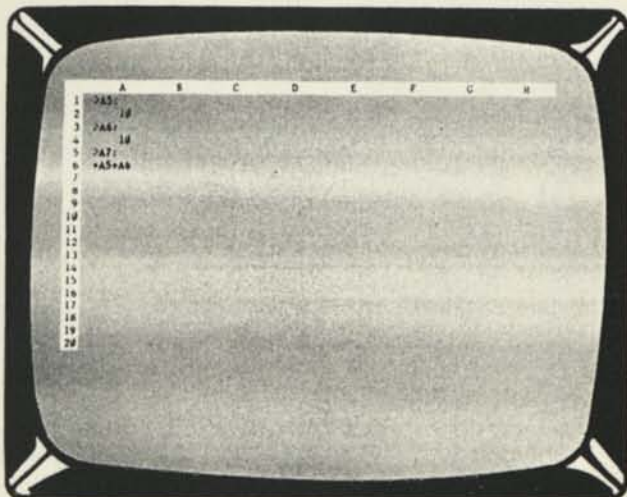
SAVING THE FILE TO DISK

When all the information is entered, move the cursor to coordinate A1 and then save the file to disk by typing:

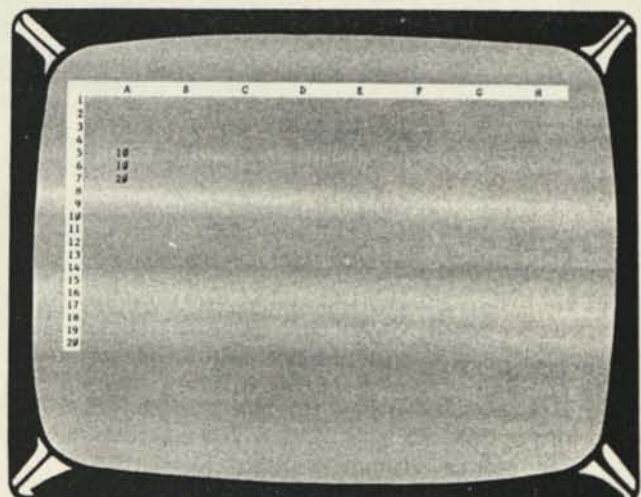
```
/PF filename/ext <ENTER>
```

Notice that the file is not being saved by the standard /SS procedure. /PF saves the file in a special format that enables it to be read back as a command file. "filename" is the name you assign to the file. For demonstration purposes, the filename will be ADD. The extension (/ext) is optional, but if an extension is not used, VisiCalc will add a /PRF extension to the filename. When the file is reloaded into VisiCalc, the extension will have to be included in the filename. Instructions for saving and reloading the ADD file are:

1. Save it to disk by /PF ADD <ENTER>
 2. Clear the screen - /CY
 3. Reload ADD with /SL ADD/PRF <ENTER>
- The result is Screen 2.



Screen 1



Screen 2

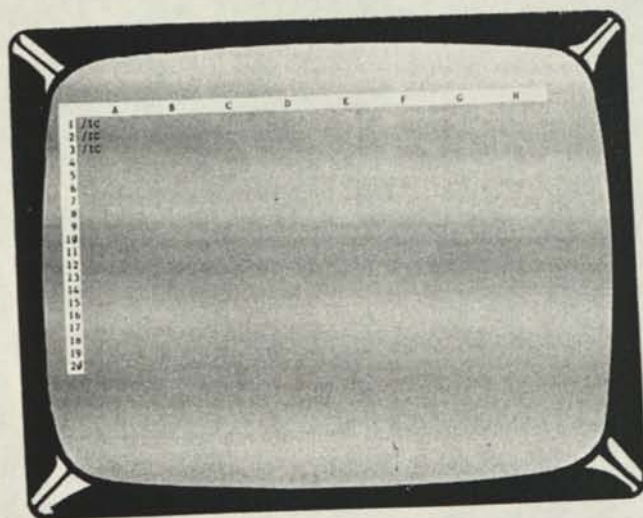
You should also be aware that /PF stores only what appears on the VisiCalc screen. Therefore, the column should be wide enough to display the entire formula.

A NOTE ON COMMAND FILE SEQUENCING

VisiCalc reads commands from left to right just as you are reading this text. So you could write the commands at A1, B1, C1, . . . , and then the next set of commands could start at A2, B2, C2, It is also possible to write the commands vertically—the first command at A1, the second command at A2, the third at A3 and so on, but a word of caution is needed. Since VisiCalc reads the file from left to right, there would be problems if the command file were written beginning at A1 for the first command, A2 for the second command, A3 for the third command, and then B1 for the fourth command, B2 for the fifth command, B3 for the sixth command, because VisiCalc would read them A1, B1, A2, B2, A3, B3 which would not be the intended order. While it is possible to have multiple rows of commands read in a left to right direction, it is not possible to write them in multiple columns.

MULTIPLE ROW AND COLUMN INSERTS

The September command file which inserted multiple rows worked just as it was given. This example will discuss inserting multiple columns. Try creating a command file like the one shown in Screen 3.

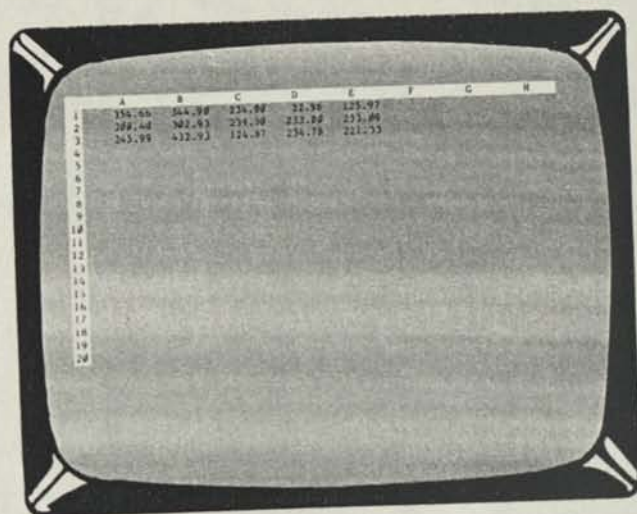


Screen 3

When stored with the /PF format and reloaded into a VisiCalc screen, this file will insert three columns beginning at the present cursor position. When creating the file remember to enter each /IC instruction as a label. The completed screen appears as Screen 3. Move the cursor to coordinate A1, and save the file in the /PF format as INS3COL.

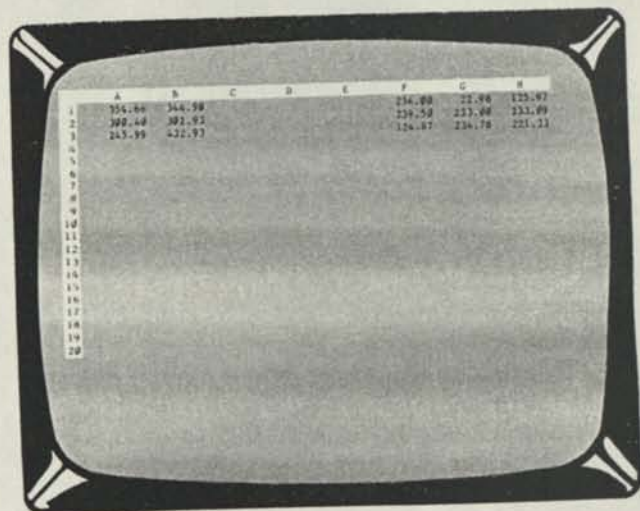
For a demonstration of the effect of INS3COL when it is loaded into a VisiCalc screen, first look at Screen 4 which contains several columns of numbers.

Three columns between the B and C columns are inserted by placing the cursor on column C and loading



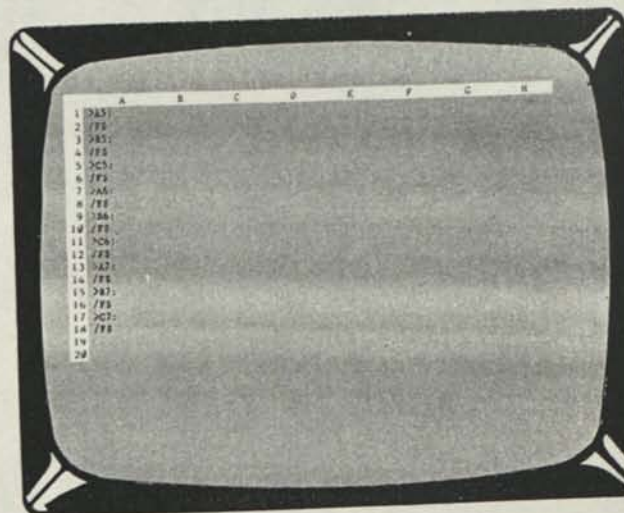
Screen 4

INS3COL/PF. The result is Screen 5.



Screen 5

A DECIMAL NUMBER MATRIX COMMAND FILE



Screen 6

(Continued on Page 38)

Modem II Programming Techniques

To correctly program the Modem II, you must follow certain conventions. First, initialize the RS-232C port. The Modem II/RS-232C protocol is 300 baud, 8 bit words, no parity, and 1 stop bit. Set these prior to programming the Modem. For example, on the Model II, you might initialize by typing (from TRSDOS READY):

```
SETCOM A=(300,8,N,1) <ENTER>
```

Set the Modem II switches to AUTO, ORIG, and TEST OFF. Now turn the Modem II POWER switch to ON. Notice that the RD indicator light is on, and remains on for about six seconds. When the RD light goes off, the TR indicator light should turn on. You MUST wait until the TR light is on before trying to program the Modem II! Once the TR light comes on, send an "*" to the modem. The TR light should turn off. If it doesn't, keep sending "*"s.

If the TR light fails to turn on, you might have a problem in the connections or have the protocol set incorrectly.

Once the TR light is off, you may begin programming the modem. Programming normally involves the following:

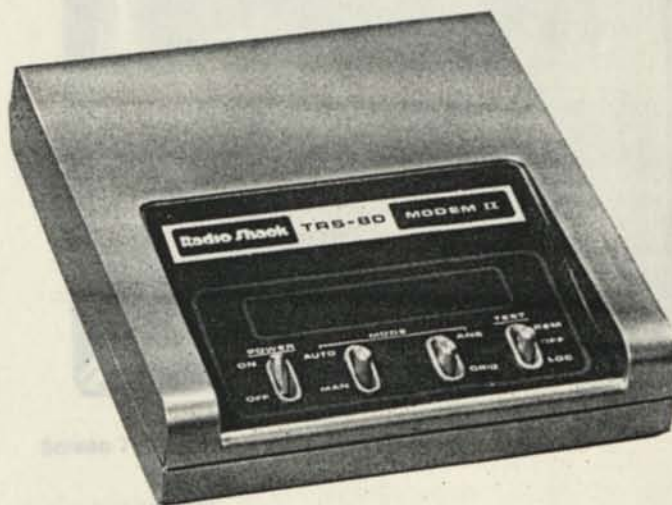
1. Send a "D" to clear the modem's program memory. This only clears the number—dialing speed and style remain at their default values of Slow Rotary.
2. If you are using rotary dialing, next send out the dialing speed, either "F" for fast, or a "S" for slow. Follow this by a "D", so that you can enter the phone number.
3. Now send the dialing style, either an "R" for rotary or a "T" for tone.
4. Enter the phone number. Valid characters include numbers and the letter "P" (for pause). Note: Do NOT enter hyphens, spaces, or parentheses! If you make a mistake, send a "D" to erase the old number and start over at Step 3.
5. When you are ready to place the call, send an "X". Once the modem has dialed the number, it waits on the other party to answer. If there is no answer after three rings, the modem disconnects, and the TR light comes on. To re-try the number, send an "*" and an "X".

Note: While entering the data into the modem, you must wait for the modem to correctly echo the last character sent BEFORE sending the next character. If the modem's echo doesn't match the character you sent, then a problem exists. Check your connections, the protocol, etc.

SAMPLE MODEM II PROGRAMS

Here are some examples of programs for the Modem II along with their interpretations. We'll assume that you are using a "terminal" program to send these to the modem.

Program	Interpretation
*DFDR5552234X	Select fast rotary dialing, then dial the number 555-2234.
*DT9P5551231X	Select tone dialing, dial 9 then pause for two seconds, and then dial 555-1231.
*DT9PPP1P8175559876X	Select tone dialing, dial 9 and pause for six seconds, dial one and pause for two seconds, then dial the number (817) 555-9876.
*DSDR2851X	Select slow rotary dialing and dial the number 2851.
*D2851X	Select slow rotary (as the default on Power up) and dial the number 2851.



SAMPLE DRIVER FOR AUTOMATIC DIALING

Here is an example of a driver program which sends a program to a modem. It is written for a Model II. To modify it for use on a Model III, simply remove the DEFUSR statements and replace the USR0 statements with OUT &HEB, SND% and the USR1 statements with ECHO%=INP(&HEB).

(Note: Naturally, you would want to use this in conjunction with a communications package. The following shows merely how to program the modem, and nothing about what to do once the communications are set up.)

```
10 'Sample Modem II Auto Dial Program written in
   ' Model II BASIC
20 '
30 ' Note: Use Channel A to run the program.
   ' First initialize this channel with the
```

```

40 ' TRSDOS command SETCOM A=(30,8,N,1). Load the
50 ' OUTP and INP routines, and then load BASIC
60 ' with the command BASIC -M:HEF50
70 '
80 DEFUSR0 = &HEF50      'OUTP Routine
90 DEFUSR1 = &HEF54      'INP Routine
100 '
110 ' Get the phone number data
120 '
130 LINE INPUT "Enter Phone Number: ";PH$
140 INPUT "Enter Dialing Style (<R>otary or
    <T>one): ";STYLE$
150 IF STYLE$ = "R" THEN INPUT "Enter Dialing
    Speed (<F>ast or <S>low): ";SPEED$ ELSE
    SPEED$=""
160 PGM$ = "D" + SPEED$ + "D" + STYLE$ + PH$
170 '
180 'Enter the program mode of the Model II by
    sending an "*"
190 '
200 TRIES = 0
210 SND% = CINT(ASC("*"))
220 X% = USR0(SND%)
230 FOR I = 1 TO 50
    : NEXT I
240 ECHO% = USR1(0)
250 TRIES = TRIES + 1
260 IF ECHO% <> SND% AND TRIES < 3 THEN 210
270 IF TRIES = 3 THEN PRINT "Can't program the
    Modem!"
    : STOP
280 '
290 ' Send the phone number data string (i.e.,
    the Modem II program)
300 '
310 FOR I = 1 TO LEN(PGM$)
320 SND% = CINT(ASC(MID$(PGM$,I,1)))
330 X% = USR0(SND%)
340 FOR N = 1 TO 50
    : NEXT N
350 ECHO% = USR1(0)
360 IF SND% <> ECHO% THEN PRINT "Can't program
    the Modem!"
    : STOP
370 NEXT I
380 '
390 ' Get the time for the number to be dialed
    and wait for the dialing time
400 ' (DT$) to equal the system time (TIMES).
    When equal, send a "X" to instruct the
410 ' Modem II to dial the number.
420 '
430 INPUT "Modem Programmed. Enter Time to Dial
    Number (HH.MM.SS)"; DT$
440 IF DT$ <> TIMES THEN 440
450 SND% = CINT(ASC("X"))
460 X% = USR0(SND%)
470 FOR I = 1 TO 50
    : NEXT I
480 ECHO% = USR1(0)
490 IF SND% = ECHO% THEN PRINT "Number Dialed!"
    ELSE PRINT "Number NOT Dialed!"
500 PRINT "Number Dialed!"

```

OUT/INP SUBROUTINES

Use these routines to input and output characters via the RS232-C Channel A port. On entry to OUTP, HL points to the character to be sent. On entry to INP, HL points to the address that is to receive the character. This is consistent with Model II BASIC's method of parameter passing.

```

PORTA EQU 0F4H      ;Channel A Data Port=0F4H
PSECT 0EF50H      ;Routines must reside in
                  high memory

```

```

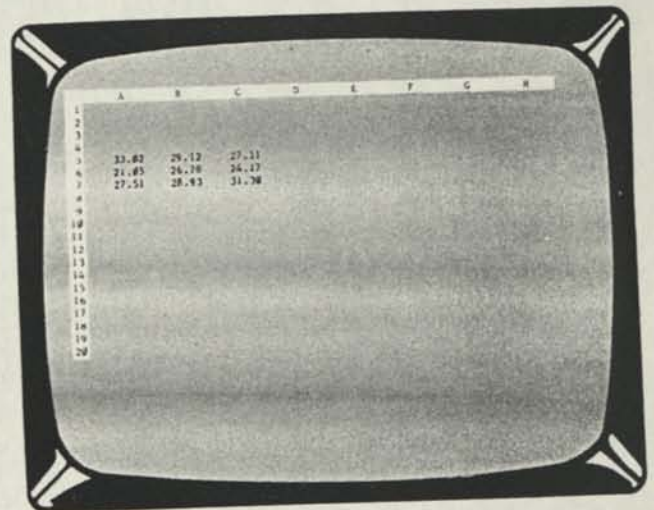
OUTP: LD    A,(HL)      ;Get character from BASIC
                        program
      OUT  (PORTA),A    ;Send character out Channel
                        A
      RET                ;Return to BASIC
;
INP:  IN   A,(PORTA)    ;Send character from Channel
                        A port
      LD   (HL),A       ;Return character to BASIC
                        (HL -> character)
      RET                ;Return to BASIC
;
      END  OUTP

```

Personal Time Savers (From Page 36)

This command file presets the value format to a dollar format for all numbers that will be entered in the following matrix positions: A5, B5, C5, A6, B6, C6, A7, B7, C7. The setup of this file would appear as Screen 6.

After saving this file in the /PF format, clearing the screen, and loading it back in, numbers entered in the matrix bounded by A5 - A7 and C5 - C7 will be displayed as decimal numbers as in Screen 7.



Screen 7

TIME SAVERS

These procedures can be true time savers by allowing you to eliminate repetitive entries by entering them once, saving the file, and reloading them into another file when needed.

Christmas Eve

Todd Day
6702 Mt. Pakron Dr.
San Jose, CA 95120

Here is a program that I thought you might enjoy for the holidays. It draws a nice Christmas Eve scene complete with a Christmas tree with a blinking star, lights, and a warm, cozy fire. This program requires 16K and Extended BASIC. Since this program uses up almost all of the available memory, it cannot be run while a disk system is connected.

One programming technique I have found useful in making graphics programs is to stick this line in the last part of your program:

```
9000 H = JOYSTK(0) * 4
: V = JOYSTK(1) * 3
: PSET(H,V)
: GOTO 9000
```

This allows you to see all of your graphics and move the dot from the joystick around. When you find the spot you need, you can "break" the program, and by printing the values of H and V you can find the required coordinates.

CHRISTMAS EVE PROGRAM

- Line 10 Reserves space for graphics and sets up the screen. L and L1 are the coordinates for the lights on the tree.
- Line 15 Draws the walls.
- Line 20 Paints the floor and starts drawing the fireplace mantle.
- Line 25 Finishes the bricks and makes the fireplace hole.
- Line 30 Draws the rug.
- Line 35 Paints the right wall blue and draws the window.
- Line 40 Paints the left wall blue, makes the picture frame and draws the Christmas tree.
- Line 45 Draws the wreath and tree base.
- Line 50 Draws the vases.
- Line 55 Draws the presents and stockings.
- Line 60 Draws the log and star.
- Line 65 Draws "Merry Christmas."
- Line 70 - 90 Blinks the star and lights, flickers the fire, and plays "We Wish You a Merry Christmas."

```
10 CLEAR 0
: PCLEAR 8
: PMODE 3,1
: PCLS 2
: SCREEN1,0
: COLOR7
: DIM L(16), L1(16)
: FOR T = 0 TO 16
: READ L(T), L1(T)
: NEXT
15 LINE(0,20)-(30,30),PSET
: LINE-(255,181),PSET,B
: LINE -(255,191), PSET
```

```
: LINE(0,255) - (30,181), PSET
: LINE(255,20) - (225,30), PSET
: LINE -(30,110), PSET, BF
: LINE(0,20) - (255,19), PSET, B
: PAINT(0,0), 1, 3
20 PAINT(245,189),1,3
: PAINT(33,120), 8, 7
: COLOR 2
: FOR T = 174 TO 117 STEP - 7
: LINE(33,T) - (222,T), PSET
: NEXT
21 COLOR 7
22 LINE (30,30) - (225,181), PSET, B
25 COLOR 2
: FOR T = 111 TO 180 STEP 14
: FOR TT = 40 TO 218 STEP 20
: LINE (TT,T) - (TT,T+7), PSET
: LINE (TT + 10, T + 7) - (TT + 10, T + 14),
PSET
: NEXT
: NEXT
: COLOR 7
: LINE (100,125) - (160,174), PSET, BF
30 CIRCLE (132,191), 30, 3, .3, .5, 0
: PAINT(132,189), 3, 3
: CIRCLE(132,191), 15, 4, .3, .5, 0
: PAINT (132,191), 4, 4
35 LINE (255,64) - (230,68), PSET
: LINE - (230,116), PSET
: LINE -(255,120), PSET
: PAINT (252,24), 7, 7
: LINE(230,92) - (255,92), PSET
: LINE(243,64) - (243,120), PSET
40 PAINT (2,27), 3, 3
: COLOR 4
: LINE (30,30) - (30,100), PSET
: COLOR 4
: LINE (96,45) - (164,99), PSET, BF
: COLOR 2
: LINE (99,48) - (162,97), PSET, BF
: COLOR 1
: LINE (44,57) - (0,174), PSET
: LINE - (88,174), PSET
: LINE - (44,57), PSET
: PAINT (45,60), 1, 1
45 CIRCLE (130,73), 22, 1
: PAINT (130,73), 1, 1
: CIRCLE (130,73), 13, 2
: PAINT (130,73), 2, 2
: COLOR 4
: LINE (126,87) - (137,90), PSET, BF
: CIRCLE (44,191), 30, 4, .3
: PAINT (44,188), 4, 4
: COLOR 2
: LINE (40,174) - (48,187), PSET, BF
50 COLOR 5
: LINE (80,93) - (82,110), PSET
: LINE - (86,110), PSET
: LINE - (88,93), PSET
: LINE - (80,93), PSET
: PAINT (82,95), 5, 5
```

```

: LINE (172,93) - (174,110), PSET
: LINE -(178,110), PSET
: LINE -(180,93), PSET
: LINE -(172,93), PSET
: PAINT (174,95), 5, 5
: COLOR 8
: LINE(222,30) - (222,112), PSET
55 PAINT (240,87), 6, 7
: COLOR 2
: LINE (16,179) - (30,189), PSET, BF
: COLOR 3
: LINE (56,177) - (76,189), PSET, BF
: DRAW "BM114,123C2D11G2L4G2D2F2R10E4U15L6"
: PAINT (118,126), 2, 2
: DRAW "BM138,123D11G2L4G2D2F2R10E4U15L6"
: PAINT (142,126), 2, 2
60 LINE (112,165) - (148,174), PSET, BF
: COLOR 4
: LINE (45,56) - (52,60), PSET
: LINE -(46,52), PSET
: LINE -(54,49), PSET
: LINE -(46,49), PSET
: LINE -(44,43), PSET
: LINE -(42,49), PSET
: LINE -(35,49), PSET
: LINE -(42,52), PSET
: LINE -(36,60), PSET
: LINE -(45,56), PSET
65 DRAW "BM62,15U8F2E2D8BR6NR4U4NR2U4R4BR6N
D8R2F2G2L2F4BR6U8R2F2G2L2F4BR8U4NH4NE4D4BR14U
8NR4D8R4BR6U4NU4R4NU4D4BR6U8R2F2G2L2F4BR6R2NR
2U8L2R4BR6NR4D4R4D4NL4BR8U8L2R4BR6ND8F2E2D8BR
6U4NR4U4R4D8BR6R4U4L4U4R4
70 FOR I = 1 TO 4
: PCOPY T TO T + 4
: NEXT
75 PMODE 3,5
: X = X + 1
: IF X > 2 THEN X = 0
: GOTO 75
: ELSE PAINT (44,51), X + 1, 4
77 Y = Y + 2
: IF Y > 21 THEN Y = 0
: PLAY "T3O2L4GO3CL8COCO2BL4AAP32A03DL8DE
DCO2L4BGP32GO3EL8EFEDL4CO2AP32L8GGL4AO3CO2BO
3C
80 R = RND(17) - 1
: CIRCLE (L(R),L1(R)), 3, RND(3) + 1
: COLOR 3
: LINE(148,164) - (112,148), PSET, BF
: ON RND(2) GOSUB 85, 90
: FOR T = 5 TO 8
: PCOPY T TO T - 4
: NEXT
: GOTO 75
85 COLOR 2
: LINE (148,164) - (139,152), PSET
: LINE -(130,162), PSET
: LINE -(121,148), PSET
: LINE -(112,164), PSET
: PAINT (120,156), 4, 2
: RETURN
90 COLOR 2
: LINE (148,164) - (139,149), PSET
: LINE -(127,162), PSET
: LINE -(121,153), PSET
: LINE -(112,164), PSET
: PAINT (120,156), 4, 2
: RETURN
95 DATA 44, 66, 48, 83, 36, 87, 33, 102, 44, 102,
56, 105, 56, 120, 44, 120, 24, 120, 20, 138,
44, 138, 64, 142, 72, 162, 52, 162, 36, 168,
12, 162, 31, 153

```



Computer Clubs

Northern Illinois Color Computer Club
 c/o John Bowman, President
 9346 Landings
 Des Plaines, IL 60016

St. Ignatius Computer Club
 St. Ignatius High School
 c/o Stephan Kinholt
 P. O. Box 400
 St. Ignatius, MT 59865

We apologize for the error in the September 1982 Microcomputer News. The listing for the Silicon Valley Computer Club should have read:

Silicon Valley Color Computer Club
 P.O. Box 61593
 Sunnyvale, Ca 94088

There is also a color computer bulletin board for the area called:

Silicon Rainbow Bulletin Board
 408-733-6809

TRS-80 Users Group of North Utah
 c/o Marcel Kinard
 3106 East 125th North
 Layton, UT 84041
 1-801-544-8970

Text Editor Revisited

Ashok Basargekar
1423 N. Cleveland St.
Orange, CA 92667

The Text Editor 2.2 program published in the July/August 1982 issue of TRS-80 Microcomputer News is really a great one. The program of course works fine when a Color Computer with Extended BASIC is used. It requires quite a bit of modification when using Disk Extended Color BASIC.

Since additional system RAM from address 1536 to 2440 is reserved for the Disk BASIC, it is necessary to change the Assembly Language code and the BASIC program so that the Text Editor starts after the reserved RAM for the Disk BASIC.

I feel that the readers having Disk BASIC and a 32K Color Computer will find the following modifications to the Text Editor very useful while using the program. I have only listed the lines that need to be changed and the new lines that are required to be added.

I have deleted the lines from 5000 to 5050 to my program and have used the Assembly Language code as a separate Machine Language program called TEXT ML. My modifications are as follows:

```
10 CLEAR 200, 32500
   : LOADM "TEXT ML"
20 CLS
   : INPUT "LINE WIDTH"; LW
30 INPUT "MARGIN"; TB
30 DEFUSR0 = &H7FEB
   : DEFUSR1 = 32719
   : DEFUSR2 = &H7FB2
40 CLS
   : PRINT "1: COMPOSE"
   : PRINT "2: EDIT"
   : PRINT "3: COPY ON PRINTER"
   : PRINT "4: SAVE TEXT ON DISK"
   : PRINT "5: LOAD TEXT FROM DISK"
   : PRINT
   : PRINT "SELECT 1 TO 5"
41 KY$ = INKEY$
   : IF KY$="" THEN 41 ELSE IF VAL(KY$) < 0 OR
   VAL(KY$) > 5 THEN 41
42 ON VAL(KY$) GOTO 100, 210, 550, 4050, 4100
   : GOTO 20
120 A = 2560
170 IF A$ = "" THEN POKE A, 255
   : LIMIT = A
   : GOTO 20 ELSE IF A$ = CHR$(8) THEN A = A -
   1
   : GOTO 140
```

Delete line 180

```
210 POKE 65479, 0
   : POKE 65480, 0
   : POKE 65483, 0
220 A = 2560
270 IF A$ = CHR$(8) THEN A = A - 1
   : IF A = 2559 THEN A = 2560
290 IF A$ = "-" THEN A = A - 32
   : IF A < 2560 THEN A = 2560
530 IF A$ = "-" THEN POKE A, 255
   : LIMIT = A
   : GOTO 230
550 CLS
   : PRINT "TEXT PRINTING ON PRINTER. TURN
   PRINTER ON AND PRESS ENTER."
   : INPUT R$
```

```
   : FOR A = 2560 TO 9215
620 IF LEN(B$) = LW THEN 3050
640 PRINT# - 2, TAB(TB); B$
   : B$ = ""
650 IF A1 = 255 THEN LIMIT = A
   : GOTO 20 ELSE IF A = 9215 THEN LIMIT = 9215
   : GOTO 20
3000 IF A > 7679 THEN POKE 65479, 0
   : POKE 65481, 0
   : POKE 65483, 0
   : POKE 65485, 0
   : RETURN
3001 IF A > 7167 THEN POKE 65478, 0
   : POKE 65481, 0
   : POKE 65483, 0
   : POKE 65485, 0
   : RETURN
3002 IF A > 6655 THEN POKE 65479, 0
   : POKE 65480, 0
   : POKE 65483, 0
   : POKE 65485, 0
   : RETURN
3003 IF A > 6143 THEN POKE 65478, 0
   : POKE 65480, 0
   : POKE 65483, 0
   : POKE 65485, 0
   : RETURN
3004 IF A > 5631 THEN POKE 65479, 0
   : POKE 65481, 0
   : POKE 65482, 0
   : POKE 65485, 0
   : RETURN
3005 IF A > 5120 THEN POKE 65478, 0
   : POKE 65481, 0
   : POKE 65482, 0
   : POKE 65485, 0
   : RETURN
3006 IF A > 4607 THEN POKE 65479, 0
   : POKE 65480, 0
   : POKE 65482, 0
   : POKE 65485, 0
   : RETURN
3007 IF A > 4095 THEN POKE 65478, 0
   : POKE 65480, 0
   : POKE 65482, 0
   : POKE 65485, 0
   : RETURN
3010 IF A > 3583 THEN POKE 65479, 0
   : POKE 65481, 0
   : POKE 65483, 0
   : POKE 65484, 0
   : RETURN
3020 IF A > 3071 THEN POKE 65478, 0
   : POKE 65481, 0
   : POKE 65483, 0
   : POKE 65484, 0
   : RETURN
3030 POKE 65479, 0
   : POKE 65480, 0
   : POKE 65483, 0
   : RETURN
3050 IF PEEK(A+1) = 143 THEN 640
3055 IF PEEK(A+1) = 32 THEN 640
3060 IF A1 >= 32 AND A1 < 48 THEN 640
3070 IF A1 > 57 AND A1 < 64 THEN 640
3080 A = A - 1
   : B$ = LEFT$(B$, LEN(B$)-1)
   : A1 = PEEK(A)
3090 IF A1 > 0 AND A1 < 27 THEN A1 = A1 + 96
4000 IF A1 = 143 THEN A1 = 32
4010 IF A1 = 255 THEN 640
4020 IF A1 = 191 THEN 640
4030 GOTO 3050
4050 CLS
   : INPUT "FILE NAME"; F$
   : SAVEM F$, 2560, LIMIT, 2560
```

```

: GOTO 20
4100 CLS
: INPUT "FILE NAME"; FS
: LOADM FS
: GOTO 20

```

TEXT ML ASSEMBLY LANGUAGE PROGRAM

```
SAVEM "TEXT ML", 32501, 32767, 32690
```

```

7FB2 3E0A00 LDX #A00
7FB5 A680 LDA X+
7FB7 E684 LDB X+0
7FB9 A780 STA X+
7FBB 1E89 EXC A.B
7FBD 81FF CMPA #FF
7FBF 2707 BEQ 7FC8
7FC1 8C11FF CMPX #11FF
7FC4 2702 BEQ 7FC8
7FC6 20EF BRA 7FB7
7FC8 E784 STB X+0
7FCA 86FF LDA #FF
7FCC A784 STA X+0
7FCE 39 RTS
7FCF 8E0000 LDX #0
7FD2 E684 LDB X+0
7FD4 E782 STB -X
7FD6 A681 LDA X++
7FD8 E684 LDB X+0
7FDA C1FF CMPB #FF
7FDC 270A BEQ 7FE8
7FDE 8C11FF CMPX #11FF
7FE1 2705 BEQ 7FE8
7FE3 E782 STB -X
7FE5 7E7FD6 JMP 7FD6
7FE8 E782 STB -X
7FEA 39 RTS
7FEB C68F LDB #8F
7FED 8E0A00 LDX #A00
7FF0 E780 STB X+
7FF2 8C0BFF CMPX #BFF
7FF5 26F9 BNE 7FF0
7FF7 39 RTS

```

```
13 IF Z=11 LET Y$="ELEVNTH"
```

The program is interesting from a technical standpoint because it uses IF statements to control PRINT or PAUSE commands. If the conditions specified in the IF statement are not met, then the PRINT or PAUSE is not executed. This is a useful programming trick because it allows you to format "error messages" and special messages that will appear only under certain conditions.

```

1 "XMAS"
2 " "CLEAR
:PAUSE "TWELVE DAYS OF CHRISTMAS"
3 FOR Z=1 TO 12
:IF Z=1 LET Y$="FIRST"
4 IF Z=2 LET Y$="SECOND"
5 IF Z=3 LET Y$="THIRD"
6 IF Z=4 LET Y$="FOURTH"
7 IF Z=5 LET Y$="FIFTH"
8 IF Z=6 LET Y$="SIXTH"
9 IF Z=7 LET Y$="SEVENTH"
10 IF Z=8 LET Y$="EIGHTH"
11 IF Z=9 LET Y$="NINTH"
12 IF Z=10 LET Y$="TENTH"
13 IF Z=11 LET Y$="ELEVENTH"
14 IF Z=12 LET Y$="TWELFTH"
15 PAUSE "ON THE ";Y$;" DAY"
:PAUSE "OF CHRISTMAS, MY TRUE"
:PAUSE "LOVE GAVE TO ME"
16 IF Z>11 PAUSE "TWELVE LORDS A LEAPING"
17 IF Z>10 PAUSE "ELEVEN LADIES DANCING"
18 IF Z>9 PAUSE "TEN PIPERS PIPING"
19 IF Z>8 PAUSE "NINE DRUMMERS DRUMMING"
20 IF Z>7 PAUSE "EIGHT MAIDS A MILKING"
21 IF Z>6 PAUSE "SEVEN SWANS A SWIMMING"
22 IF Z>5 PAUSE "SIX GEESE A LAYING"
23 IF Z>4 PAUSE "FIVE GOLDEN RINGS"
24 IF Z>3 PAUSE "FOUR CALLING BIRDS"
25 IF Z>2 PAUSE "THREE FRENCH HENS"
26 IF Z>1 PAUSE "TWO TURTLEDOVES AND"
27 PAUSE "A PARTRIDGE IN A"
:PAUSE "PEAR TREE"
:NEXT Z
28 PAUSE "MERRY CHRISTMAS TO YOU!"
:END

```

PC-1 Twelve Days of Christmas

Robert K. Phelps
143 Forest Park
7800 Tayloe Drive
Manassas, VA 22111

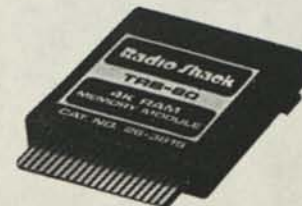
This program is based on the song "The Twelve Days of Christmas." It can be run on a PC-1 as a "fun" program. It can also be adapted to other computers. The program takes only 28 program lines so it is easy to key into memory. After the program is loaded, you switch to the DEF mode and press <SHFT> <SPACE> to start the run. The program can be converted to use the printer, but it is mainly designed to use the computer as a stand-alone unit.

There is a program bug that was left in deliberately. This is on line 13. The computer will print ON THE ELEVENT DAY. The final H is dropped because it exceeds the 7-letter limit for string values. This is such a minor problem that it is not worth the time and effort to correct the bug. It does not affect the running of the program. If you want a program correction, then you can drop the E between the V and N on line 13. This will change line 13 as follows.

Availability of 16K RAM Modules for the Pocket Computer II

There have been many inquiries regarding availability dates for the 16K RAM module for the Pocket Computer II. There simply is not an availability date in existence at this time. Presently it is not possible to manufacture CMOS memory chips in the density necessary to produce a 16K module.

Radio Shack will have 16K modules as soon as it is feasible to produce them, but, for the present, a 16K RAM module looks to be a minimum of one to two years away.

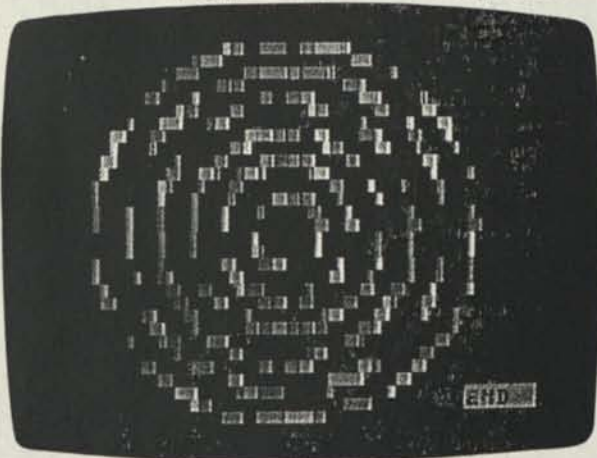


Concentric Circles on the CoCo

Donald E. Whitelock
402 Ganttown Road
Turnersville, NJ 08012

In your June 1982 issue I was particularly interested in a program by Stephen Havens entitled "Concentric Circles." I was disappointed when it would not run on my Color Computer so I updated the program to run on the Color Computer and discovered the best I could come up with was concentric ellipses.

Also, in your June issue I found an article by R.A. Esquivel concerning a height correction factor for Color Computer graphics. I found his .9 correction factor to be valid when operating within the guidelines of PMODE. If you were using the SET command for graphic operation, as Mr. Havens and I did in the above mentioned "Concentric Circles" program, the .9 correction factor no longer remained valid. Under this set of circumstances the correction factor for height is .6. I believe that .6 does not deliver an exact one to one ratio, but even if it is not mathematically close, it is certainly visually close.



```

10 REM CONCENTRIC CIRCLES II
20 REM DONALD E. WHITELOCK 1982
30 F = 0
40 CLS(0)
50 PRINT "ENTER THE RADIUS OF THE CIRCLE."
60 INPUT "----(2 TO 25)---";A
70 IF A < 2 OR A > 25 THEN 50
80 CLS(0)
90 PRINT "ENTER ANY NUMBER GREATER THAN
100 INPUT "20 FOR THE RESOLUTION";B
110 IF B < 20 THEN 90
120 CLS(0)
130 PRINT "ENTER THE NUMBER OF SPACES"
140 PRINT "BETWEEN EACH CIRCLE";C
150 CLS(0)
160 FOR D = 1 TO B STEP .5
170 F = F + 1
180 IF F = 8 THEN LET F = 1
190 I = 30 - (A * SIN(D))
200 J = 16 - (.60 * (A * COS(D)))
210 SET (I,J),F
220 NEXT D
230 A = A - 1

```

```

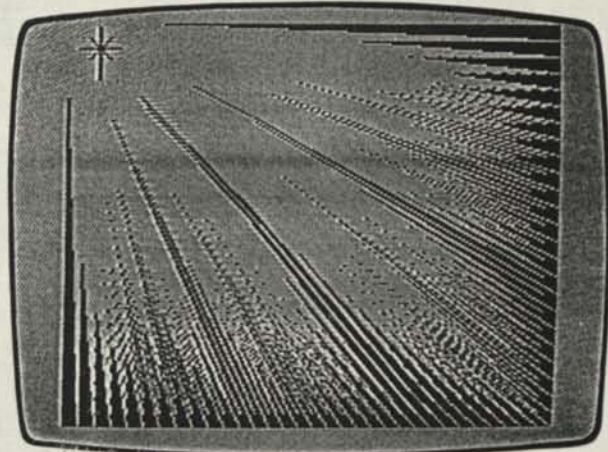
240 IF A < 2 THEN 250 ELSE 160
250 PRINT @ 475, "END"
260 GOTO 260

```

Christmas Card

G. Pigounias
2226 Texas Avenue
Savannah, GA 31404

This program can be used with the Line Printer VII to make a nice Christmas letter to send to friends. After saving the BASIC program on tape, CLOADM the Screen Print Routine (26-3012) using normal procedures, and type POKE 16303,255. Run the BASIC program and break execution when the screen is completed. Then press the <SHIFT> <↑> to begin the screen dump to the printer.



```

10 '*****
20 '***** XMAS CARD *****
30 '*****
40 PMODE 4,1
   : PCLS
   : SCREEN 1,1
50 FOR X = 0 TO 256 STEP 8
60 FOR Y = 0 TO 192 STEP 8
70 LINE (0,0) - (X,Y), PSET
80 NEXT Y
90 NEXT X
100 DRAW "M16,12; C0; N; V10; N; R10; N; D15; N;
   : L10; N; E5; N; G5; N; H5; N; F5;"
110 GOTO 110

```



Radio Shack Computer Center Addresses and Telephone Numbers

ALABAMA
BIRMINGHAM 2428 Green Springs Hwy., (205) 945-0792
HUNTSVILLE 1400 N. Memorial Pkwy., (205) 536-1581
MOBILE 405 Bay Ave., (205) 471-1617
MONTGOMERY #24 Union Square S.C., (205) 271-1500

ARIZONA
PHOENIX 10233 Metro Pkwy. E., (602) 861-1124; 4301 N. 7th Ave., (602) 277-3031
SCOTTSDALE 2525 N. Scottsdale Rd., (602) 990-2241
TEMPE 83 E. Broadway, (602) 894-2065
TUCSON 5622 E. Broadway, (602) 748-0101; Campbell Plaza, 2830 N. Campbell Ave., (602) 861-8003

ARKANSAS
LITTLE ROCK Town & Country S.C. University & Asher, (501) 568-5694

CALIFORNIA
ANAHEIM 509 Katella, (714) 776-9540
BAKERSFIELD 2018 Chester Ave., (805) 323-8074
BERKELEY 1922 Grove St., (415) 848-9170
BEVERLY HILLS 8500 Wilshire Blvd., (213) 659-8870
BREA Imperial Shopping Center, 391 South State College, (714) 871-1681
CANAJOCA PARK 8371 Topanga Canyon, (213) 347-9800
CARMICHAEL 8305 Fair Oaks Blvd., (916) 484-6815
CHICO 1834 Mangrove Ave., (916) 894-0101
CHULA VISTA 1261 3rd Ave., (619) 542-3810
CITRUS HEIGHTS 7405 Grand Back Ln. at San Juan, (916) 723-5183; 969-8752
DOWNEY 6031 Florence Ave., (213) 927-7862
ESCONDIDO 347 W. Mission Ave., (714) 741-6032
FREMONT Fremont Hub, 39114 Fremont, (408) 790-1112
FRESNO Princeton S.C., 2721 N. Blackstone Ave., (209) 225-5551
GARDEN GROVE 12821 Knott St., (714) 891-1160
GLENDALE 236 N. Brand Blvd., (213) 246-9310
HAYWARD 24784 Hesperian Blvd., (415) 887-8420
HOLLYWOOD 8922 Hollywood Blvd., (213) 466-8681
LAKEWOOD 5830 Lakeside Blvd., (213) 920-9671
LA MEZA 5346 Jackson Dr., (714) 460-3610
LONG BEACH 2119 Bellflower Blvd., (213) 597-3377
LOS ANGELES 740 S. Olive St., (213) 629-2455
MONTCLAIR 5237 Arrow Hwy., (714) 825-5430
MONTEREY 484 Washington St., (408) 375-8430
MOUNTAIN VIEW 1833 E. Camino Real W., (415) 961-0542
OAKLAND 1733 Broadway, (415) 763-3183
PASADENA 575 S. Lake Ave., (213) 449-5424
PLEASANT HILL 508 G. Contra Costa Blvd., (415) 676-9666
RIVERSIDE 3844 La Sierra Ave., (714) 889-0340
SACRAMENTO 4749 J. St., (916) 454-2387
SAN BERNARDINO 764 Indiana Center Dr., (714) 884-6871
SAN DIEGO 3062 Clairmont Dr., (714) 276-6050; 3902 E. Canyon Blvd., (619) 280-0227
SAN FRANCISCO One Market Plaza, (415) 777-1910; 2920 Geary Blvd., (415) 221-8716
SAN JOSE 1229 S. Bascom Ave., (408) 297-2603
SAN MATEO 3180 Campus Dr., (415) 573-8607
SANTA ANA 2320 S. Fairview St. at Warner, (714) 966-2787
SANTA BARBARA 4141 State St. A-1, (805) 967-4538
SANTA FE SPRINGS 14138 East Firestone Blvd., (213) 921-0702
SANTA MONICA 511 Wilshire Blvd., (213) 458-1802
SHERMAN OAKS 14636 Ventura Blvd., (213) 778-2639
STOCKTON College Sq. S.C., 963 West March Lane, (209) 957-3676
TARZANA 18545 Ventura Blvd., (213) 343-1696
TORRANCE 3040 Sepulveda at Hawthorne, (213) 373-0306
VENTURA 406 E. Main St., (805) 654-1916
WEST COVINA 2916 E. Workman St., (213) 915-5791

COLORADO
BOULDER Arapahoe Plaza, 3550 Arapahoe, (303) 443-7142
COLORADO SPRINGS 4341 Academy, (303) 553-7500
DENVER 800 E. Quincy, (303) 770-1362; Green Briar Plaza, 7075 Pecon, (303) 428-8787
LAKEWOOD 2099 Wadsworth Blvd., (303) 232-6277

CONNECTICUT
EAST HAVEN 511 Frontage Rd., (203) 467-8864
FAIRFIELD 1196 Kings Hwy. & Rt. 1, (203) 255-6099
HARTFORD The Richardson Bldg., 942 Main St., (203) 549-5717
MARCHETTA 228 Soemen St., (203) 649-8210
NORWALK Rt. 7-345 Main Ave., (203) 846-3418
ORANGE Woodco S.C., 538 Boston Post Rd., (203) 795-1291
WATERBURY 105 Bank St., (203) 873-8800
WATERFORD 122 Boston Post Rd., (203) 443-0716
WEST HARTFORD 39 S. Main St., (203) 523-4293

DELAWARE
DOVER Edgehill Shop. Ctr., Rt. #13, (302) 734-5232
WILMINGTON 3847 Kirkwood Hwy., (302) 999-0193

DISTRICT OF COLUMBIA
WASHINGTON 1800 M St. N.W., (202) 822-3933

FLORIDA
ALTA MONTE SPRINGS 766 B. East Altamonte Dr., (305) 339-1313
BOCA RATON 1662 N. Federal Hwy., (305) 391-3308
CLEARWATER 2460-D US 19 North, (813) 797-3223
DAYTONA BEACH Volusia Plaza, (404) 255-5909
FT. LAUDERDALE 6368 N. Federal Hwy., (305) 672-6797
FORT MYERS Edison Mall, (813) 939-3454
FORT PIERCE Center West Shopping Center, U.S. Hwy. 1, (305) 466-2200
GAINESVILLE 3315 Archer Rd., (904) 375-7063
HOLLYWOOD 429 S. State Rd. #7, (305) 966-4382
JACKSONVILLE 8252 Arlington Expy., (904) 725-2594; Roosevelt Mall, Roosevelt Blvd., (904) 388-6611
LAUDERDALE LAKES 4317-25 N. State Rd. 7, (305) 466-2240
MIAMI 9459 S. Dixie Hwy., (305) 667-2316; 1601 Biscayne Blvd., (305) 374-6433; 15 SE 2nd Ave., (305) 374-7310
MIAMI BEACH The Promenade, 1777 N.E. 163rd St., (305) 540-6887
ORLANDO 1238 E. Colonial Dr., (305) 894-0570
ST. PETERSBURG 3451 66th St. N., (813) 381-2366
SARASOTA 5251 S. Tamiami Tr. Hwy. #11, (813) 923-4721
TALLAHASSEE 2529 S. Adams Pkwy., (904) 222-4440
TAMPA 4555 W. Kennedy, (813) 879-7470; 1825 E. Fowler Ave., (813) 971-1130
W. PALM BEACH 2271-A Palm Beach Lakes Blvd., (305) 683-3100

GEORGIA
AUGUSTA 3435 Wrightsboro Rd., (404) 738-5998
ATLANTA 2108 Henderson Mall, (404) 539-8688; 49 W. Paces Ferry, (404) 231-9604; Akers Mill S.C., 2537 Cobb Parkway NW, (404) 955-5235; 413 Peachtree St., (404) 223-5904
COLLEGE PARK 3339 Old National Hwy., (404) 761-3055
COLUMBUS Columbus Square Mall, 3050 Macon Rd., (404) 568-7200
DORAVILLE 5697 Buford Hwy., (404) 458-2691
SAVANNAH Chatham Plaza, 7805 Abercorn St., (912) 355-6074

IDAHO
BOISE 691 S. Capitol Blvd., (208) 344-5450

ILLINOIS
AURORA 690 North Lake St., (312) 844-2224
CHICAGO 4355 S. Archer Ave., (312) 376-7617; CNA Plaza, 309 S. Wabash, (312) 622-0536; 72 West Adams, (312) 236-7262
ELMWOOD PARK 7212 W. Grand Ave., (312) 452-7500
FAIRVIEW HEIGHTS #4 Market Place, (618) 398-8410
HOMERWOOD/GLENWOOD 329 Glenwood Lansing, (312) 758-0444
LAGRANGE One S. LaGrange Rd., (312) 482-3484
LIBERTYVILLE 1350 S. Milwaukee Ave., (312) 367-8230
LOMBARD 4 Yorktown Center, (312) 629-5350
NILES 8348 Golf Rd., (312) 470-0670
OAK LAWN 4815 W. 95th St., (312) 425-9130
PEORIA 4125 N. Sheridan Rd., (309) 685-7056
ROCKFORD North Town S.C., 3600 N. Main St., (815) 282-1001
SCHAUMBURG 651 Mall Dr., (312) 864-8600
SPRINGFIELD Sherwood Plaza, 2478 Wabash, (217) 787-3066

INDIANA
EVANSVILLE 431 Diamond Ave., (812) 426-1715
FT. WAYNE 747 Northcrest S.C., (219) 482-8547
GRIFITH 208 W. Ridge Rd., (219) 838-3000
INDIANAPOLIS 6242 E. 82nd St., Castleton Plz., (317) 849-6986; Speedway Plaza, 6129 B Crawfordsville, (317) 244-2221; 10013 E. Washington St., (317) 898-4887
TERRE HAUTE 3460 U.S. Hwy. 41 S., (812) 234-3212

IOWA
CEDAR RAPIDS 111 First Ave., S.E. (Downtown), (319) 362-7399
DAVENPORT 616 E. Kimberly Rd., (319) 386-3457
DES MOINES 7660 Hickman Rd., Sherwood Forest S.M., (515) 270-0193

KANSAS
OVERLAND PARK 8619 W. 95th, (913) 642-1301
TOPEKA White Lakes Plaza, West Tower, 3715 Plaza Dr., (316) 257-6420
WICHITA 2732 Blvd. Plaza S.C., (316) 681-1212

KENTUCKY
FLORENCE 7727 Mall Rd., (502) 371-2811
LEXINGTON 2908 Richmond Rd., (606) 289-7321
LOUISVILLE 2000 Taylorville Rd., (502) 458-9901; Louisville Galleria, (502) 589-1013

LOUISIANA
BATON ROUGE 7007 Florida Blvd., (504) 928-5260
MOBILE 2348 W. Park Ave., (504) 873-7713
LAFAYETTE University Square at Congress Blvd., (518) 235-6177
METairie 3750 Veterans Hwy., (504) 454-3681
NEW ORLEANS 327 St. Charles Ave., (504) 523-6408
SHREVEPORT 1545 Line Ave., (318) 221-5125

MAINE
BANGOR Maine Square, (207) 945-6491

MARYLAND
BALTIMORE 7942 Belair Rd., Putty Hill Plaza, (301) 882-9583
BETHESDA 18 Charles St. at Livingston, (301) 539-7251
BETHESDA 7900 Wisconsin Ave., (301) 556-0692
CATONSVILLE One Mile West S/C, 6600 B. Belt, Nat'l. Pk., (301) 788-3277
FREDERICK Shoppers World, Rt. 40W, (301) 695-8440
NEW CARROLLTON-LANHAM 7949 Annapolis Rd., (301) 459-8029
PASADENA 8120 Ritchie Hwy., (301) 544-2352
ROCKVILLE Congressional Plaza, 1673 Rockville Pike, (301) 984-0424
SALISBURY Shoppers World S/C, Rt. 50, (301) 546-9223
TEMPLE HILLS 4520 S. Barnabas Rd., (301) 896-0740
TOWSON-LUTHERVILLE Yorktown S/C York Rd. at Ridgely Rd., (301) 561-2001

MASSACHUSETTS
BOSTON 730 Commonwealth Ave., (617) 739-1704; 111 Summer St., (617) 542-0361
BRAINTREE South Shore Plaza, 250 Granite St., (617) 846-2929
BROCKTON 675 Belmont, (617) 583-2270
BURLINGTON Crossroads Plaza, Rt. 3 S., (617) 229-2650
CAMBRIDGE Harvard Square, 28 Boylston St., (617) 354-7694
CHESTNUT HILL 200 Boylston St., (617) 969-2031
NATICK 1400 Worcester Rd., (617) 875-8721
SAUGUS 243 Broadway, (617) 233-4865
SPRINGFIELD 1985 Main St., Northgate Plz., (413) 732-4745
WORCESTER Lincoln Plaza, (617) 850-8644

MICHIGAN
ANN ARBOR 2515 Jackson Rd., (313) 761-6266
BIRMINGHAM 3620 W. Maple Rd., (313) 647-2151
DETROIT DOWNTOWN 1558 Woodward Ave., (313) 961-8555
FLINT 6308 Main St., (313) 732-2530
GRAND RAPIDS 3142 28th St. SE., (616) 957-2040
KALAMAZOO 25 Kalamazoo Center, (616) 343-0780
LANSING 2519 S. Cedar St., (517) 372-1120
PLAINFIELD North Kent Mall, (616) 364-7558
LIVONIA 33470 W. 7 Mile Rd., (313) 476-6800
ROSEVILLE 31873 Gratiot Ave., (313) 296-6120
SOUTHFIELD 17651 West 12 Mile Rd., (313) 569-1027
TROY Oakland Plaza, 322 John R. Rd., (313) 585-3900

MINNESOTA
BLOOMINGTON 10566 France Ave. S., (612) 884-1641
GOLDEN VALLEY Golden Valley S/C, 8016 Olson Memorial Hwy., (612) 542-8471
ST. PAUL 605 Wabasha, (612) 291-7230

MISSISSIPPI
JACKSON 979 Ellis Ave., (601) 352-5001

MISSOURI
DES-PERES 11960 Manchester Rd., (314) 965-9911
FLORISSANT 47 Florissant Oaks S.C., (314) 921-7722
INDEPENDENCE 1326 S. Holaday Rd., (816) 254-3701
KANSAS CITY 4025 N. Oak Trafficway, (816) 455-3381
ST. ANN 10472 St. Charles Rock Rd., (314) 428-1400
SPRINGFIELD 2684 S. Glenstone, (417) 883-4320

NEBRASKA
OMAHA 3006 Dodge St., (402) 346-4003

NEVADA
LAS VEGAS Commercial Center, 953 E. Sahara #31-B, (702) 371-9956
RENO 3328 Kietzke Lane, (702) 826-6327

NEW HAMPSHIRE
MANCHESTER Hampshire Plaza, 1000 Elm St., (603) 625-4040
NASHUA 429 Amherst St., Rt. 101A, (603) 881-8588

NEW JERSEY
BRIDGEWATER 1472 U.S. Highway 22 East, (201) 469-3232
E. BRUNSWICK 595 A R. 18, (201) 238-7142
E. HANOVER Rt. 10, Hanover Plaza, (201) 884-1200
LAWRENCEVILLE Rt. 1 & Texas Ave., (518) 822-5403
NEWARK 595 Broad, (201) 822-1399
NORTHFIELD 322-24 Tilton Rd., (809) 645-7676
PARAMUS 175 Rt. 17 S., (201) 262-1920
SPRINGFIELD Rte. 422 Center Rd., (201) 487-9827
VORHEES 835 Eagle Plaza, (609) 346-0600

NEW MEXICO
ALBUQUERQUE 2108 San Mateo NE., (505) 255-5687

NEW YORK
ALBANY Shoppers Pk., Wolf Rd., (518) 469-5527
BAYSHORE 1751 Sunnyside Hwy., (516) 666-1800
BETHPAGE 422 N. Wantagh Ave., (516) 822-5403
BROOKLYN 516 86th St., (212) 236-3576
BUFFALO 839 Niagara Falls Blvd., (716) 837-2590
FRESH MEADOWS 187-12 Horace Harding Exp., (212) 454-1075
JOHNSON CITY Giant Shopping Center, Harry L. Drive, (607) 729-8312
NIAGARA KINGS Mall, Rt. 9W, (514) 336-4262
MELVILLE TSS Mall, Rt. 110, (516) 873-4546
NEWBURGH Zayre Plaza, Rt. #17K, (514) 361-2960
NEW ROCHELLE 242 North Ave., (914) 635-0700
NEW YORK 365 5th Ave., (212) 688-1345; 130 E. 42nd St., (212) 683-6053; 19 W. 2nd St., (212) 691-1881; 347 Madison Ave., (212) 867-8650; 270 Park Ave. South, (212) 533-0470
NIAGARA FALLS Pine Plaza, 8351 Niagara Falls Blvd., (716) 283-2941
REGO PARK 87-77 Queens Blvd., (212) 897-5200
ROCHESTER 3000 Winton Rd., (716) 244-5400
SCARSDALE 365 Central Park Ave., (914) 472-2520
SPRING VALLEY White House Center, 86 W. Rt. 59, (914) 425-2828
STATEN ISLAND 2409 Richmond Ave., (212) 698-3100
SYRACUSE 2544 Erie Blvd., (315) 446-3017; Hoop Syracuse, 510 S. Warren St., (315) 471-8663
UTICA Riverside Mall, (315) 735-1933

NORTH CAROLINA
CHARLOTTE 3732 Independence Blvd., (704) 535-6320; Tyvola Mall, 5401 South Blvd., (704) 527-3373
FAYETTEVILLE Eutaw Shopping Center, 815 Elm St., (919) 483-2418
GREENSBORO 3718 High Point Rd., (919) 294-5529
RALEIGH Townsquare, Hwy. 70 W., (919) 871-8380
WINSTON-SALEM 629 Peters Creek Pkwy., (919) 722-0030

OHIO
AKRON Fairview Plaza, 2727 W. Market St., (216) 836-5203
CINCINNATI 5217 Northside Rd., (216) 862-2477
CANTON 5248 Dressler Rd. NW, (216) 494-7230; Melet Plaza, 3825 W. Tuscarawas, (216) 478-1878
CENTERVILLE 2026 Miamisburg-Centerville Rd., (513) 435-5167
CINCINNATI 9725 Montgomery, (513) 783-8688; 16-18 Convention Way, (on Saybrook), (513) 361-4664
CLEVELAND 419 Euclid (Downtown), (216) 575-0800; 27561 Euclid Ave., (216) 289-6823
COLUMBUS 862 S. Hamilton, Great Eastern S/C, (614) 864-2806; The Patio Shop, Ctr., 4861 Karl Rd., (614) 436-4666; 400 N. High St., (614) 484-2781
DAYTON Northwest Plaza, 3279 West Siebenmorgen, (513) 277-6500
FAIRFIELD 7255 Dixie Hwy. (1/4 M. North of I-275), (513) 874-5994
NORTH OLMSTED Great Northern S/C, (216) 734-2255
PARMA 7551 W. Ridgewood Dr., (216) 842-4030
TOLEDO 5844 W. Center Ave., (419) 531-5799
YOUNGSTOWN Union Square Plaza, 2543 Belmont Ave., (216) 744-4541

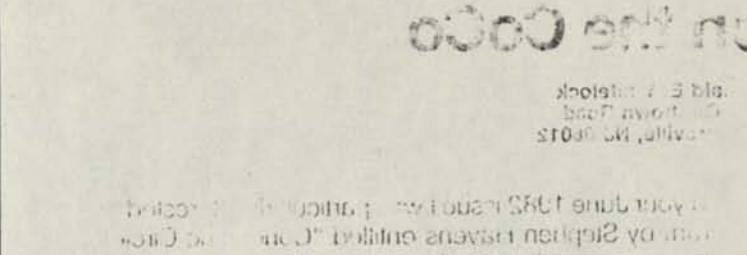
OKLAHOMA
OKLAHOMA CITY 4732 SE 29th St., (405) 670-4561; Springdale S/C, 4469 NW 50th, (405) 943-8712; 1101 SW 59th St., (405) 634-2406
TULSA 7218 & 7220 E. 41st St., (918) 663-2190

OREGON
EUGENE 390 Coburg Rd., (503) 687-0082
PORTLAND 7463 SW Barbur Blvd., (503) 246-1157; 9131 SE Powell, (503) 777-2223
SALEM Salem Plaza, 403 Center, (503) 588-7095

PENNSYLVANIA
ALLENTOWN Crest Plaza S/C, Cedar Crest Blvd. US 22, (215) 935-7155
BALA CYNWYD 67 E. City Line Ave., (215) 666-9950
ERIE 5755 Peach St., (814) 858-5541
HARRISBURG Union Deposit Mall, Union Deposit Rd. #17, (717) 564-6753
LANCASTER Park City Plaza, US 30, (717) 393-5617
MONROEVILLE 3828 Wm. Penn. Hwy., (412) 823-3400
MONTGOMERYVILLE Airport Sq., Rt. 309, (215) 382-1200

Radio Shack

TRS-80 Microcomputer News
P.O. Box 2910
Fort Worth, Texas 76113-2910



ADDRESS CHANGE
 Remove from list
 Change as shown
 Please detach address label and mail to address shown above

NEW JERSEY
BRIDGEWATER 1472 U.S. Highway 22 East, (201) 469-3232
E. BRUNSWICK 595 A R. 18, (201) 238-7142
E. HANOVER Rt. 10, Hanover Plaza, (201) 884-1200
LAWRENCEVILLE Rt. 1 & Texas Ave., (518) 822-5403
NEWARK 595 Broad, (201) 822-1399
NORTHFIELD 322-24 Tilton Rd., (809) 645-7676
PARAMUS 175 Rt. 17 S., (201) 262-1920
SPRINGFIELD Rte. 422 Center Rd., (201) 487-9827
VORHEES 835 Eagle Plaza, (609) 346-0600

NEW MEXICO
ALBUQUERQUE 2108 San Mateo NE., (505) 255-5687

NEW YORK
ALBANY Shoppers Pk., Wolf Rd., (518) 469-5527
BAYSHORE 1751 Sunnyside Hwy., (516) 666-1800
BETHPAGE 422 N. Wantagh Ave., (516) 822-5403
BROOKLYN 516 86th St., (212) 236-3576
BUFFALO 839 Niagara Falls Blvd., (716) 837-2590
FRESH MEADOWS 187-12 Horace Harding Exp., (212) 454-1075
JOHNSON CITY Giant Shopping Center, Harry L. Drive, (607) 729-8312
NIAGARA KINGS Mall, Rt. 9W, (514) 336-4262
MELVILLE TSS Mall, Rt. 110, (516) 873-4546
NEWBURGH Zayre Plaza, Rt. #17K, (514) 361-2960
NEW ROCHELLE 242 North Ave., (914) 635-0700
NEW YORK 365 5th Ave., (212) 688-1345; 130 E. 42nd St., (212) 683-6053; 19 W. 2nd St., (212) 691-1881; 347 Madison Ave., (212) 867-8650; 270 Park Ave. South, (212) 533-0470
NIAGARA FALLS Pine Plaza, 8351 Niagara Falls Blvd., (716) 283-2941
REGO PARK 87-77 Queens Blvd., (212) 897-5200
ROCHESTER 3000 Winton Rd., (716) 244-5400
SCARSDALE 365 Central Park Ave., (914) 472-2520
SPRING VALLEY White House Center, 86 W. Rt. 59, (914) 425-2828
STATEN ISLAND 2409 Richmond Ave., (212) 698-3100
SYRACUSE 2544 Erie Blvd., (315) 446-3017; Hoop Syracuse, 510 S. Warren St., (315) 471-8663
UTICA Riverside Mall, (315) 735-1933

NORTH CAROLINA
CHARLOTTE 3732 Independence Blvd., (704) 535-6320; Tyvola Mall, 5401 S. South Blvd., (704) 527-3373
FAYETTEVILLE Eutaw Shopping Center, 815 Elm St., (919) 483-2418
GREENSBORO 3718 High Point Rd., (919) 294-5529
RALEIGH Townsquare, Hwy. 70 W., (919) 871-8380
WINSTON-SALEM 629 Peters Creek Pkwy., (919) 722-0030

OHIO
AKRON Fairview Plaza, 2727 W. Market St., (216) 836-5203
CINCINNATI 5217 Northside Rd., (216) 862-2477
CANTON 5248 Dressler Rd. NW, (216) 494-7230; Melet Plaza, 3825 W. Tuscarawas, (216) 478-1878
CENTERVILLE 2026 Miamisburg-Centerville Rd., (513) 435-5167
CINCINNATI 9725 Montgomery, (513) 783-8688; 16-18 Convention Way, (on Saybrook), (513) 361-4664
CLEVELAND 419 Euclid (Downtown), (216) 575-0800; 27561 Euclid Ave., (216) 289-6823
COLUMBUS 862 S. Hamilton, Great Eastern S/C, (614) 864-2806; The Patio Shop, Ctr., 4861 Karl Rd., (614) 436-4666; 400 N. High St., (614) 484-2781
DAYTON Northwest Plaza, 3279 West Siebenmorgen, (513) 277-6500
FAIRFIELD 7255 Dixie Hwy. (1/4 M. North of I-275), (513) 874-5994
NORTH OLMSTED Great Northern S/C, (216) 734-2255
PARMA 7551 W. Ridgewood Dr., (216) 842-4030
TOLEDO 5844 W. Center Ave., (419) 531-5799
YOUNGSTOWN Union Square Plaza, 2543 Belmont Ave., (216) 744-4541

OKLAHOMA
OKLAHOMA CITY 4732 SE 29th St., (405) 670-4561; Springdale S/C, 4469 NW 50th, (405) 943-8712; 1101 SW 59th St., (405) 634-2406
TULSA 7218 & 7220 E. 41st St., (918) 663-2190

OREGON
EUGENE 390 Coburg Rd., (503) 687-0082
PORTLAND 7463 SW Barbur Blvd., (503) 246-1157; 9131 SE Powell, (503) 777-2223
SALEM Salem Plaza, 403 Center, (503) 588-7095

PENNSYLVANIA
ALLENTOWN Crest Plaza S/C, Cedar Crest Blvd. US 22, (215) 935-7155
BALA CYNWYD 67 E. City Line Ave., (215) 666-9950
ERIE 5755 Peach St., (814) 858-5541
HARRISBURG Union Deposit Mall, Union Deposit Rd. #17, (717) 564-6753
LANCASTER Park City Plaza, US 30, (717) 393-5617
MONROEVILLE 3828 Wm. Penn. Hwy., (412) 823-3400
MONTGOMERYVILLE Airport Sq., Rt. 309, (215) 382-1200