Recreational COMPUTING

FOR THE IMAGINATIVE SMALL COMPUTER USER!
VOL ISSUE 47 SEPT-OCT 1980

Recreational COMPUTING
Sept/Oct 1980
Volume 9, Issue 2

Editor's Notes

It might not seem so now, but the People's Computer Company was a pretty daring adventure when it was founded eight years ago. "Bring computers to the people," indeed! Computers, then, were large and intimidating machines that took up whole rooms and cost a fortune. How could you bring something like that to the people? Where would you put it—in your living room? The whole idea of ordinary people using computers seemed like an impossible dream at the time.

An impossible dream. But there was the People's Computer Company, publishing their newspaper, bringing tidbits of information to all those people who thought the dream was not so impossible. As a celebration of that long-ago vision, we are including in our Eighth Anniversary Issue a few pages from our upcoming book, "The Best of The People's Computer Company." Those of you who were around then will read these pages a little wistfully; the rest of you, we hope, will enjoy them for their somewhat madcap but always informative stance.

This issue of *Recreational Computing* is somewhat of a transitional issue. We are celebrating the past with the tribute to the early years of PCC, but we want also to look forward to our future. Starting with the next issue, *RC* will have a new editor. She is Joan Hiraki, a Journalism graduate of The University of California at Berkeley. Joan has a technical editing and writing background, having worked as a Publications Manager for Gould Inc., and a Technical Editor/Writer for Lawrence Livermore Laboratory in Berkeley. Joan brings a lot of enthusiasm and plenty of ideas to her new job, and she will definitely want to hear from you in the weeks and months ahead. Let her know what you think is good about this magazine, what could stand improvement, what it lacks. With your help *RC* will be better than ever.

We hope you enjoy this issue as much as we enjoyed putting it together for you. —SR

ADVERTISING

ADVERTISING space is available in this publication. Please direct inquiries to the Advertising Director, People's Computer Company, Box E, Menlo Park, CA 94025. (415) 323-3111.

SUBSCRIPTION INFORMATION

U.S. RATES

☐$12/1 yr. (6 issues)
☐Retaining subscription @ $25 ($13 tax deductible)
☐Sustaining subscription @$100+ ($88 + tax deductible)
☐Lifetime subscription @ $900+ ($700 tax deductible)

Please allow 6-9 weeks for your first issue to arrive.

FOREIGN RATES

Payments must be in $US drawn on a US bank.
☐$20 Canada First Class
☐$26 Rest of World Airmail
☐$16 World Surface Mail

Delivery of foreign mail is slow and unreliable. We strongly advise airmail.

BACK ISSUES

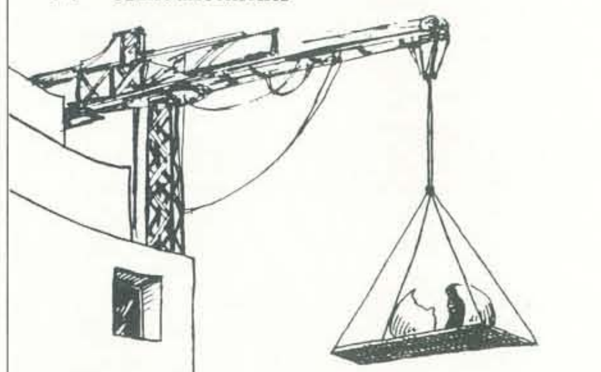
$2.50 each:
Vol 7, Nos. 1, 2; Vol. 6, Nos. 1, 2, 3, 4, 5
Outside the U.S. add $.50 per issue.

$3.00 each:
Vol. 7, Nos. 3, 4, 5, 6
Vol. 8, No. 1, 2, 3, 4, 5

Cover by Fred Carpenter

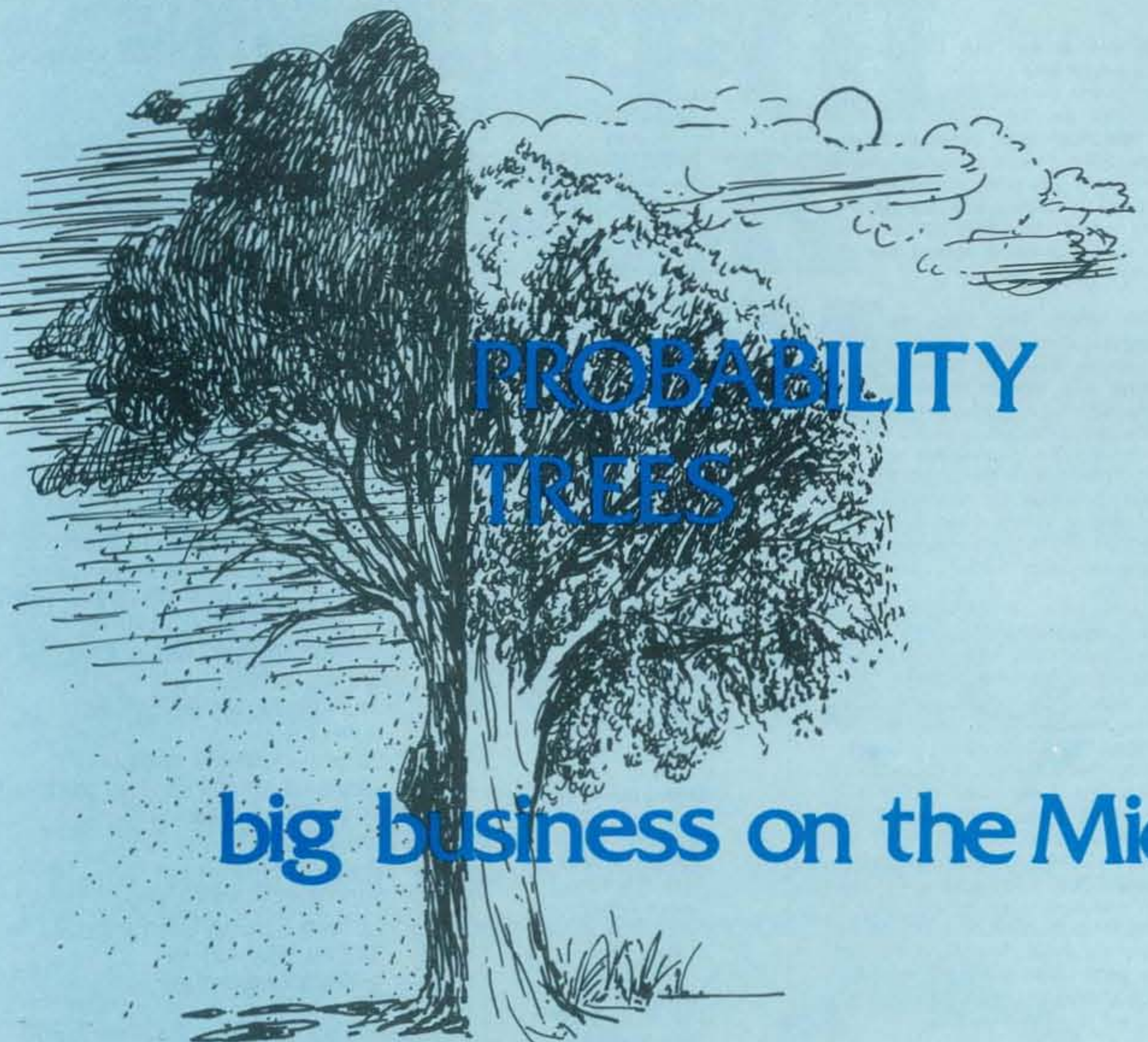
8TH ANNIVERSARY ISSUE

Departments



Recreational Computing (ISSN #0164-5846) is published bimonthly by People's Computer Company, 1263 El Camino Real, Box E, Menlo Park, CA 94025. People's Computer Company is a tax-exempt, non-profit corporation. Donations are tax-deductible. Second class postage paid at Menlo Park, California, and additional entry points. Address correction requested. Postmaster: send form 3579 to Box E, Menlo Park, CA 94025.

FOREIGN DISTRIBUTORS OF *RECREATIONAL COMPUTING*: UK & EUROPE: L P Enterprises, 8/11 Cambridge House, Cambridge Road, Barking, Essex, 1G11 8NT, Great Britain. Hofacker-Verlag, Tegernseer Strasse 18, D-8150 Holzkirchen, West Germany. Computerland/Computer Store AB, Box 7134, Kungsgatan 19, S-10387 Stockholm, Sweden. **CANADA:** RS-232, 186 Queen Street W., Toronto, Ontario M5V 1Z1, Canada. **ASIA & AUSTRALIA:** Electronic Concepts Pty Ltd., 55 Clarence Street, Sydney, NSW 2000, Australia. Computer Store, POB 31-261, 22B Milford Road, Milford, Auckland 9, New Zealand. ASCII Publishing, 305 Hi Torio, 5-6-7 Minami Aoyama, Minato-ku, Tokyo 107, Japan. **AFRICA:** Computers, Etc., 170 Hendrik Verwoerd Drive, POB 51978, Randburg 2125, Republic of South Africa.



PROBABILITY TREES

big business on the Micro

By Ramon Zamora

In the world of business planning, large computers are often used to generate scenarios for alternative business plans. One way to do this is to "grow" probability trees, attach a model to the endpoints of each path through the tree, run the model based on the path being traveled, then aggregate the model results in the form of an outcome probability distribution.

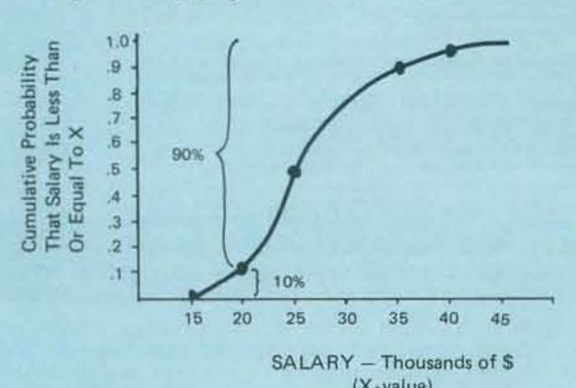
This article briefly describes how to generate probability trees on your micro. Let us hear from you on any applications you devise using this big business technique on your small computer. —*RZ*

WHAT IS A PROBABILITY TREE?

An event can be described in terms of the probabilities of the event happening. For example, if I ask a friend what his or her salary will be five years from now, and probe the answers I get, I can end up with statements like these concerning the future event:

- There is a 90% chance the salary will be greater than $20,000.
- There is a 10% chance the salary will be greater than $35,000.
- There is a 50-50 chance the salary will be $25,000.
- There is a 5% chance the salary will be greater than $40,000.
- There is no chance the salary will be less than $15,000.

A way to represent this information concisely is to use a cumulative probability graph:



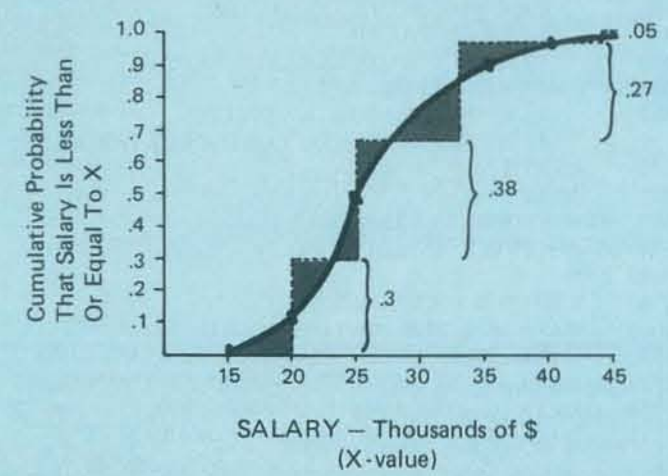
Cumulative Probability Distribution on Salary

The graph captures the information of the verbal statements and gives a "picture" of what the future salary conditions for this person might be. Each verbal statement can be phrased in two ways:

- There is a 90% chance that the salary will be greater than $20,000, or
- There is a 10% chance that the salary will be less than or equal to $20,000.

Look at the cumulative probability graph and determine for yourself that these two statements are equivalent. What are the equivalent statements for some of the other salary probability comments?

A way to use the probability graph in computations is to take many points from the graph and input them into the computer. The array of points can then be *sampled* to provide data for models. Another technique is to make a discrete approximation of the continuous curve in the following manner:

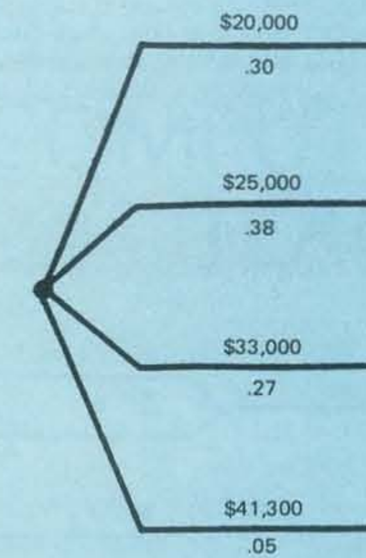


Discrete Approximations to Continuous Curve

The graph has been reorganized into four discrete *lumps* of probability. The resulting approximation, if done carefully, preserves the information in the curve and reduces the number of data items to be fed to the computer. The number of *lumps* is determined by the overall problem and by the importance of the variable being approximated. A general rule of thumb is that the more important the variable, the more *lumps* needed. A variable is considered to be *important* if changes in the variable significantly affect the outcomes or results of the problem being analyzed.

The approximation technique just used involves creating small triangular shaped sections of nearly equal area for each *lump*. (See the shaded areas on the last figure.) The human eye is quite good at making these approximations directly. Of course, a computer program can probably be created that will do the same thing, but try it with your eye first.

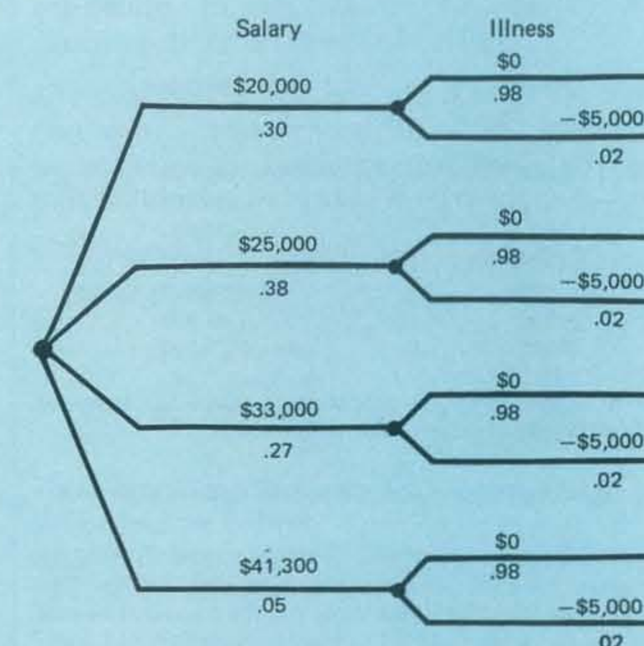
When a continuous graph has been broken into *lumps*, it can be represented in an alternate graphical format:



Discrete Version of SALARY Distribution

This form of the information is often referred to as a *node* or *probability node*. Each node has *branches* that correspond to the number of *lumps* taken from the original curve. The branches show the probability and values that go with each *lump*.

When several of these probability nodes are combined into a single structure, the result is a probability tree.



Simple Probability Tree

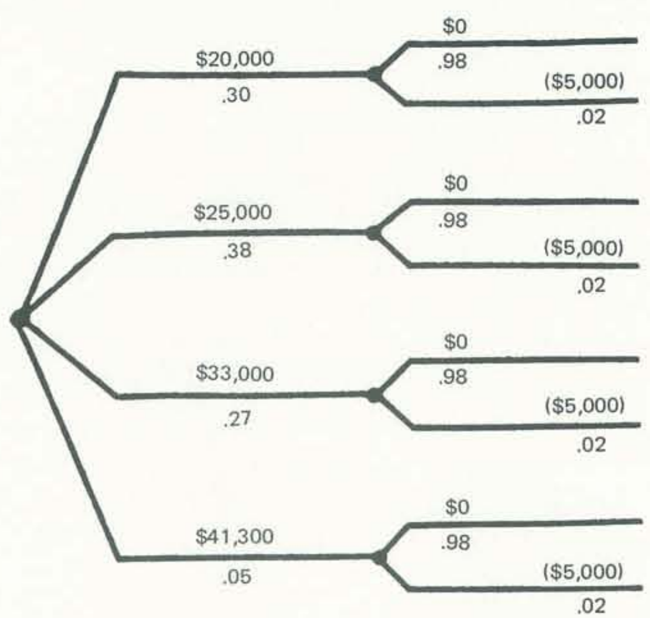
The first node in this example is the discrete distribution on salary. The second node might represent the probability of a serious illness (.02), with a corresponding loss of $5,000. There are eight outcomes or endpoints for this structure—eight distinct salary and loss combinations, and eight probabilities, one for each outcome.

SOLVING A PROBABILITY TREE

Solving a probability tree involves a process of *rolling forward* through the tree structure (traveling along each path to an endpoint), and multiplying probabilities together while summing the branch values. For example, in the salary/illness tree, *rolling forward* along the topmost branch yields:

Probability path = .3 x .98 = .294
Net Salary = 20000 + (0) = 20000 dollars

Here are the results of *rolling forward* along each path of the salary/illness tree:



Prob	Net Salary
.294	$20,000
.006	15,000
.3724	25,000
.0076	20,000
.2646	3,000
.0054	28,000
.049	41,300
.001	36,300

The sum of the endpoint probabilities times the endpoint rewards yields the *expected value* of the solution:

$\Sigma p_i v_i$ = expected value = $26,375 for the salary/illness tree

This set of hand computations is simple enough for the example given, but the solution is not so simple if the trees get larger. The question arises: how can the computer be used to generate and solve this kind of tree structure?

LET THE MICRO SOLVE IT

There are many ways to generate tree structures. Probability trees and their associated computations form a special class of tree structure that can be generated and solved directly with nested FOR-NEXT loops. The following program generates the salary/illness tree, develops and displays the computations for each path, and computes the expected value of the outcomes.

Program

```
100  REM*** PROB TREE GENERATION
110  REM*** GET NUMBER OF BRANCHES
120  READ LS, LI
130  REM*** GET SALARIES S(I) AND PROBABILITIES PS(I)
140  READ S(1), S(2), S(3), S(4)
150  READ PS(1), PS(2), PS(3), PS(4)
160  REM*** GET ILLNESS COSTS AND PROBABILITIES
170  READ ILL(1), ILL(2)
180  READ PILL(1), PILL(2)
190  REM*** SET EXPECTED VALUE TO ZERO; CLEAR
     SCREEN
200  EV = 0: CLS
210  PRINT TAB(2)"I" TAB(6)"J" TAB(11)"PROB" TAB(21)
     "NET SALARY"
220  PRINT "--- --- ------- ----------"
230  REM*** GENERATE TREE PATHS
240     FOR I = 1 TO LS
250     FOR J = 1 TO LI
260        REM*** COMPUTE PATH PROBABILITY
270        PROB = PS(I) * PILL(J)
280        REM*** COMPUTE NET SALARY
290        NET = S(I) + ILL (J)
300        REM*** COMPUTE EXPECTED VALUE
310        EV = EV + PROB * NET
320        REM*** DISPLAY PATH RESULTS
330        PRINT TAB(1) I TAB(5) J TAB(10) PROB TAB(20) NET
340     NEXT J
350     NEXT I
360  REM*** DISPLAY EXPECTED VALUE
370  PRINT: PRINT "EXPECTED VALUE = "; EV
380  END
1000    REM*** DATA VALUES
1010    DATA  4, 2 :REM  LS, LI NUMBER OF BRANCHES
1020    DATA  20000, 25000, 33000, 41300 :REM  SALARIES
1030    DATA  .3, .38, .27, .05 :REM  SALARY PROBABILITIES
1040    DATA  0, -5000 :REM  ILLNESS COSTS
1050    DATA  .98, .02 :REM  ILLNESS PROBABILITIES
```

When the program is entered and RUN, the screen will show the same results that were just discussed for solving the salary/illness probability tree by hand. The program is written in Level II BASIC for the TRS-80.

For tree structures with more nodes, the program can be altered by adding more FOR-NEXT loops for each node in the tree, and setting up the additional data elements. If this approach is used on larger problems, a few key items must be considered.

SIZE AND DEPENDENCIES

The size of the tree (the number of paths) grows quickly as the number of nodes is increased. An eight node tree where each node has three branches will have 3^8, or 6561 paths. Adding one more node of three branches to this tree gives a total of 19,683 paths! This fact underscores the idea of including only important variables in the tree. Otherwise, the computation time can become prohibitive.

The other item to be considered is the relationships that might exist between one node and the next. In the salary/illness tree, the probability of illness was assumed to be *independent* of the salary figure; the probability of health or illness did not vary as the salary changed. A case could be constructed where the opposite might be postulated: as the salary increases, the probability of illness also increases due to extra stress. In the latter case, the probabilities in the second node would be *dependent* on the branch being traveled in the first node. The data might look like this:

Salary	*Probability of Illness*
$20,000	.02
25,000	.04
33,000	.10
41,300	.20

The example program would have to be altered so that the probability of illness (PILL) array would be a 4 x 2 matrix, PILL(4,2), and the probability calculation changed to:

PROB = PS(I) * PILL(I,J)

SUMMARY

This article has discussed, in brief, a method for generating and solving probability trees using FOR-NEXT loops. The technique provides a way to systematically produce the paths along a multi-stage tree-like structure, and associate data with the path being traveled. The program is particularly applicable to probability trees with both independent and dependent data elements.

Graphics

Games & the Arts

Computers & Learning

Creativity & Programming

Sports & Medicine

Each and every issue of **Recreational Computing** is replete with articles you can work with. **Recreational Computing** offers you active, live ideas to get the most out of your computer.

Subscribe now and find out that between the lines is a whole new experience.

Recreational COMPUTING
Box E, Menlo Park, CA 94025

I want to subscribe to **Recreational Computing!** Please bill me $12 for 6 issues.

Name______________________________

Address____________________________

City________________________________

State______________ Zip______________

K7

Q. Your students are gathering around the several PET computers in your classroom. And they all are hungry for hands-on turns at the keyboards. Some students are just beginning to understand computers; others are so advanced they can help you clean up the programs at the end of the period. How do you set up a job queue, how do you keep the beginners from crashing a program, how do you let the advanced students have full access? And how do you preserve your sanity while all this is going on?

A. With the Regent.

Q. What is the Regent?

A. The ultimate in classroom multiple PET systems. A surprisingly inexpensive, simple, effective way to have students at all levels of computer capability work and learn on a system with up to 15 PETs while the instructor has complete control and receives individual progress reports.

Up to 15 PETs, one dual disk drive and as many as five printers can interface with the Regent, and do all those good things we promised. It's designed to operate with 8K, 16K, 32K PET/CBM models and with the Commodore disk drives and new DOS.

Five levels of user privilege, from the Systems Level, through Levels One and Two, Student; Levels One and Two, Operator. From only the use of system commands

and to the disk and bus. This situation can, of course be corrected partially by the Proctor, completely by the

s inadvertant disruption when one loading and another is being used.

harge of the bus and resolves cts. Each student can load down but cannot inadvertently load to or Good for computer aided instruction ications, offering hundreds of ing computer users.

ardware and software protects the ed erasures and settles IEEE bus y the instructor or a delegate can he disk. Yet all the PETs in the to all disk programs. Available for s. SUB-it and PET intercontrol own-loading software) are included.

nsive are these classroom

word is **inexpensive**. The **Regent** 0 for the first PET; **$150** for each e system. The **SUB-it** is **$40**. (Add t **$22.50** if the PET is an original **octor** is $95.

ilable, too: 1 meter at $40 each; 4 meter, $90 each.

to answer any questions and to send you the complete information package.

Skyles Electric Works
231 E South Whisman Road, Mountain View, California 94041 (415) 965-1735

Dr. Dobb's Journal
of Computer Calisthenics & Orthodontia

12 ISSUES Just $21

Name ____________________ Please Print

Address ____________________

City, State, Zip ____________________

Signature __________ Offer good in USA only. Foreign Rates upon Request.

K3

(Save $9 off newsstand price)

Send No Money, Mail Today!

Each and
Computing i
can work wi
offers you ac
most out of y
Subscribe
between the
experience.

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 756 MENLO PARK, CA

POSTAGE WILL BE PAID BY ADDRESSEE

Dr. Dobb's Journal
of Computer Calisthenics & Orthodontia

SUBSCRIPTION SERVICE
P.O. BOX E
MENLO PARK, CA 94025

I want
Comp...

Name__________

Address__________

City__________

State__________ Zip__________ K7

Q. **Your students are gathering around the several PET computers in your classroom. And they all are hungry for hands-on turns at the keyboards. Some students are just beginning to understand computers; others are so advanced they can help you clean up the programs at the end of the period. How do you set up a job queue, how do you keep the beginners from crashing a program, how do you let the advanced students have full access? And how do you preserve your sanity while all this is going on?**

A. With the Regent.

Q. What is the Regent?

A. The ultimate in classroom multiple PET systems. A surprisingly inexpensive, simple, effective way to have students at all levels of computer capability work and learn on a system with up to 15 PETs while the instructor has complete control and receives individual progress reports.

Up to 15 PETs, one dual disk drive and as many as five printers can interface with the Regent, and do all those good things we promised. It's designed to operate with 8K, 16K, 32K PET/CBM models and with the Commodore disk drives and new DOS.

Five levels of user privilege, from the Systems Level, through Levels One and Two, Student; Levels One and Two, Operator. From only the use of system commands to complete control for the exclusive use of the instructor.

There's complete system protection against the novice user crashing the program; the instructor has total control over, and receives reports concerning, usage of all PETs.

A complete set of explanations for all user commands is stored on the disk for instant access by all users. And a printout of the record of all usage of Regent is available at the instructor's command.

The Regent includes a systems disk with 100,000-plus bytes for program storage, a ROM program module, together with a Proctor and a SUB-it . . . and complete instructor and student user manuals.

Q. SUB-it? Proctor? What are they?

A. **The SUB-it** is a single ROM chip (on an interface board in the case of the original 2001-8 models) that allows up to 15 PETs to be connected to a common disk via the standard PET-IEEE cables. The Commodore 2040, 2050 or 8050 dual disks and a printer may be used.

(The SUB-it has no system software or hardware to supervise access to the IEEE bus. The system is thus unprotected from user-created problems. Any user—even a rank novice—has full access to all commands and to the disk and bus. This situation can, of course be corrected partially by the Proctor, completely by the Regent.)

The SUB-it prevents inadvertant disruption when one unit in a system is loading and another is being used.

The Proctor takes charge of the bus and resolves multiple user conflicts. Each student can load down from the same disk but cannot inadvertently load to or wipe out the disk. Good for computer aided instruction and for library applications, offering hundreds of programs to beginning computer users.

A combination of hardware and software protects the disk from unexpected erasures and settles IEEE bus usage conflicts. Only the instructor or a delegate can send programs to the disk. Yet all the PETs in the system have access to all disk programs. Available for all PET/CBM models. SUB-it and PET intercontrol module and DLW (down-loading software) are included.

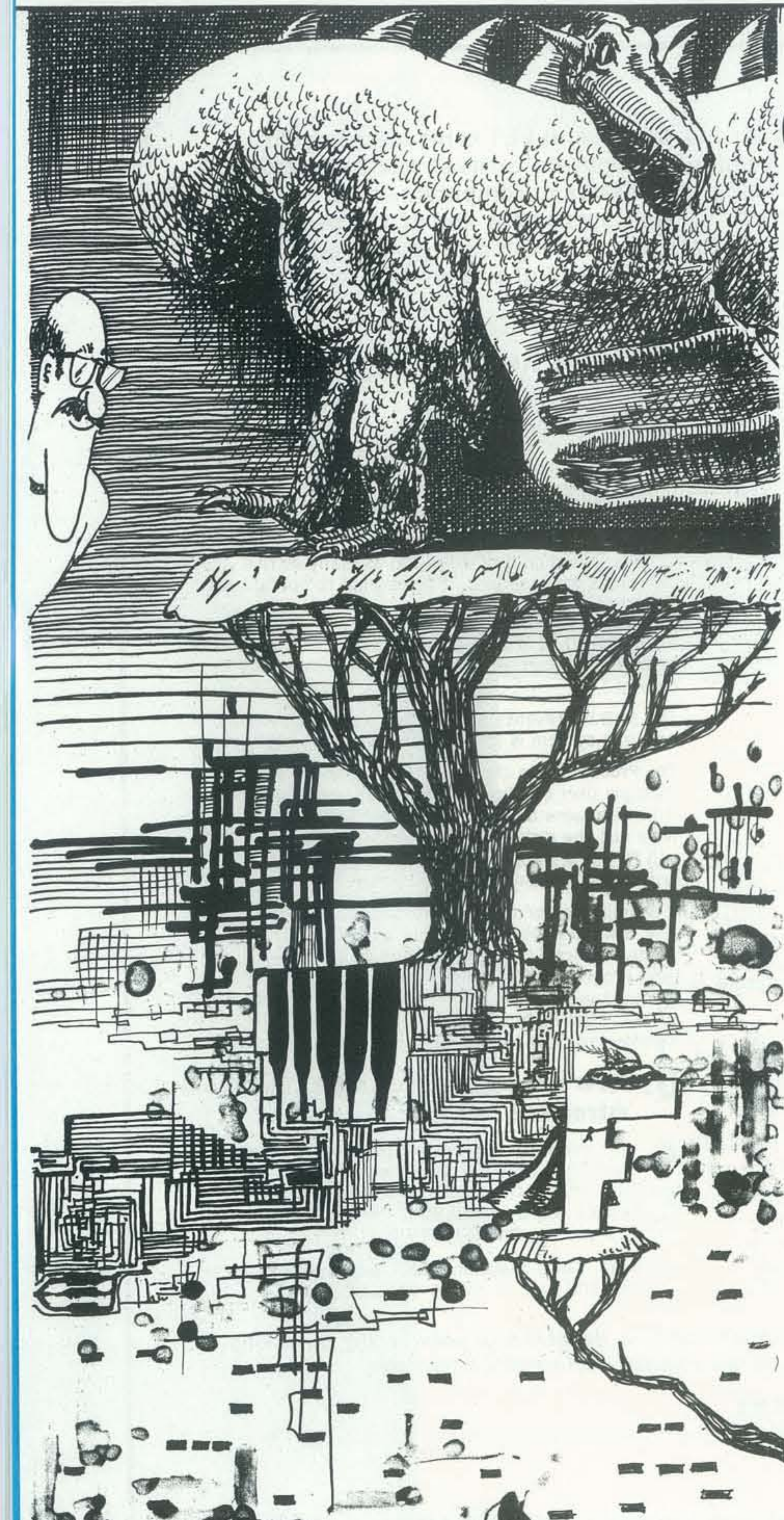
Q. How expensive are these classroom miracles?

A. We think the word is **inexpensive**. The **Regent** system is **$250** for the first PET; **$150** for each additional PET in the system. The **SUB-it** is **$40**. (Add an interface board at **$22.50** if the PET is an original 2001-8.) And the **Proctor** is $95.

There are cables available, too: 1 meter at $40 each; 2 meter, $60 each; 4 meter, $90 each.

Phone or write for information. We'll be delighted to answer any questions and to send you the complete information package.

Skyles Electric Works
231 E South Whisman Road, Mountain View, California 94041 (415) 965-1735



Computers Are Mostly Used Against People

From Volume I, No. 1
People's Computer Company Newspaper
October 1972

Way back in 1972, in those far away days before there really was such a thing as a home computer, a few farsighted individuals decided to bring computers to the people. The best way they could find to do this was through a small store, the People's Computer Center, where people could come in, pay $3 and have access to a computer for an hour. It was a kind of very early and egalitarian timesharing system. Eventually, the group, which was led by Bob Albrecht and included Jerry Brown, Le Roy Finkel, Mary Jo Bajada, Keith and Lois Britton, Joanne Verplank and Dennis Allison, decided to start a newspaper.

Instead Of For People ~ Used To Control Instead Of To FREE Them ~ Time To Change All That ~ We Need A PEOPLE'S COMPUTER COMPANY!

It was eight years ago this October that the first issue of the People's Computer Company newspaper rolled off the press. Wild, wacky, chaotic, absurd, infuriating, hilarious, chock full of ideas, riddled with dragons, drowned in excesses of beer suds and strong Peet's coffee, and never, never aloof: this was the PCC newspaper.

In eight years a lot of things have happened. What had seemed like a fuzzy idea—computers for the people—is now a reality. The first real home computer, the Altair 8800, made the scene in 1975 with a price tage of under $400. Accessible to many, but not to all: the Altair was a kit. A couple of years later, Commodore introduced its PET, the first standalone, inexpensive home computer. The market opened up immediately and the rest is history.

We here at the People's Computer Company are proud to have been a vital part of that history. We were here at the beginning. The PCC newspaper is no longer a newspaper; for some years it has been a magazine, and a year ago we changed our name to *Recreational Computing*. Commencing with the November/December issue, *RC* will have a new editor, Joan Hiraki, a talented woman who is brimming with ideas and plans for the future.

And so, on our eighth anniversary, let's pause for a brief look back at some of the highlights. On the next few pages we present excerpts from PCC's upcoming book, "The Best of the People's Computer Company," which will be available this winter. You can see the gamut that's been run in our pages: everything. We've never had a hard and slick format, and the newspaper/magazine has just kept changing with the times. We hope you enjoy this look back.

Thank you, one and all for reading us, and we hope you continue to do so for years to come. Here's to the future.

~ PEOPLE'S COMPUTER CENTER ~

People's Computer Center is a funky amusement place – a place to have fun with computers. Never touched a computer? What will it do? Well, come on down and find out. We believe COMPUTERS ARE FOR PEOPLE TO PLAY WITH. Think of them as learning tools or toys for young and old. A computer is a general purpose game-box that abides by the rules you give it. We at PCC make up new games and play them all the time. You can too! What do you like? We've got lots of computer games. All sorts: Star Trek, cave exploration, logic puzzles, number guessing, whatever. A computer and your imagination can take you to a fantasy land where you can boldly match your mental skills against seemingly intelligent beings and all manner of obstacles. You may encounter strange behaving creatures which you come to accept once you discover their pattern. It is all harmless, but the challenge is there and the head stimulation can keep you on edge for hours.

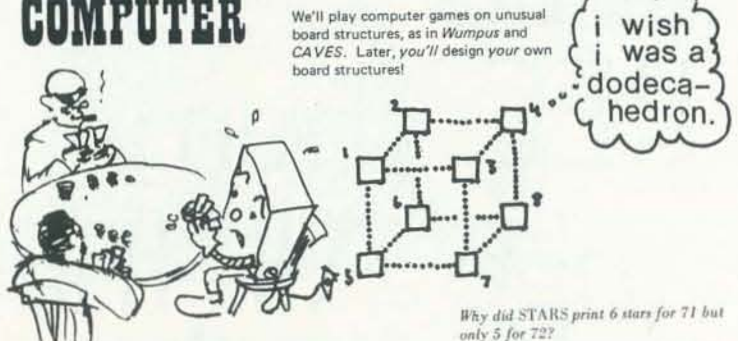
BEAT THE COMPUTER

Gamesters: Come play computer games. Learn how they work. We can arrange times and number of meetings.

WELCOME TO THE CAVES, WUMPUS LOVERS!!

Have you ever played *Hunt the Wumpus*? Wumpi live underground, slinking around the caves and caverns. But watch out for bottomless pits and superbats! Or, you can use your Wumpi sensors to track friendly Wumpi.

We'll play computer games on unusual board structures, as in *Wumpus* and *CAVES*. Later, *you'll* design *your* own board structures!



Write computer games to amaze your friends!! Or solve useless scientific equations in five micro-seconds or less!!

Looking for an interesting place to visit Friday night?

Between 7 and 10 pm, our doors are open and for only one dollar (half if you're in 8th grade or younger) you can play

Games! Games! There are board games, there are computer games – sometimes we even have people games.

FRI
GAMES NIGHT: 7—10

COMPUTER GAMES
- STAR TREK
- MUGWUMP
- HURKLE
- CAVES
- STARS
- BAGELS
- STAR TRADER
- REVERSE

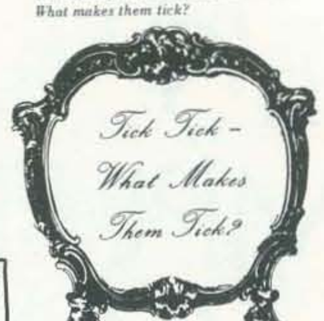
BOARD GAMES
- GO
- GRAVITY TRAP
- CHESS
- TAU
- CONFRONTATION
- SMESS
- CUBIC
- IMPASSE

Why did STARS *print 6 stars for 71 but only 5 for 72?*

Why can't HAMURABI *plant all of his land?*

Why did my LUNAR *module crash and blast a new crater 51.3 miles in diameter?*

REVERSE, TAXMAN, *and lot's more – What makes them tick?*



Maybe HAMURABI *should be democratically controlled or how about* CHOMP*ing a donut, instead of a cookie, or suppose the object of* STARS *is to get 4 stars and if you guess the number you lose, or . . .*

Contact the TICKMAN now at 323-6117

Hours to be arranged

Each ticket entitles you to:

- 4 two-hour instruction classes with experienced PCC staff on any Tuesday, 4-6 or 7:30-9:30 pm.
- 4 one-hour computer terminal times by arrangement

Learn at your own pace. For example, with a ticket you can come once a week, once every other week, or twice on the same Tuesday. Terminal time can be flexible. You don't ever miss a class because the ticket is good next week (for a two month period).

Tickets: $20 each.

Basic and Up

How about reserving the Center for a party?

We have lots of games for kids of all ages.

Afternoon, evening, or night – check if the Center is available.

potluck dinner

Bring your favorite dish, a bottle of wine, or a sixpack of coke and join us for our weekly get together.

P.S. – If you arrive too late, the food may be gone.

Wednesdays 6:00 - 8:00

Computer Hardware Class

Wednesdays 7:30 - 9:00

An informal, *free* class about the innards of computers. We may even build some stuff.

ReNt TiMe

Terminal time cost per hour grows bigger as you grow bigger.

$1/HOUR up through grade 8

$2/HOUR High School

$3/HOUR Beyond HS

We have two computerized game versions of Star Trek plus a big, instellar game called Star Trader.

If you like science fiction, then this class is for you! Our computers can write their own (original – of course) sci fi stories. Would you like to learn how?

We've also got a growing sci fi library, and we'll be talking about *designing* computer space games . . .

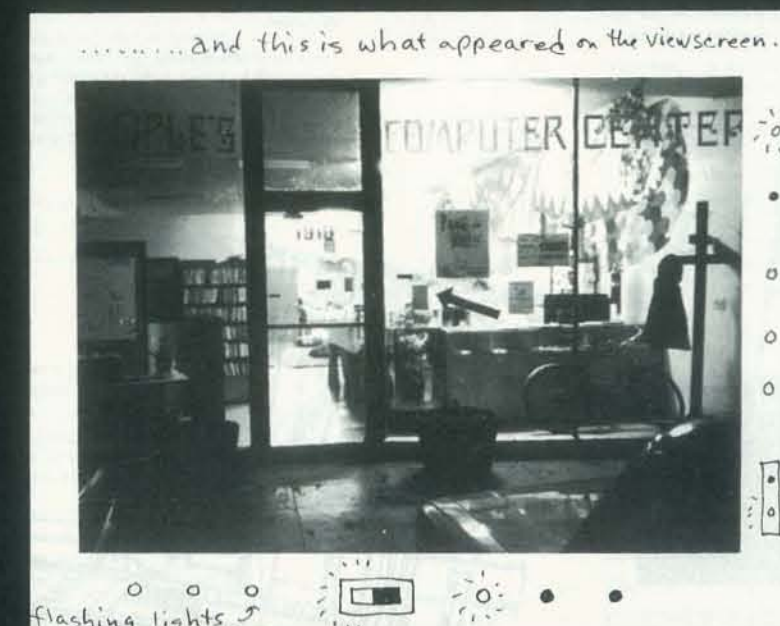
FLY YOUR OWN SPACESHIP!

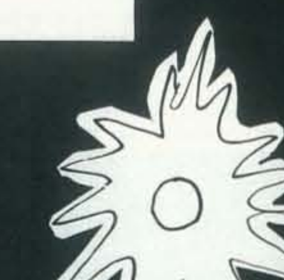
Hours to be arranged

CLOSED SUN

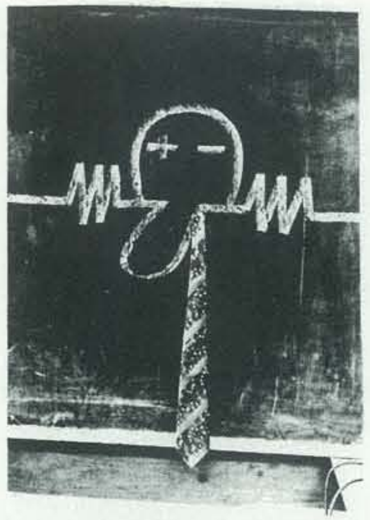
The Center is located at 1919 Menalto Avenue (on the corner of Gilbert Avenue) in Menlo Park and our phone number is 323-6117. We are open Monday through Saturday (closed on Sunday) – usually by 10AM (earlier if you want to schedule a class then) and close around 6PM. Our late evening hours vary. Are you a newcomer? Drop by Friday night, our Game Nite, or Saturday during the day. No previous experience necessary. Just come on in . . .

suddenly the ship jolted. the image on the screen went blank. "scott, what's happened to the viewscreen?" "i don't know sir. it just died." "sir," said spock, looking into the scanner, "it would appear that we have somehow transmutated into 2-dimensional space. specifically, we are appearing on a newspaper page for the entertainment of an inferior species of the human race called 'programmers'." "how embarassing" said the captain. "how do things like this *happen?*" "sir," said scott: " somethings appearing on the viewscreen............"





"peoples computer center..." said kirk mysteriously. "we can all read" replied spock sarcastically. "but i thought it was peoples computer *company*," said kirk. "no, dummy, thats the name of the newspaper we're in. this is the place they used to live next door to: the *community computer center*." "then why does it say 'peoples computer center' on the window?" "well, its very simple........." replied spock. he peered into the scanner for a few minutes. "uh, yes–" he said: "you see, they *were* going to be called the peoples computer center but when they applied for nonprofit status the state of california tc they couldn't have that name because it was too close to the name of an existing company - namely the peoples computer company - but the state didn't know that that was the parent organization and the center didn't know that that was the reason they couldn't have that name and that they really *could* have had that name if they wanted but by the time it was straightened out the signpainter had left town and they couldn't find anybody to repaint it." and with that, the text on the first page of the article ended.

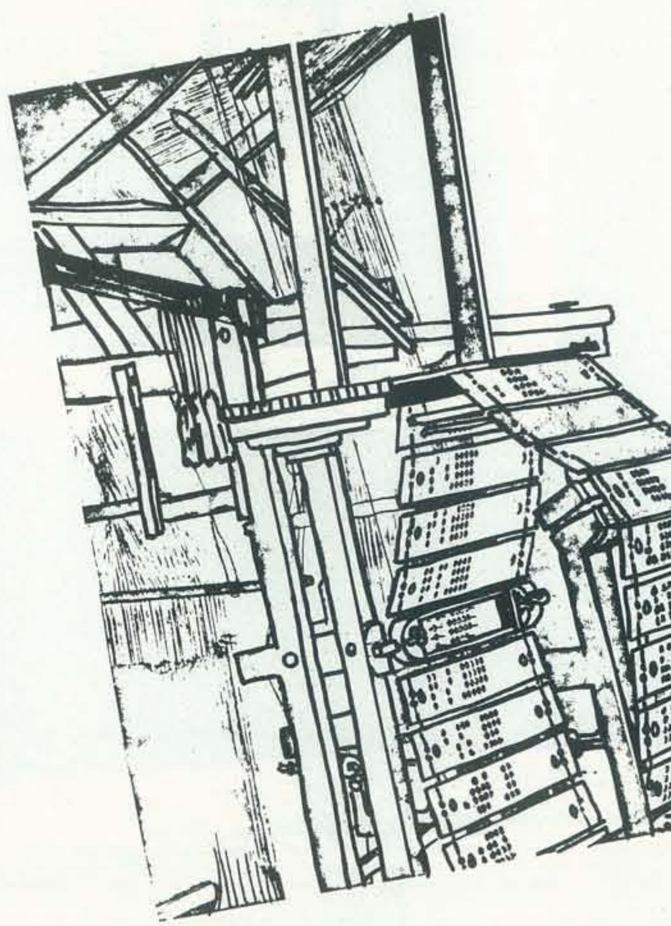


The Computer & the Weaver

Regrettably, the name and company of the Japanese artist that produces this computer generated weaving, was lost in the shuffle.

A French weaver by the name of Joseph Jacquard invented a very sophisticated type of loom that first arrived in America about 1820 and was operated by hand with the flying shuttle. The Jacquard attachment could be added to looms already in use for Double Weave coverlets, and thus the mechanization of weaving had begun. Jacquard's invention consisted of a series of cards with large and small punched holes that activated the harnesses of the loom (as many as 40 at a time) and made the pattern. Weavers became very proficient and could "punch" their cards so as to satisfy the design whims of their customers.

Some information about the use of a computer to generate designs will be found in the HANDWEAVER and CRAFTSMAN magazine. LOURIE, JANICE R., Winter, 1966, "The Textile Designer of the Future" and VELDERMAN, PATRICK, Fall, 1971, "Computer Generated Overshot Pattern."



WEAVING BY THE CARD
In 1728 a French engineer invented this automatic loom. An endless chain of punched cards was set to rotate past the needles of the loom. As the cards moved by only the needles which matched holes were able to penetrate and their threads determined the pattern.

DRAGON EMBALMERS

Mortals and/or Dragons:

We, the morticians, having observed your paper for a year, have decided to subscribe. Please enter us upon your mailing list.

We have been using a remote hook-up with the Univac 1108 at the Illinois Institute of Technology. Recently we obtained a second hook-up with the IBM 360 at the Chicago Board of Education. We have implemented several of your games on the 360. Some of these games are *Chomp, Super Wumpus, Number, Snark, Hurkle,* and soon *Star Trader.* For the most part, the translation to the 360 (using a 'rax' compiler, which we don't like at all) has been smooth. However, we did run into some problems. For example, you can't transfer to a DIMENSION, REMARK, or to the first statement of a subroutine.

Anyone who had or is having similar problems or anyone who wishes to know more about our implementations is invited to contact us at the following address:

The Morticians
3326 E. 191st. Street
Lansing, Il. 60438
c/o Paul A. Kubinski

Ordinarily, we refrain from doing anything constructive but as they say 'the times are-a-changin.' So, ye readers of PCC, we morticians in defense of the under-privileged, in pursuit of justice, and in recognition of those who are not exceptional examples to mankind but yet are consistently mediocre, end this letter.

Freedom of the Press ....
The Morticians

GAME PARLOR

Dear People's Company:

Believing that computers can serve people in their recreational as well as their business lives, we at CRI are currently preparing to open a computer time-shared game parlor in the New Jersey area. Eventually we hope to be able to provide a total computer amusement environment within which our users can

(1) play games with each other,
(2) play games with the machine,
(3) watch other people play games,
access a variety of amusement-oriented databases.

We would greatly enjoy hearing from PCC people and readers with regards to their ideas, observations, and opinions on our venture. In particular, we are currently considering for purchase any and all nature of BASIC programmed amusements, games, diversions, and pasttimes. We welcome proposals from programming enthusiasts who have or would be willing to develop BASIC programs for CRI.

Scott B. Guthery
CRI Corporation
P.O. Box F
Cliffwood, New Jersey 07721

SCOTT, LOOKS LIKE A STORY HERE.
WOULD YOU WRITE IT AND SEND IT?

TED, COME HOME – ALL IS FORGIVEN

To my honorary parents and members of my *karass* at PCC (read Vonnegut's *Cats Cradle!*)

I send you greetings and salutations once again from the land of sand, milk, and honey – although (excepting the Sinai and the Dead Sea area) more and more of the sand is turning into GREEN!

It is with pleasure that I announce that I am to give my first "paper" (to be published with the Conference Proceedings) at the second Jerusalem Conference on Technology. From what I can see, it looks like a loner in a forest of technologically-oriented articles, since the title is: "CAI – Computer Assisted Inhibition or Inspiration?" – in which I tried to point out that CAI has been losing out on its biggest potential by concentrating on reinforcing "right" answers, instead of letting children play and explore new concepts, and that continuation of dogmatic behaviorism will end up producing children inhibited towards creative thinking. Pop Albrecht is quoted (from the Saturday Review article) – I'll try to send a copy of the article to PCC, although the finks don't give me any reprints!

Your newspaper has been a smash hit here – its's being used more and more for ideas, and I suspect you will be receiving more orders from teachers over here for subscriptions of their own.

More happy news. I received an NIMH pre-doctoral fellowship (2 years) to do research on developing a series of computer assisted (Inspirational) programs to teach certain strategies of problem solving and creative thinking. One of the main ideas was to try to put certain related computer games and simulations into optimal learning sequences – like use different kinds of board games together or do a series of games related to inductive reasoning. If anyone has heard of any work being done like this, please have them contact me –

Ted Kahn
c/o Rehov Maoz, 4
Givatayim, Israel

MIKE PITT STRIKES AGAIN

Dear Big A,

YEA YEA YEA!!!

This is just a note to tell you I'm dead and living in NY.

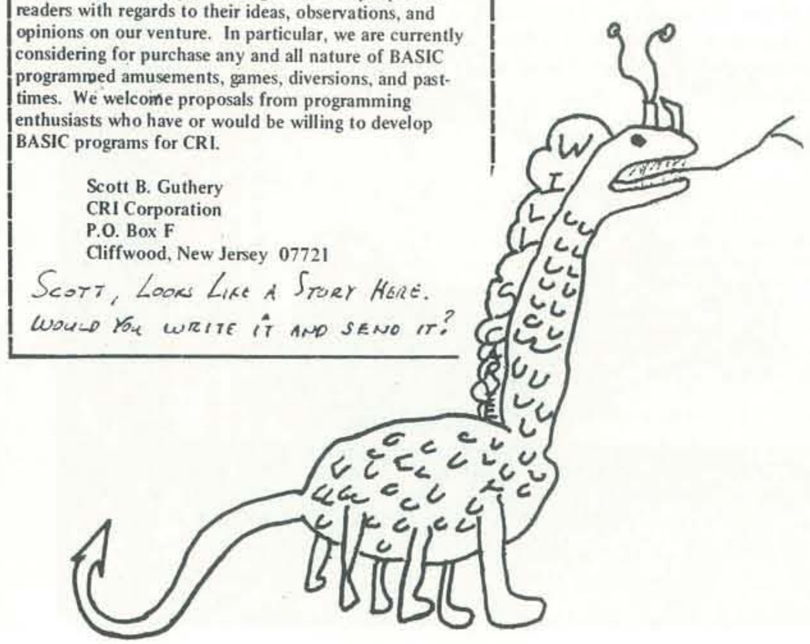
I've already got a response from my letters you printed, a guy from Texas is sending me a printout of the Game Wumpus in RPG, is that totally insane! I don't even know his name.

I'm going to Columbia University during the summer to learn me some BASIC and other stuff. OR I'LL go to a NSF funded thingy at Brooklyn Polytechnic University where they let you play around with the stuff they got there WOH!

I still don't know what to do with the MONROBOT XI don't you have any ideas? Please ask around.

Mike Pitt
213-17 86th Avenue
Queens Village, NY 11427

MIKE, YOU COULD START A MUSEUM. Bob



Saturday Review of Education

SATURDAY REVIEW EDUCATION is published monthly at $12 per year by Saturday Review, Inc. Send all correspondence about subscriptions and remittances to: Subscription Department, Saturday Review, Box 2043, Rock Island, Illinois 61207.

People's Computer Center. "Gotchal" cries a twelve-year-old as he "shoots down" an enemy spaceship blipping across a computer screen. Another computer plays football and is on the short end of a 35–0 route at the hands of a dignified middle-aged man. My computer plays "guess a number." I guess three-digit combinations; it signals "bagels" if none of the digits are correct, "pico" if a digit is correct but in the wrong place, and "fermi" if a digit is correct and in the right place. I try five times, and it types, "You got it."

Bob Albrecht runs this workshop. It's much like his People's Computer Center in Menlo Park, which, he hopes, will be a prototype for setting up "friendly neighborhood computer centers" everywhere. "We'd like people to think of us as the local bowling alley —a place to come have fun," says Albrecht. What about the schools? Another workshop leader says nearly a million students in grade school and high school are now getting some part of their education from computers. "A computer can respond moment by moment to the fluctuations of a kid's curiosity," he says. "Kids can learn to program computers as early as fourth grade."

I had reopened doors into ways of learning that I had shut or that had been closed upon me for years.

Name The Dragon

Dear PCC,

How about a name the dragon (the one with 3 heads) contest?

Ira Wexler
3031 Glen Ave.
Baltimore, MD 21215

PEOPLE'S COMPUTER COMPANY sounds like a DP-oriented WHOLE EARTH CATALOG. I'm interested.

Dover, Delaware

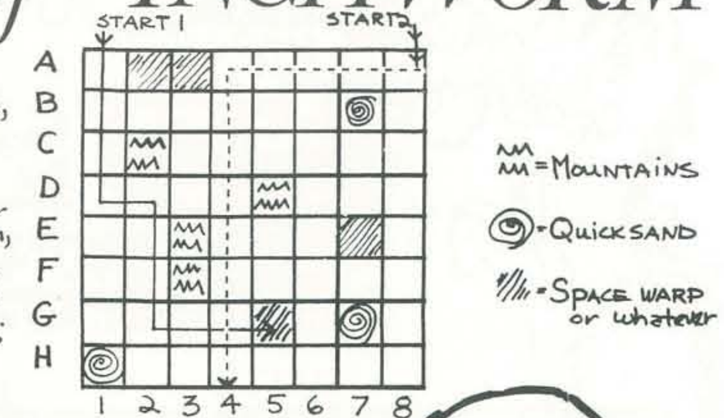
Dear P.C.C.,

'Allo, I'm in prison in Italy. I can't stop bending me mind. 2 years to go. Please bend it some more, with yr splendid newsletter. I need re-creating & educating.

Sardegna, Italy

FRIENDS of INCHWORM

Here's an idea for an INCHWORM GAME. One person makes an island grid with various obstacles (mountains, quicksand, etc.) where INCHWORM can't go. Two starting corners are left empty. Two players prepare a program for INCHWORM to wander around. Now the players try out their programs, alternating between them, one step at a time. Whenever the two INCHWORMS meet, the one who got there first goes back one step in the program on his next turn. The other INCHWORM skips one step on his next turn. The first one to step into an obstacle or go off the island loses. If they both do, neither wins. If they both end, the game is a draw. !!! see example

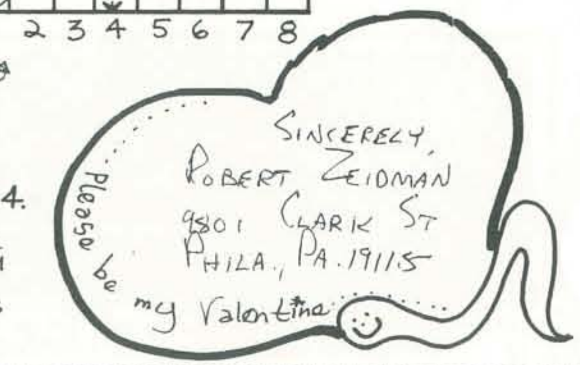


PROGRAM 1 = SSSESSSEENNEENNWW

PROGRAM 2 = WWWWSSSSSSWSWEEEE

(BOTH HAVE 17 STEPS)... Collission at G4.

INCHWORM one goes EAST from G4 into space warp at G5 and, "POOF," loses. INCHWORM two wins since it goes SOUTH to H4. (If it continues, it won't cross any obstacles anyway.)



remember : *last time we asked for your ideas ... also about* **INCHWORM'S** *maximum path*

Dear Sir:

I have found a way to beat your highest score of 19 points in 14 steps (referring to Vol. 3, No. 2) on your "INCHWORM" here to there board. My program gets a score of 21, the program is: EEEEESSSSSEES. Then I eat that with a score of 22 and in 14 steps, that path for a score of 22 is: SSESSSEEEESEES.

I have also found a way to get less than your lowest score, which you said was 6, with my score of 5 and this is also in 14 steps, the program for 5 is ESESSSESSESEEE. Are either one of these good enough to win a reward? Thank you very much.

Sincerely,

Carl Weiner

Carl Weiner
25 Maryland St.
Spfld., Ma. 01108
Dec. 12, 1974

galley:

* Minimum path
Sum: 5
PROGRAM: SEESSSESSESEE

*Maximum path in 14 steps:
Sum: 22
PROGRAM:
SSESSSEEEESEES

Sincerely,

Eric K. Olson
"The Focal-Basic-Fortran-PAL-APL-Algol-.. Kid"

My idea is to set up an INCHWORM type game along the lines of WUMPUS. One person sets up a board with various hazards. Another person is designated as the 'worm'. The object is to get the worm from corner (1,1) to the opposite corner. The person who made up the board now becomes the INCHWORM's eyes because everyone KNOWS inchworms are blind. To play, the worm starts out by moving in any direction, from (1,1). The eyes tell the worm if it comes within one square of any hazard. The worm must then avoid the hazard. If the worm hits the hazard, the hazard, whatever it is, is activated. In other words, this is the game of WUMPUS, translated into INCHWORM, with the 'eyes' person acting as the computer. How's that ?

Andy Finkel

DRAGON SMOKE

Remember your first birthday?
Maybe the cake looked like this

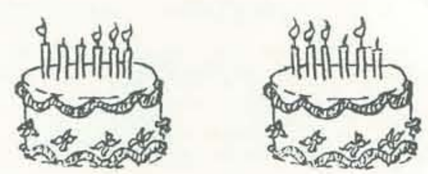
Well, on your second birthday there were 2 candles, on your third birthday 3 candles, and so on. As we grow older, the number of candles gets embarrassingly large and harder to blow out.

Sooo . . . why not try a binary birthday cake?

ONE YEAR OLD TWO YEARS OLD THREE YEARS OLD FOUR YEARS OLD

FIVE SIX SEVEN

and so on. With 4 candles, we can go up to 15 years old, with 5 candles, up to 31 years old and with 6 candles, up to 63.



39 YEARS OLD 58 YEARS OLD

But if you are 58, get people to look at the cake from the back instead of the front . . . in fact, if you don't put something on the cake to indicate the front, your age will be ambiguous (unless, of course, it is a binary *palindrome!*).

A PALINDROME!

How many candles would Methuselah need?
How many candles would Gandalf need?
How many candles would a Dragon need?

And, for all you computer people out there –

Here is a cake with an extra BYTE

How Do You Spell PEEPEL?

Look it up in the *New Phonetic Spelling Dictionary* by Behzad Kasravi. First, remove the vowels, then look up PPL and find

PPL	People	Aha! Here it is.
PPL	Papilla	
PPL	Pupil	

In other words, this is a phonetic spelling dictionary that goes strictly by the pronounced sounds of the *consonants* of words.

Interested? Go to the source.

Behzad Kasravi
INTERBOND
P. O. Box 5566
Santa Barbara, Ca.
93108
(805)962-9905

Going to buy a Minicomputer for ☆ yourself! ☆ your school! ☆ your friend?

Before you do it, GET

JANUARY 1975 VOLUME 7, NUMBER 1
Popular Electronics®
WORLD'S LARGEST-SELLING ELECTRONICS MAGAZINE

and read about

THE HOME COMPUTER IS HERE!

For many years, we've been reading and hearing about how computers will one day be a household item. Therefore, we're especially proud to present in this issue the first *commercial type* of minicomputer project ever published that's priced within reach of many households—the *Altair 8800*, with an under-$400 complete kit cost, including cabinet.

To give you some insight to our editorial goal for this momentous project, we were determined *not* to present a digital computer demonstrator with blinking LED's that would simply be fun to build and watch, but suffer from limited usefulness. High chip costs would have made this a most expensive toy. What we wanted for our readers was a state-of-the-art minicomputer whose capabilities would match those of currently available units at a mere fraction of the cost.

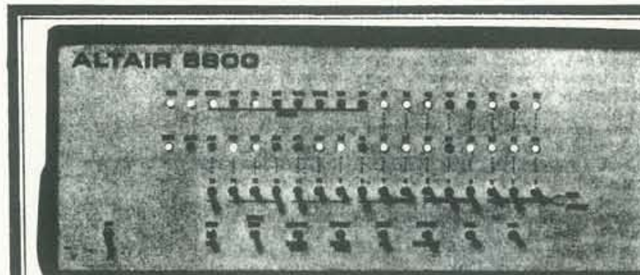
After turning down three computer project proposals that did not meet these requirements, the breakthrough was made possible with the availability of the Intel 8080 n-channel CPU (central processor unit)—the highest-performance, single-chip processor available at this time. As a result, *Altair 8800* offers up to 65,000 words of memory, 256 inputs and outputs simultaneously, buss line expansion, subroutines that are enormously deep, and fast cycle time, among other desirable characteristics. Peripheral equipment such as a "smart" CRT terminal is expected to be available, too, to make up a within-pocket-book-reach sophisticated minicomputer system.

Unlike a calculator—and we're presenting an under-$90 scientific calculator in this issue, too—computers can make logical *decisions* for an accounting system, navigation computer, time-shared computer, sophisticated intrusion system, and thousands of other applications. The "power" of *Altair 8800* is such that it can handle many programs simultaneously.

What we're presenting to you, the POPULAR ELECTRONICS reader, therefore, is a minicomputer that will grow with your needs, rather than one that will be obsoleted as you move more deeply into computerized applications. With minicomputers exhibiting an annual growth rate of some 50%, according to the E.I.A., and with predictions that six out of ten computers sold by 1975 will be mini's, you can be sure that there will be manifold uses we cannot even think of at this time.

There'll be more coverage on the subject in future issues. Meanwhile, the home computer age is here—finally.

Art Salsberg



If you want to buy your very own ALTAIR 8800, or at least find out more about it, write: MITS,INC., P.O.Box 8636, Albuquerque ,NM 87108 . . . or call(505) 265–7553

ALTAIR 8800
PROCESSOR DESCRIPTION

Processor: 8 bit parallel
Max. memory: 65,000 words (all directly addressable)
Instruction cycle time: 2 microseconds
Inputs and outputs: 256 (all directly addressable)
Number of basic machine instructions: 78 (181 with variants)
Add/subtract time: 2 microseconds
Number of subroutine levels: 65,000
Interrupt structure: 8 hardwire vectored levels plus software levels
Number of auxiliary registers: 8 plus stack pointer, program counter and accumulator
Memory type: semiconductor (dynamic or static RAM, ROM, PROM)
Memory access time: 850 ns static RAM; 420 or 150 ns dynamic RAM

The HEART of the ALTAIR 8800 is the INTEL 8080 *MICROPROCESSOR*, a complete CPU (Central Processing Unit) on a "chip." We couldn't find a picture of the INTEL 8080 in time for this issue, but here is a picture of the INTEL 8008, little sister to the 8080.

Next issue, we will start a series about the INTEL 8080 microprocessors so that people won't accuse us of becoming old-fashioned. Behold! P.C.C. leaps to the future!

ZIFF-DAVIS PUBLISHING COMPANY
Popular Electronics
Editorial and Executive Offices
One Park Avenue New York, New York 10016
212-725-3500

WE WILL PUT OUR CHIPS ON THE CHIP. IF YOU ARE ASSEMBLING A HOME COMPUTER, SCHOOL COMPUTER, FRIENDLY NEIGHBORHOOD COMPUTER, COMMUNITY MEMORY COMPUTER... GAME-PLAYING FUN LOVING COMPUTER... USING AN INTEL 8008 OR INTEL 8080, PLEASE WRITE A LETTER TO THE PCC DRAGON!

FUZZY AUTOMATA are prone to melancholy, and brood a great deal

A FULL SET OF CANINES

A Guided Tour of Computer Programming in BASIC

A Guided Tour of Computer Programming in BASIC *by Thomas A. Dwyer and Michael S. Kaufman*

available from *Houghton Mifflin Company* *Boston, MA 02107* — *People price $3.60* *School price $2.70*

If you are a 12 year old dragon and want to learn BASIC, get this book and begin! But don't be surprised if older dragons start looking over your shoulder . . . this is a fine get-started book for *anyone* who wants to learn BASIC. When you are finished with *A Guided Tour . . .* you will be ready to go it alone and learn about STRINGS and FILES and MAT statements and other exotic features of the BASIC language. *ba.*

This book is divided into four parts:

PART 1 will tell you a little about computers and what to expect of them. It will also show you how to get the computer ready to "talk" to you (this is sometimes called *logging in*).

PARTS 2 AND 3 form the main part of the tour. They show you how to write computer programs. A *program* is a list of instructions that makes the computer work for you, following your wishes with great precision and speed.

PART 4 is where the fun begins. It introduces you to professional computer applications, including such things as an airline reservation system, automated game playing, and a program that "writes" payroll records.

APL News

APL Press is a new publishing house devoted exclusively to APL. Its first book, to appear this summer, is a high school text on elementary analysis by Iverson. Several other titles are planned for publication this year, and further manuscripts are being sought.

A newsletter is also planned, to present brief articles, problems, definitions of functions, reports on conferences, correspondence, and other items of interest to the APL community. The first issue, which is scheduled for July, will include a report by Professor Jenkins on a recent APL implementors workshop, an article on magic cubes by Professor Mauldon, and material on a new form of function definition excerpted from a forthcoming book.

Readers interested in receiving the newsletter and information on other publications, or in submitting material for publication, should write to APL Press Box 27, Swarthmore, PA 19081.

"It's a subset of APL!"

© DATAMATION ®

GAMES FOR YOU TO PROGRAM

by the Dragon

REVERSE for a Video Terminal

Last time, you may recall, we described nine different variations of the game REVERSE. For you lucky people who have a video terminal with an addressable cursor, here is a version of REVERSE just for you. This version features some simple "animation" on the screen as the numbers get reversed right before your eyes. Ready? Here we go . . . the game starts like this.

```
RUN

THIS IS THE GAME OF REVERSE.  I WILL GIVE YOU A
SCRAMBLED LIST OF NUMBERS.  YOU UNSCRAMBLE THE LIST ...
PUT IT INTO NUMERICAL ORDER.  FOR EXAMPLE,

SCRAMBLED LIST:       2 5 1 4 7 6 3

PUT IT IN THIS ORDER: 1 2 3 4 5 6 7

WHEN YOU ARE READY, PRESS RETURN AND I WILL TELL YOU MORE
ABOUT THE GAME OF REVERSE█
                         CURSOR
```

Easy enough so far . . . we press RETURN and:

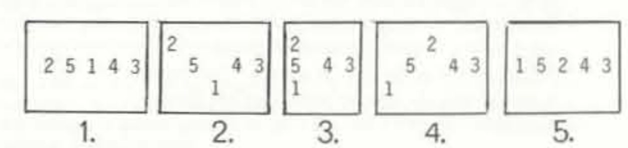
```
YOU MAY REVERSE TWO OR MORE NUMBERS IN THE LIST, STARTING
FROM THE LEFT END OF THE LIST.  FOR EXAMPLE,

  ORIGINAL LIST:   2 5 1 4 7 6 3

  WE REVERSE 3:    1 5 2 4 7 6 3

YOU MAY CHOOSE THE SIZE OF THE LIST.  YOU MAY ASK FOR UP
TO 9 NUMBERS.  HOW MANY NUMBERS DO YOU WANT?█
```

Let's try an easy game. We type 5 and press RETURN.

```
SCRAMBLED LIST:  2 5 1 4 3  HOW MANY SHALL I REVERSE?█
```

Obviously (we think), we should REVERSE 3 so that 1 will be in the first place. So, we type '3' and press RETURN.

Wow! If we could just show you the action on the screen in glorious SUPER-8! Since we can't, here is a frame-by-frame description of what happened.



The first three numbers, from the *left* end of the list, have been REVERSED.

```
ORIGINAL LIST            NEW LIST

 2 5 1 4 3               1 5 2 4 3
 └─┬─┘                   └─┬─┘
   └───────────────────────┘
```

Well, we did get 1 in the first place, but the rest of the list is still scrambled. Here is the current state of affairs.

```
SCRAMBLED LIST:  1 5 2 4 3  HOW MANY SHALL I REVERSE?█
```

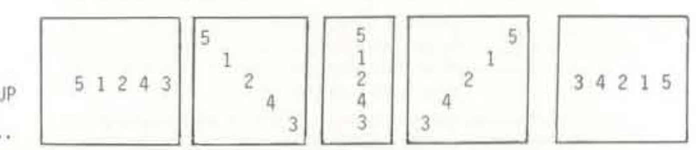
Let's reverse 2. That should put the 1 and 2 together. We type 2 and press RETURN. The 1 and the 5 march around on the screen and settle into their new places.

```
SCRAMBLED LIST:  5 1 2 4 3  HOW MANY SHALL I REVERSE?█
```

Aha! 5 is now in the first place. If we REVERSE 5, we will put 5 in the fifth place! So . . . we do it.

```
SCRAMBLED LIST:  3 4 2 1 5  HOW MANY SHALL I REVERSE?█
```

Just to remind you, we will show the action during that last move. All five numbers will move on the screen!



When you program this, be sure to leave enough room above and below the list, so that the numbers have room to move in. Remember, there can be up to 9 numbers in the list.

Eventually, we did put the list in order and the computer displayed the following message.

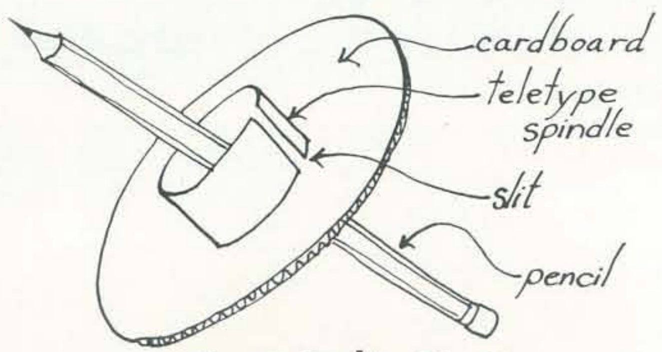
```
UNSCRAMBLED LIST:  1 2 3 4 5  YOU DID IT!!!

DO YOU WANT TO PLAY AGAIN?█
```

If you type YES, the computer will begin again, asking you how many numbers you want. If you type NO, the computer might say something like: THANKS FOR PLAYING WITH ME. LET'S PLAY AGAIN SOMETIME.

YOUR TURN. Write the program. If you do, we would sure like to see it. If you send us the program, please tell us what version of BASIC (or other language) you have used and the name of your video terminal.

TELETYPES

Cheap Tape Winder



WI$E?

Is it wise to buy a Teletype from Teletype Corp?

In the interest of saving money we checked out the possibility of buying teletypewriters from Teletype Corporation directly, rather than buying through a local dealer (which is what we previously recommended).

We found Teletype Corp. to be most cooperative. They suggested that interested readers contact their Los Angeles sales office at (213)724-6040 or their central number (312)982-2500, or write directly to the plant:

Teletype Corporation
Sales Department
5555 West Touhy Ave.
Skokie, Ill. 60076

You should note the purchase of a Teletype is a cash operation, payment is due in total in 30 days. If you want to lease a TTY, you'll have to deal with RCA, Western Union or a local dealer. You should also note that prices are F.O.B. plant, which means the buyer pays the freight charges from Skokie, Illinois – this will probably run a minimum of $20 per unit.

We were unable to get any definitive response to a question on delivery dates. Rumor has it that you will wait 6–8 months for delivery when you buy from the factory. Even if delivery is only 4 months (we've been wrong before) you have to PLAN AHEAD.

Once delivered, your TTY will have to be "installed." Though it's checked out at the factory, 1000 miles in a truck may beat it up. Installation (final test and adjustment) by Teletype will cost you $75/unit plus mileage if you're more than 25 miles from an office. Our friends at Data Terminals will only charge about $35/unit if you have the machine shipped to their office in San Jose – then they will deliver it to you. Data Terminals claims that unless you have an experienced man at your school to check out your new TTY, you could possible find yourself in *real* trouble a few weeks later. —

How Many terminals to you need?

Grades 9–12 Enrollment	Starting Program[1]	Intermediate Usage[2]	Total Program[3]
Under 500	1	3	4
501 – 1000	2	4	5
1001 – 2000	3	5	7
2001 – 3000	4	8	11
3001 – 4000	5	11	15
4001 – 5000	6	14	19
Over 5000	8	18	24

(1) Elective course in computer literacy or supplemental use in one year of mathematics. (10-15% of students will use computer regularly; 5% occasionally.)
(2) Elective course in computer literacy and supplemental use in one or two years of mathematics and use of simulations in physics, biology, or social studies. (20-25% of students will use computer regularly; 10-15% occasionally.)
(3) Elective courses in computer literacy and computer science and supplemental use in three years of math and use of simulations in two or more subject areas. (30-35% of students will use computer regularly; 25-30% occasionally.)

Compliments of Dave Ahl, DEC.

Well, is it worth it? We priced out the following unit which is perfect for a T/S terminal. ASR 33 Data Terminal with paper tape reader/punch, friction feed paper, automatic reader control (X-on, X-off), ME type wheel (give a slashed zero), pedestal mount including chad box and copy holder.

Model 3320/5JC	Price	$970
To which we add	Transportation	20
	Installation	50 (happy medium)
	TOTAL	$1040 (coupler is not included)

The same unit from our friends at Data Terminals is only $1100 and they can deliver quickly and will assume all responsibility for bad units etc.

Is it worth it? We don't think so. Not if you're buying onesy, twosy units. The $60/unit profit you give to your local dealer will save you more than that in aggravation and risk. BUT, if you're buying lotsa terminals, it might well be worth the risk to buy all your units direct from Teletype and have them installed by your local men.

Other words of wisdom –

You can buy a builtin modem from Teletype also. The modem runs an additional $200 or so. It must be hard wired to a telephone Data Access Arrangement (DAA). You cannot use a builtin acoustic coupler with the 3320/5JC for some strange reason. You have to to use a separate coupler sitting on the floor or nearby.

Teletype recommends maintenance every 750 hours or 6 months. How do you keep track? For $15 more, they will equip your TTY with an elapsed timer that records the "motor-on" time to the nearest hour. We think it's a worthwhile investment! Model SOP 188660.

There are a zillion other options available when you're buying a TTY. The unit and information described above is basic, but will meet your needs adequately.

CHIP TALK REVISITED

BY DON INMAN

Microprocessor - What is it? According to Sippl and Sippl [1], a microprocessor is "a device capable of receiving data, manipulating it, supplying results usually of an internally stored program." In the words of Brice Ward [2], "A microprocessor is a very small processor, and a processor is a special machine that has been devised for the express purpose of processing information – of performing specific tasks. It represents an extremely powerful and inexpensive design approach to a wide variety of industrial, commercial, recreational and educational applications."

The microprocessor is often referred to as a *microcomputer*. Although it does contain the arithmetic and logic functions associated with computers, it is not really a microcomputer. In the typical microcomputer, the processor is the heart of the computer. It is contained in a single chip (integrated circuit - sometimes referred to as an IC).

Such chips are called *microprocessor chips* or simply *microprocessors*. They are used in the construction of a microcomputer, minicomputer, or even peripheral devices or full size computers. To a large extent the microprocessor determines the characteristics of the microcomputer of which it is a part.

"The modern microprocessor is the product of two technological developments: Large-scale integration (LSI) and low cost semiconductor memories." LSI and semiconductor memories are described in the last issue of PCC. [3] Through these developments it has been possible to design processors which will routinely handle complex problems. The manufacturing techniques have made these chips relatively inexpensive while retaining sophisticated capabilities. Inexpensive semiconductor memories have provided a compact way for microprocessors to store instructions and data.

As a result of these developments, microprocessor circuits have been designed with relatively few integrated circuits. When memory and input/output (I/O) devices are added, it is possible to put the entire microcomputer on a single printed circuit board.

Microcomputer - What is it? Everyone agrees that it is a computer, but there the general agreement ends. The terms "microprocessor and microcomputer" have been so loosely used that they have become almost interchangeable. The microcomputer is something more than a microprocessor.

One could say that a microcomputer is merely a very small computer, but they are really a new and different product. The use of logic chips and the price of microcomputer devices becomes the most important distinction. They do share common ancestry however.

Another "gray" area exists in the difference between microcomputers and calculators. Motorola Semiconductor [4] makes this distinction: "Any hardware system will be designated computer as opposed to calculator when the following conditions are fulfilled:

- it has a random access memory for read/write operations;
- it has a controllable input-output system;
- its repertory of instructions allows:
  - a) the manipulation of words stored in the memory (arithmetic, logic or transfer operations)
  - b) the modification of any bit in a word;
  - c) transferring the control of a programme by branching when the necessity arises (decision making power of a computer)
  - d) controlling the external equipment with the aid of an interruption facility;

  the instructions, that is the programme, are stored and processed using the same hardware as for the data.

This definition already allows us to get a first idea of the structure of a computer or of a minicomputer, which has the same organization as a big computer but differs from it essentially as regards price, performance and the field of applications which it covers."

With the variety of definitions for a microcomputer, it becomes necessary to examine what a manufacturer is including when a microcomputer is discussed. Is it just the microprocessor, is it the microprocessor and closely associated chips, or is it a complete microcomputer?

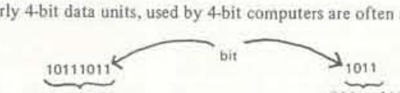
Bits, Bytes, Nibbles & Words - "The reason there is no fundamental difference between a microcomputer and any other computer is because all computer products are based on the same fundamental computing comcepts - which in turn devolve to one fundamental logic concept - that of the binary digit." [5]

A binary digit can be expressed as either a 0 or a 1. Binary digits can be likened to a switch which is either on or off. Binary digits are also like voltages - either high or low. Since computers are electronic devices, they can readily take advantage of such electrical analogies. Numbers larger than one can be expressed by a string of zeros or ones. The term "bits" can be considered as merely a short nickname for "binary digits". A bit is represented by a zero or a one.

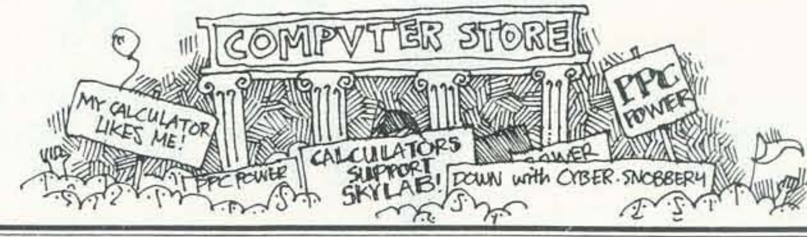
Binary numbers are used by the computer. These numbers are made up of bits. Binary numbers are organized into groups so that a computer can easily handle them. Different computers may operate with different sized groups. The particular group used by a given computer is known as its word size. An 8-bit computer is organized to handle words which are 8 bits in size.

An 8-bit data unit is commonly referred to as a *byte*. In other words, as the computer is handling data it processes the data by bytes. It "bytes" off 8 bits at a time.

Similarly 4-bit data units, used by 4-bit computers are often referred to as *nibbles*.



·PPC POWER·



by Craig Pearce

Originally published as *The Programmable Pocket Calculator Owner: Who Does He Think He Is?* in The CACHE Register, Vol 1:7; reprinted with permission.

It began innocently enough at the last meeting of CACHE, and the incident has left such a scar on me that I feel I must bring the entire thing out into the open. Of course, the whole thing should have been clear to me from the start, and would have been were it not for my naive faith in human nature.

I was speaking with Ted Nelson, author of the book *Computer Lib and Dream Machines* about the coming age of a computer in every home. It is a good and exciting dream and Mr. Nelson's enthusiasm on bringing the uPs out of the closet and away from the 'cybercrud' types that can still be found veiling these computing devices in shrouds of mystery can not be out done

Since my interests are many, I naturally brought up the subject of the programmable calculators. You'll never guess what Ted – a computer for everyone – Nelson did. He laughed! He flatly stated that if it couldn't do graphics on a CRT it wasn't a computer, and laughed. When he learned that some machines had the ability to store and play back programs on magnetic cards, he roared even more.

"Why bother . . . what for?" he added, equating that feature as being as ridiculous as a somewhat off-color joke in Mel Brooks' "Silent Movie."

Needless to say, I was stunned. Here is a man that wants to see the computer come out from the false complexities that surround it, and make that power available to everyone, and then makes a statement as he did.

Then, slowly, the pieces began to fall in place. I began to see an ever clearing picture. It isn't just Mr. Nelson alone, it's nearly everyone. Didn't Bill Precht himself include the PPC in any survey only when reminded that they exist? Hadn't all attempts to stir interest in a calculator sub-group failed? It's all a clear case of Cyber-*snobbery*.

Ah, you may say, but my machine *is* better. I can control several input and output ports, I can run things in my home, my memory is expandable, my speed quicker. And, granted, it's all true, and I would hardly be the one to want and say that the PPC is actually better than a SWTP 6800, or Altair or Imsai or what have you. After all, I own a uP too. The question is, is a uP better than a PPC? I let you be the judge.

Over the past several meetings, I have seen and heard talk on several pieces of software. These include such interesting ones as diagnostics (to see if the damn machine is working), an 8080-Educator program that actually lets you see 4 whole registers as you input a limited number of commands, one at a time. Then there is a whole list of programs that can transfer data. Now that's really something.

"What can you do with your computer?" a friend asks. "I can run a program that relocates itself in RAM!" you proudly answer. Terrific. Or how about one that will fill a CRT screen with a character or some oddball pattern. Great way to spend an evening.

And what about all those "things" that can be controlled around the house? Heating, air-conditioning and the like. We've all got that programmed in, right? We don't? Takes too long to rewire the house, you say. Don't really want to trust your machine at running all times? I see.

Okay, so what I'm getting at is that while the uP has great promise, all the predictions haven't yet come to pass.

What can the PPC do, however? Probably nothing that you couldn't program your micro to do, or course. Certainly not in the number of steps, however. For example, with the HP-65, the user has 100 6-bit program steps at his disposal. How many can program their micro to multiply two 18 digit numbers and produce a 36 digit product in 100 bytes? How about Hexpawn, or a cybernetic Nimb game? If I were so inclined, I could pass a card through my 65 and load in a 100 step program to perform "Parallax Transformations in a Celestial Reference System". I can also balance my check book, perform trend line analysis, compute components for a chebyshev filter, check male pulmonary functions, navigate a ship, fly a plane by one or two VORs, have a game of Hangman using an alphabetic over-lay of the keys, or simulate a dime slot machine that duplicates all standard payoff combinations. And I can do all this at the time I need it. My machine fits in a pocket and operates from batteries.

An owner of an SR-52 has the ability to do binary searches; linked list; manipulation of subscripted variables and arrays; interrupt processing; dynamic code modification; op code translation; linked editing, loading and execution; overlays paging; and, yes Ted, even output graphics, via the attachable printer. The new HP67/97 series opens up even more advanced programming techniques.

Speed? The programmable calculator is slow. Remember, however, that it runs in an interpretive mode. A loop that takes 15 minutes on a KIM I might take 30 days on a PPC. However, whatever reason for the loop, chances are the function is already available at the touch of a key on the PPC. Accuracy can't be beaten. The PPCs I'm familiar with have 10 digits of accuracy with a range of 1x10**-99 to 9.999999999x10 ** 99 for both positive and negative numbers. That range actually exceeds the volumn of the known universe in cubic microns!

Mr. Nelson predicts that over 10,000 people are going to attend the upcoming convention and that this will really get the public aware of computers. Well, just as a point of interest, many people I've talked to have become interested in this, the greatest of all hobbies, through the PPCs. And there are 70,000 of us. I personally belong to a PPC club from California that has over 1250 members nation wide, with membership growing through word of mouth only. Fact is, the PPC has always had more public exposure than the average micro.

Just who *does* a PPC owner think he is? He (or she) is a person that needs computing power, without the time to wait for time sharing; a person who needs this power at odd times and places that won't allow for some remote terminal. A PPC owner takes pride in accomplishing difficult computing tasks on a small, limited memory machine.

(Did anyone know that two HP-65s went along on the American/Russian skylab mission and were used to back up and confirm the results obtained by the onboard computer?)

Maybe this whole thing has been stated pretty strong. And maybe it has to be. I'm not saying down with the micro-computer. I'm saying down with cyber-snobbery. Maybe the lowly PPC can't ever hope to do all the advanced functions of a genuine micro, and it really shouldn't. But to just laugh, and think that it will never play a role in personal computing is absurd. It already has.

DATA HANDLER: ANSWERS

Here are typical answers to the homework problem on page 18. Your choice of words may be different.

Address	Data	Mnemonic	Mode	Description
FC00	A9	LDA	IMMEDIATE	Load the accumulator with the HEX number 36
FC01	36			
FC02	8D	STA	ABSOLUTE	Store the number in the accumulator into memory location FD05
FC03	05			
FC04	FD			
FC05	4C	JMP	ABSOLUTE	Change the program counter to FC05. Continually loop through steps FC05, FC06 and FC07 until halted.
FC06	05			
FC07	FC			

Application of Computer Analysis to Athletics

BY GIDEON B. ARIEL, PH.D.

Interest in athletic activity has stimulated both performer and spectator for many centuries. Concomitant with the sporting mania is the question of what differentiates the Gold Medal winner from the less successful competitor. The answer is, of course, very complex.

In the past, athletic achievement depended mainly on the individual's talent although skill was often enhanced or diminished by existing facilities and equipment. However, with the advent of new measurement tools and knowledge in the field of sport sciences, athletic achievement has attained a new dimension. Countries such as those of Eastern Europe and Cuba, with relatively small populations, have achieved a spectacular level of success in many of the athletic events which has previously been dominated by the Western world. Current evidence suggests the likelihood of this trend continuing into the 1980s. Such domination stems, in part, from the application of science to the realm of athletic performance.

The establishment of Olympic Training Centers in the United States shows an acknowledgement that winning will not result from the dedication of individual athletes alone. With the increasing international interest in competitive athletics, recreation and fitness, it was inevitable that computers would be used for the analysis of sports techniques. Application of Newtonian physics is perhaps the cornerstone of modern Eastern European and Russian sport groups. In this country, advanced technology in computers and computer graphics makes it possible to perform similar biomechanical analyses and to utilize biomechanics to quantify motion as well as optimize performance.

The first computer graphics system was developed by Computerized Biomechanical Analysis, Inc. (C.B.A.) in Amherst, Massachusetts, utilizing the first available sonic digitizer and one of the first graphic systems manufactured by Megatek, Inc. This computerized biomechanical system has grown steadily since 1970 and today consists of several Data General computer models in addition to the advanced Megatek graphics system. The application of this system to sports analyses has increased, and with the establishment of a sophisticated laboratory at the USOC Training Center, this work will continue.

Several manufacturers have made substantial donations to the U.S. Olympic efforts, including the Eclipse S/250 computer system (Data General Corp.), the Wizzard 7000 graphics system (Megatek, Inc.) and a digitizer (Talos Systems, Inc.). This equipment will be located in the Biomechanics Computer Laboratory at the U.S. Olympic Training Center in Colorado Springs and will allow the Sport Medicine Committee to apply advanced scientific-computer technology beyond the limits of what the human eye can see and intuition deduce. Human judgment is still critically important, however. As in business and industry, where decisions are based ultimately upon an executive's experience and interpretive ability, the coach is and will continue to be the ultimate decision maker in athletic training. The computer should be regarded as a tool which must be skillfully used by people in order to achieve the desired end.

Computer technology has facilitated the combining of high speed photography, anatomical data, and the utilization of man as an integral part of a system. Thus, the long tedious hours of tracing and hand calculations have been reduced to a matter of minutes and make whole body motion analysis practically obtainable. This analysis provides a quantitative measure of the motion and permits perfection and optimization of human performance.

METHOD

Biomechanical research relies primarily on data obtained from high speed cinematography and force platforms for measuring body motion and forces. The analysis of the data consisted of kinematic data including a description of the motion in terms of displacement, velocities, and acceleration of body segments and kinetic data consisting of the measurement of forces, moments of force, and center of gravity analysis.

The athlete is photographed performing the particular skill or activity of interest. After the film is developed, the films are projected upon a 30 X 40-inch glass digitizer screen. This digitizing process involves touching the projected joint centers with a stylus which transmits the X-Y coordinates into the computer memory for further analysis. As each frame is digitized, the joint centers are projected onto a graphic display screen and connected by lines to form stick figures. The complete movement can then be recreated in stick figure form on the screen so that examination and corrections, if needed, can be made. These procedures are repeated for all camera views so that three-dimensional analyses are possible. Calculation of forces and moments of force require knowledge of the mass of each segment as well as its center of gravity. These parameters are available in a publication by the Aerospace Medical Research Laboratory and are used in the present system.

Utilizing special software developed by C.B.A. for this purpose, the segment lengths, angular displacements, velocities, and accelerations are computer for each body segment. The effects of different cubic spline smoothing factors can be visually observed utilizing the Megatek graphics system. The displacement of each body segment is thus smoothed independently with as small a value as possible in order to best represent the raw data. Utilizing NASA data on the location of body segment centers of gravity, components of center of gravity displacement, velocity and acceleration are also calculated. Smoothing factors are then applied to the displacement of the center of gravity. Following these calculations, the information is then communicated to the coaches and athletes for their use.

Data acquisition at the U.S. Olympic Training Centers at Squaw Valley and Colorado Springs was guided by questions posed by the coaches and athletes of various sport groups. The analyses varied depending upon both the particular sport and the guidance sought by the participants. In some cases the data were collected during national or international competitions rather than at the training sites to afford evaluation of the superior competitors within that sport.

Some selected examples from among the numerous findings resulting from these studies will be presented here. The sports studied include long distance and sprint running, kayak, weight lifting, diving, figure skating, hammmer throwing, shot putting and discus throwing. In addition, two non-Olympic sports – golf and tennis – are illustrated.

RESULTS

Long Distance Running Figure 1 to 3 illustrate utilization of the graphic system in analyzing a runner. Although long distance running is generally considered to be a cardiovascular event, the present study revealed that biomechanical factors are extremely important since cardiovascular demands depend on the individual's work output. Running speed and the runner's work output depend on the stride length and frequency. Studies have suggested that one advantage to running with long strides is the resulting reduction in the number of strides per mile. However, our study indicated that each running stride is associated with a breaking force which stops the forward motion of the athlete. The larger the stride, the greater the resulting breaking force created. This phenomenon is a function of the relationship of the location of the body's center of gravity with respect to the location of foot contact.

When the runner extended the forward leg, the contact point was ahead of the body's center or gravity and a greater breaking force was produced. This resulted in a less efficient running motion. A stride which is too small, of course, will require a faster leg motion and more strides per mile. It was calculated that each athlete has an optimal stride length when the breaking force is at a minimum. Calculation of the precise relationship can improve running efficiency by as much as 20 percent as verified from energy measurement studies. Leaning forward slightly at the hip joint also contributed to running efficiency as did landing on the ball of the foot rather than on the heel – a common characteristic of efficient runners.

Angular displacement measurements at the knee and ankle joints revealed that running is associated with large amounts of elastic energy. The electrical potential of the muscles associated with running is activated prior to contact with the running surface. The muscular contraction is eccentric in nature, absorbing kinetic energy in the same manner as bouncing a basketball. In other words, the better runner is the one who can absorb more kinetic energy in the elastic component.

Weight lifting Once an event of American glory, weight lifting in the U.S. has lost its position to the Eastern Europeans. The reason may reflect the improved techniques developed by the winners. Figure 4 graphically portrays a Russian weight lifter. Bulgarians, Soviets, and Germans have developed coordinated techniques allowing the lifters to get under the weight and accelerate upward when the bar is at a lower point than that characteristic of less successful lifters. Our studies showed that the U.S. athletes delayed getting under the bar until it had begun accelerating downward. This technique prevented the U.S. athletes from lifting greater loads since, once the weight was descending, the lifter had to overcome both the inertial forces and the weight of the bar.

Another characteristic displayed by the European lifters was that the path of the weight was found to coincide with the path of the athlete's center of gravity. The American athletes demonstrated deviations from this center of gravity path resulting in inefficient performances.

Diving Diving is judged by aesthetic as well as performance capabilities. The American athletes share successes in this event with other countries. However, a need for a defined base line for successful performances exists.

For example, Greg Louganis has a unique technique which allows him to perform better than most divers. Our research findings revealed that Louganis' method of absorbing kinetic energy in the diving board differs from that of the other divers who were tested. This technique incorporates a coordinated movement with Greg collapsing his knees before loading the board. At the same time his arms accelerated downward, a motion which caused the direction of the force to be upward and was counteracted by the collapsing knees. When he reached a knee joint angle of approximately 90 degrees, he abruptly decelerated the body downward. This motion caused a loading of the diving board without additional body motion. At this point, Louganis accelerated his arms upward. This movement created an additional downward force adding to the decelerating force of the body and increased the loading force on the diving board. When his arms reached approximately a horizontal position, Louganis began to decelerate them. At the same instance that his arms began decelerating, the diving board started to unload with a high potential energy that was transferred to kinetic energy. At that point, Louganis prepared for the dive with the diving board providing the upward force. In other words, from this point Louganis was able to concentrate on only the diving stunt without being required to generate additional effort. Most other divers provided muscular forces throughout the dive – a phenomenon which is less efficient than Louganis' technique.

Hammer Throw At one time American throwers dominated most field events, including the hammer throw. In recent years, however, American hammer throwers have failed to produce distances comparable to those of their Soviet and Eastern block counterparts. In fact, at the Montreal Olympic Games, no American exceeded the qualifying standard of 226 feet while the Russians had more than twenty-five athletes capable of heaving the hammer that distance. A graphic illustration of the Gold Medal winnner is presented in Figure 5.

The computer analysis revealed the discrepancies between the Americans and the medal winners. The shorter throws of the American athletes were paralleled by relatively low velocities during the turns and low linear velocities of the hammer during the delivery phase. The reason for these low velocities stemmed from inefficient center of gravity displacement of the American throwers.

Non-Olympic Sports In addition to analyzing Olympic sports, biomechanical methods have been applied to recreational sports such as tennis and golf. In tennis, various top players, including Connors, Evert, and Nastase, were analyzed at the Coto De Caza Sport Research Center in Orange County, California. Figures 6 through 9 graphically illustrate several of these tennis players in action. A top tennis researcher, Vic Braden, asked various questions about the velocity of the tennis ball and the time of contact between the ball and the racquet.

The ball/racquet interaction tests provided a new, and sometimes completely different, insight into the mechanics of the game. A series of tests were performed and displayed on the graphic system each time a tennis ball was fired into a racquet held by a simulated hand grip and mounted on a force platform. The residence time varied between 3.8 and 4.2 milliseconds. Since human reaction time is approximately 70 milliseconds, it is not the ball on the racquet which is felt but rather the racquet's reaction to the impact. When the ball leaves the racquet, the racquet head has just begun to move.

Analysis of the athletes during play revealed that as much as five times the body weight must be absorbed in the knee and ankle joints. In other words, players weighing 150 pounds subject their knees and ankles to forces of as much as 750 pounds. Tennis shoes and courts, then, must be designed to have the correct energy absorption.

A term frequently heard on the tennis court concerns the "sweetspot" of the racquet. The term refers to the center of percussion of the racquet and can be calculated mathematically quite readily. It is usually found to be on the racquet somewhere between the center of the strings and the throat of the racquet. This is a result of assuming the pivot point to be at the handle. Analysis of high speed film, however, has shown that the handle-wrist-hand connection is a fairly rigid one and that the pivot point is actually the shoulder. Using the whole arm as the system results in a center of percussion slightly above the wrist. These biomechanical studies showed that there is much yet to be understood about the interrelationships of the various components that make up the game of tennis.

Golf The player's desire on the tee is to translate maximum bodily effort and complex timing sequences to the ball so that it will travel a great, yet reasonably straight, distance. In order to achieve this goal of maximum impact, which must occur within a very short time – approximately 1.2 milliseconds, the whole body works in unison. The forces of all body segments from the feet to the hands are summed and transmitted via the club to the club head. Thus, with proper timing, maximum kinetic energy will be transmitted to the golf ball, and to produce the optimum timing sequence, the body segments must be coordinated like the instruments in an orchestra. The body's center of gravity has to follow a certain pattern with specific velocities and accelerations, and each muscle must produce its own melody of effort in a coordinated pattern. In an effort to determine how "symphonic" their swings are, the golf drives of Jack Nicklaus and former President Gerald Ford were studied. Figures 10 and 11 show a portion of the swings of these two golfers. The analysis revealed that it took Ford .2 seconds to complete the swing, that is, from the first movement of the club in the direction of the ball through impact. The same stroke phase for Nicklaus required .18 seconds. At impact, Ford's club was positioned approximately 300 degrees from the right horizontal and Nicklaus' was approximately 270 degrees. In other words, Ford's club passed the vertical position by 30 degrees before meeting the ball. The ball-club position is extremely important since the application of force is the most efficient when it is applied at a perpendicular angle. In the present analysis, the perpendicular position is at 270 degrees relative to the earth. There are some occasions, of course, when hitting the ball at different or non-vertical angles could yield different, yet desirable, flight patterns. One of the most important differences between the two golfers was that Nicklaus exhibited greater separation in the timing of the peak velocity among the sequences of the different arm segments as compared to the timing produced by Ford. This separation or delay allowed a more efficient interaction between Nicklaus and his club. This physical relationship between the body segments is a demonstration of superior neuromuscular coordination originating in the higher centers of the central nervous system.

SUMMARY

Athletic achievement has emerged into the world of measurement and diagnostic expertise. With the engineering principles described by Newton and the rapid calculations provided by the computer graphic system, man and machines can lead athletic performance to new levels. The information presented in this paper has briefly described the possibilities that exist for biomechanics and athletic performance. The art of coaching men and women will certainly be enhanced by effective and timely utilization of modern medical and scientific techniques. Through the efforts of the United States Olympic Sports Medicine Committee and with the aid of the donations of the Data General computer, the Megatek Graphic system, and the Talos digitizer, this technology will be made available to coaches and athletes at all levels.

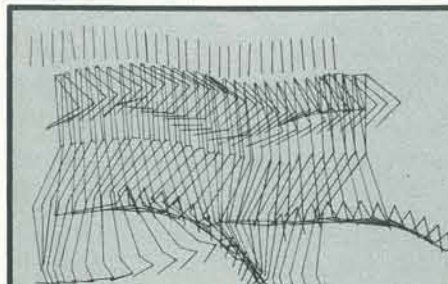


Figure 1. Bill Rodgers' Regular Marathon Pace

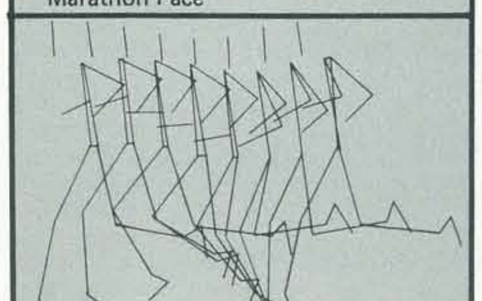


Figure 2. Bill Rodgers' Regular Marathon Pace

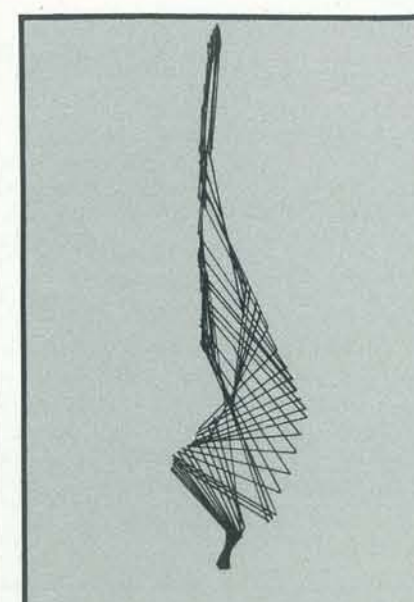


Figure 4. Voronin Weightlift

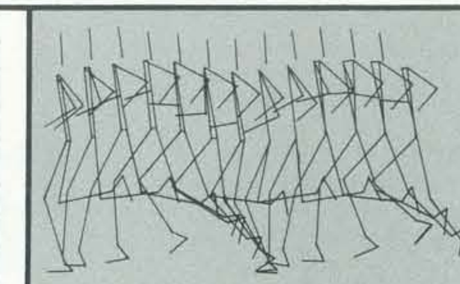


Figure 3. Bill Rodgers' Regular Marathon Pace

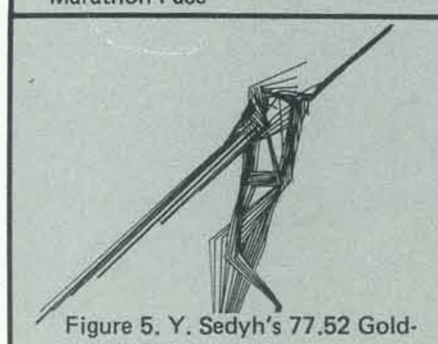


Figure 5. Y. Sedyh's 77.52 Gold-medal Hammerthrow

(continued on page 49)

WORD SEARCH
a hunt for
HIDDEN WORDS

BY LEN LINDSAY

The following program is written to work on all models of Commodore PET and CBM microcomputers, from the original PET 2001-8 to the latest 80 column screen CBM 8032. Only standard microsoft BASIC commands are used with no trickery, so other computer users should be able to modify it for their computers as well.

WHAT THE PROGRAM DOES

This program will create a word search puzzle from any list of words you supply. It can print the puzzle on your screen or printer. The puzzle dimensions are variable, and an answer key can be supplied as well as a list of the words hidden in the random looking box of letters. A MENU is provided to aid in the user's choice of what to do next. A different puzzle using the same list of words is one of the options.

OPTION MENU

1. Print a list of the hidden words to screen or printer.
2. Print the puzzle on the screen or printer.
3. Print an answer key on the screen or printer.
4. Create a new puzzle using the words already supplied.
5. Start a new puzzle with new words as input.
6. End the program.

HOW IT WORKS

Two arrays are used. W$(X, Y) is used to store each spot in the puzzle. The puzzle dimensions are X and Y. All spots in this array are a hyphen "–" unless occupied by a letter in one of the words hidden. S$(X) is the array of words to be hidden in the puzzle array. Lines 150 and 152 in the program set the size of the puzzle array. The number of words to hide is determined in lines 200 and 210.

Words are entered in lines 410 to 500. It is possible to change a word that is entered incorrectly using the REDO option. Simply enter an ARROW ACROSS " ← " as your word and you can alter or change the previous word. You also can stop entering words early by hitting the backslash as your word. Each word can be fit into the puzzle either forwards or backwards, up and down, across, or diagonally. Lines 1000-2150 calculate where to hide the words in the puzzle. Lines 5000-5140 are routines used to pick a location. If you don't want any words spelled backwards, change line 2080 as follows:

```
2080 B=0
```

Lines 3000-3130 print the puzzle. Lines 4000-4120 print an answer key. Lines 4500-4560 print a list of the words that are hidden in the puzzle.

Comments are sprinkled throughout the listing to help you keep track of what is going on. As listed, it takes under 6K bytes of RAM and thus can RUN in any model of PET with 8K or more.

HOW WORDS ARE FIT INTO THE PUZZLE ARRAY

Each word is worked on one at a time. First a direction is chosen (1 for horizontal, 2 for diagonal, and 3 for vertical) and then random locations are tried. If each position in the puzzle needed for the word is not occupied, the word is placed there. If not, another random location is tried. After 9 random tries without success, a systematic search is embarked on to place the word. If that too fails, then another direction is tried. After failing all three directions, you are informed what word won't fit and asked if you would like to try another arrangement of the words.

Each word is printed on your screen as it is placed into the puzzle. Also, while the PET is thinking about each word, it prints a " * " for each time through the main loop. This lets you know that it is still thinking and did not forget you. If the PET has trouble fitting a long word into the puzzle, try reentering your words. If you enter your longest words first and shorter ones last, your odds for success will increase.

POSSIBLE USES

Word search puzzles are an all-time favorite pastime. Now you can create personalized puzzles. Teachers can take the current list of spelling or vocabulary words and hide them in a puzzle for their students to find. Create puzzles for your next party, hiding the names of all your guests. Send a Christmas puzzle of all your family names and important events. The list of uses can go and on. But most important, it is fun. You control what words are hidden. With that power you can rule the puzzle world.

HOW TO USE THE PROGRAM LISTING

The Program listing follows the standard listing conventions for the PET special characters. All special keys are indicated by their keycap identifier enclosed in square brackets. Do NOT type the brackets and letters in these cases, rather just hit the specified key. If there is a number inside the brackets, it indicates to hit the specified key that many times. Examples:

```
[CLR].....CLEAR SCREEN KEY
[RVS].....REVERSE ON KEY
[OFF].....REVERSE OFF KEY
[DOWN]....CURSOR DOWN KEY
[LEFT]....CURSOR LEFT KEY
[DEL].....DELETE KEY
```

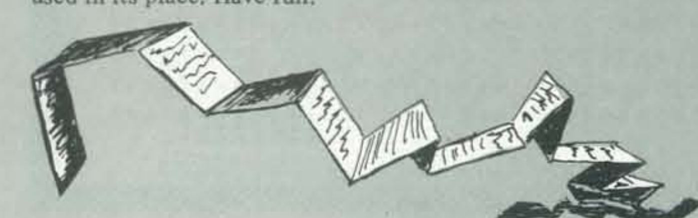
It is a two-step process to program the DELETE into your program. For each DELETE you must first hit an INSERT. And you cannot be in QUOTE MODE. Thus to have ZZ$ equal 40 DELETEs, you would type:

```
ZZ$="[DEL]"[40 INST][40 DEL]"
```

The above line will turn out to be

```
ZZ$="[40 DEL]"
```

Only one graphic character is used. In lines 10200 and 455, the "▪" is the " ? " key shifted. This character is not available on the business keyboard, and any other character may be used in its place. Have fun.



Word Search Charts

Most people who type in programs from listings usually also make some modifications or changes to suit their particular situation. For this reason, WORDSEARCH was analyzed, and the following charts have been printed for your benefit.

COMMANDS

This is a list of the BASIC commands and the line numbers that use them.

MNEUMONICS (BASIC KEYWORDS)

This chart lists all PET Microsoft BASIC keywords in a alphabetical order. The number printed before each word indicates the number of times that BASIC word was used in the program. If no number precedes a BASIC word, then the word was not used in the program. This chart will be helpful to anyone wishing to try the program on a different model computer (check if all words used are available in your dialect of BASIC). It is also worth noting that the two BASIC words used most are PRINT and REM.

ANALYSIS SUMMARY

This shows some of the program statistics. Anyone curious about program listings should be interested in this.

NUMERIC VARIABLES

This is a list of all variables used in the program, as well as the lines that they are used in. If you add a routine to the program, this chart will tell you what variable names are not used, and thus provide no conflict. Or if you change any use of a variable, the chart tells you all the other lines that also use that variable so you can check them out also.

STRING VARIABLES & STRING ARRAYS

This chart is similar to the NUMERIC VARIABLES list, except for STRING variables.

BRANCH INSTRUCTIONS

This chart is extremely useful if you wish to move some lines up or down, or delete lines completely. The first number of a line is the line number, followed by every line in the program that references it with a GOTO, GOSUB, or IF . . . THEN.

SPECIAL NOTES:

The program listing employs what many people refer to as PRETTY PRINT. Pretty print is an attempt to make a program listing as readable as possible to HUMANS. Thus, extra spaces are inserted before or after many BASIC keywords, to make the program line easier to read. All line numbers end in the same column so as to provide a nicely aligned listing. Lines within FOR . . . NEXT loops are indented by a COLON " : ". Plus REM statements are interspersed throughout the listing to aid in understanding what is going on.

When typing in the program, you do not need to type in any of the REMs nor any of the extra spaces. Colons at the beginning of a line may also be ignored if you wish.

Wordsearch Puzzle

THE HIDDEN WORDS

COMMODORE	AIM
ATARI	COSMAC
APPLE	SHARP
RADIOSHACK	SOFTWARE
COMPUCOLOR	MICROCOMPUTER
INTELVISION	RECREATIONAL
PET	PERSONAL
TRS80	EDUCATIONAL
OHIOSCIENTIFIC	ENJOYABLE
NORTHSTAR	GAMES
CROMEMCO	SIMULATIONS
ALTAIR	FANTASY
SOUTHWESTTECH	DRAGON
PROCESSORTECHNOLOGY	BASIC
KIM	PILOT
SYM	PASCAL

```
R S L S W F W A P K D Q J M F C D T B P R M K T L W V J R G D K S W Z A M R K
M O T D A V Y U S C D Q P Q U O M L T W U I K P Z T P R E S U Z M D Z G J O K
J L C W G F S N U A V T X X T M Z I C V S V A I L C I Y C N W Q E K Y U T K S
F T O I D N D Z B H U Z Z M M M M A C I Y E X T M P L Y R V E V Y P S B I T O
T B N P D Q B H X S L C T I E O M F Z R F X X Y L V O E E X X G Z E N K H X U
H I T B Q J O S R O A C D N I D F E M T O I E X L A T C A H S D M R W N A C T
V H R Y O T N O B I U P E Q P O M W N J E C T D L F U E T L A C S A P D D E H
U C K Y O J Z C Z D Z T P S R R G B T O T D O N E S B A I V J Q E F J Z D N W
A S X E Y J N T L A X C C L I E N Q A A I S U M E O T P O T R P O P W W D J E
H J B A B I F Q U R S M C G E H S M O U P S N C P I M C N Q C Y N E D X L O S
E Z A D G A M E S K D A R J R T R S 8 0 J Q I O A U C K A Z U Y J T U Y U Y T
Y G O L O N H C E T R O S S E C O R P A M F Z V I T T S L N W K B B L D Y A T
J P V Q M I A T K Q N F L A W J U U K A R X C J L T I E O L Y W J L E Y Q B E
G G Q X X I F P D J J A U N L J P J C G F I L R P E A O R I L T I R A T A L C
A D R O L O C U P M O C T M C E K K A O T A Y B V E T L N Y H Q Z I Q W O E H
M Y I H O Q V X I W Y S V L M G T T Q G S Z E K W G K N U A X O E D J F Q S L
J C R O M E M C O D Q K G X R I C L U S B M F X Z O I S I M L K R F Z G J E N
O Q I T U U O K V I U C W B W E F I D R N A A R D J P Y P M I V A E B U Y P F
U O E R X O V K B R N U M F T X U B S Q Q J G C A B X M F Q H S T T N V S A M
O P O V B Q D V O O O F G S J W F Y U A S F M M L A N O S R E P S C Q P A H I
T R V B R X U Y W V O Q K R J D I S O B B G D T W D V Z W M O N H M G B T M B
L A O P O M B Y S P M H E W F H M F B I B V K J F R F U N Y Y V T O J U N J A
G H J E W M N O G A R D Y W M Z M E R A W T F O S P O X G Y J S R X B Y A A H
Y S Z M C N X M S Z T X O J X Y K U I A G D Q C H E C T B I H E O K X J F N D
H L B N W O G I M R G M X P H S M L G I O Z B O Z I X E L Y M E N F A S L B X
```


Charts

COMMANDS

CLOSE	2330
DIM	190 220
END	2900
FN	40 500 2080 3050 5030 5040
GET	10130 10140 10220
INPUT	10270
ON	2000 2260 2330
OPEN	2320
PEEK(	10 10
POKE	15
RUN	3700

MNEUMONICS

```
   ABS(        3 AND        ASC(         ATN(        1 CHR$(      1 CLOSE
   CLR           CMD          CONT         COS(          DATA       1 DEF
 2 DIM         1 END          EXP(       6 FN         12 FOR          FRE(
 3 GET        17 GOSUB     13 GOTO      43 IF          1 INPUT      1 INT(
 3 LEFT$(      7 LEN(         LET          LIST          LOAD         LOG(
 2 MID$(         NEW       12 NEXT         NOT         3 ON         1 OPEN
 5 OR          2 PEEK(      1 POKE         POS(       57 PRINT        READ
74 REM           RESTORE   11 RETURN       RIGHT$(     2 RND(       1 RUN
   SAVE          SGN(         SIN(         SPC(          SQR(       1 STEP
   STOP        1 STR$(        SYS          TAB(          TAN(      43 THEN
12 TO            USR(       2 VAL(         VERIFY        WAIT         UNKNOWN
```

ANALYSIS SUMMARY

```
 169  PROGRAM LINES (0-10399)
 313  INSTRUCTIONS
 303  TOTAL VARIABLES
  47  VARIABLE NAMES
  50  BRANCHES
  43  LOGIC DECISIONS
  35%  RELATIVE COMPLEXITY
MINUTES TO PROCESS: 27.4508333
```

LISTINGS

```
0 REM WORKS ON PET 2001-8, NEW PET/CBM, BASIC 4.0, AND CBM 8032 & 8016
1 REM *** WORDSEARCH
2 REM *** ORIGINAL PROGRAM
3 REM *** (C) 1980
4 REM *** LEN LINDSAY
5 REM *** LATEST REVISION 4/18/80
6 REM *** WILL BE PART OF UPCOMING BOOK
7 REM *** PET GAMES AND RECREATIONS
8 REM *** FROM RESTON PUBLISHING
9 REM
10 ZP=PEEK(50003):IF ZP>1 THEN ZP=2:IF PEEK(34816)<>32 THEN ZP=3:REM SET PET
TYPE
11 REM ZP MEANS: 0=OLD, 1=NEW, 2=BASIC 4.0, 3=80 COLUMN WITH BASIC 4.0
15 POKE 59468,12
20 PRINT "[CLR]";
30 X=RND(-TI):REM SEED RND GENERATOR
40 DEF FN R(X)=INT(RND(1)*X+1):REM CHOSE RANDOM # FROM 1 TO X
50 ZZ$="[ 40 DEL]":REM 40 DELETES
51 IF ZP=3 THEN ZZ$=ZZ$+ZZ$:REM FOR 80 COLUMN SCREEN
52 ZL$="[ 10 LEFT]":REM 10 CURSOR LEFT
54 ZS$="                                        ":REM 40 SPACES
96 XX$="[CLR][RVS]WILL YOU BE USING A PRINTER?":GOSUB 10100:PRINT:IF XA$<>"Y
" THEN 99
97 XQ$="PRINTER DEVICE #":XP$="4":GOSUB 10200:PD=XA
98 IF PD<3 OR PD>15 THEN PRINT ZZ$;:GOTO 97
99 IF RP THEN RETURN
100 IF RN>0 THEN 2200:REM RESTART SKIP TO MENU
102 PRINT "[CLR]WELCOME TO WORD SEARCH"
110 PRINT "[DOWN]I WILL HIDE WORDS YOU SUPPLY INTO A"
120 PRINT "[DOWN]WORDSEARCH PUZZLE. I CAN ALSO GIVE YOU
130 PRINT "[DOWN]AN ANSWER KEY.[ 2 DOWN]"
150 AW=19:REM PUZZLE WIDTH - FOR PRINTER MAY GO UP TO   AW=39
152 AH=11:REM PUZZLE HEIGTH- FOR PRINTER MAY GO UP TO   AH=31
190 DIM W$(AW,AH):REM PUZZLE ARRAY
200 XQ$="HOW MANY WORDS TO HIDE":XP$="9":GOSUB 10200
210 NW=XA:IF NW>99 OR NW<1 THEN PRINT ZZ$;:GOTO 200
220 DIM S$(NW):REM WORD ARRAY
230 PRINT "[CLR]WHAT IS THE TITLE OF THIS PUZZLE?":GOSUB 10200:TP$=XA$
400 PRINT "[CLR]PUZZLE: [RVS]";TP$:PRINT "[DOWN]ENTER YOUR WORDS NOW."
404 PRINT "[DOWN]ENTER [RVS]←[OFF] TO GO BACK AND REDO A WORD.
407 PRINT "[DOWN]ENTER [RVS]\[OFF] TO END INPUT OF WORDS EARLY.[DOWN]"
410 FOR Z=1 TO NW
420 :IF Z<1 THEN Z=1
422 :PRINT "[RVS][DOWN]ENTER WORD NUMBER "+STR$(Z)+" PLEASE":GOSUB 10200
425 :IF XA$="←" AND Z=1 THEN PRINT "[CLR][RVS]YOU DON'T HAVE ANY WORDS YET":G
OTO 404
430 :IF XA$="←" THEN Z=Z-1:PRINT "BACK TO REDO "+S$(Z):GOTO 420
435 IF LEN(XA$)>AW AND LEN(XA$)>AH THEN PRINT "[RVS]THAT WORD IS TOO LONG":PR
INT:GOTO 422
440 :S$(Z)=XA$
450 :IF XA$="\" THEN Q=Z-1:Z=NW:NW=Q
455 :IF XA$="▘" THEN PRINT ZZ$;:GOSUB 10200:GOTO 430
460 NEXT Z
470 PRINT "[CLR]THANK YOU - PLEASE WAIT WHILE I THINK"
500 D=FN R(3):GOSUB 3310:REM INIT DIRECTION & PUZZLE ARRAY
1000 FOR CW=1 TO NW:REM PUT WORDS IN ARRAY
1005 :PRINT CW;"OF";NW;"-";
1010 :CW$=S$(CW):REM CURRENT WORD
1020 :CL=LEN(CW$):REM CURRENT WORD LENGTH
1030 :DF=0:REM DIRECTION FLAG
1040 :DF=DF+1:REM INCREMENT DIRECTION FLAG COUNTER
1050 :IF DF>3 THEN PRINT "[CLR]";CW$;" WON'T FIT":PRINT "TRY [RVS]D[OFF]":GOTO
2220
1060 :D=D+1:IF D>3 THEN D=1:REM CIRCULAR DIRECTION
1070 :DX=0:W=AW:IF D<3 THEN W=W-CL:DX=1:IF W<0 THEN 1040:REM MOVE>>
1080 :DY=0:H=AH:IF D>1 THEN H=H-CL:DY=1:IF H<0 THEN 1040:REM MOVEVV
1090 :PF=1:REM POINT FLAG 1=RANDOM,2=ALL,3=DONE
1999 :REM PICK START COORDINATE
2000 :ON PF GOSUB 5000,5100:IF PF>2 THEN 1040:REM TRY NEXT DIRECTION
2010 :REM TEST
2015 :TX=X:TY=Y:TF=0
2020 :FOR T=1 TO CL
2025 ::IF TY>AH OR TX>AW THEN TF=1:T=CL:GOTO 2050:REM WORD PAST BORDER
2030 ::IF W$(TX,TY)<>"-" THEN TF=1:T=CL:REM SPOT IS OCCUPIED
2040 ::TX=TX+DX:TY=TY+DY:REM UPDATE COORDINATE TO NEXT POINT IN WORD
2050 :NEXT T:PRINT "*";
2060 :IF TF THEN TF=0:GOTO 2000:REM NO FIT
2070 :REM ADD WORD
2080 :B=FN R(2)-1:REM 0=WORD AS IS  1=WORD BACKWARDS
2090 :FOR T=1+((CL-1)*B)TO 1+((CL-1)*(1-B))STEP 1-(2*B)
2100 ::W$(X,Y)=MID$(CW$,T,1)
2110 ::X=X+DX:Y=Y+DY
2120 :NEXT T
2130 :PRINT:PRINT "[RVS]";CW$;"[OFF]";" IS ADDED"
2150 NEXT CW
2200 RN=1:PRINT "[CLR][RVS]OPTION MENU"
2220 PRINT "[DOWN][RVS]W[OFF]ORD LIST"
2230 PRINT "[DOWN][RVS]P[OFF]UZZLE"
2240 PRINT "[DOWN][RVS]A[OFF]NSWER KEY"
2241 PRINT "[DOWN](HIT [RVS]SPACE[OFF] TO CONTINUE AFTER THE ABOVE)"
2242 PRINT "[ 2 DOWN]  OR"
2243 PRINT "[DOWN][RVS]D[OFF]IFFERENT PUZZLE-SAME WORDS"
2244 PRINT "[DOWN][RVS]N[OFF]EW WORDS & PUZZLE"
2245 PRINT "[DOWN][RVS]E[OFF]ND"
2250 XX$="[DOWN][RVS]YOUR CHOICE?":GOSUB 10100:PRINT
2255 XM$="WPADNE":GOSUB 10300:CH=XF:REM EDIT
2260 ON CH GOSUB 2270,2270,2270,4700,3700,2900
2265 GOTO 2200
2270 DV=3:IF PD THEN XX$="[CLR]PRINT TO [RVS]S[OFF]CREEN OR [RVS]P[OFF]RINTER"
:GOSUB 10100:PRINT
2280 IF PD>0 AND XA$="P" THEN DV=PD
2310 PRINT "[CLR]";
2320 OPEN 1,DV
2325 IF DV>3 THEN PRINT "PRINTING TO YOUR PRINTER"
2330 ON CH GOSUB 4500,3000,4000:CLOSE 1
2335 IF DV=3 THEN GOSUB 10130:REM FREEZE TILL KEY HIT
2340 GOTO 2200
2900 PRINT "[CLR]THAT WAS FUN - TRY IT AGAIN LATER.":END
3000 PRINT#1,LEFT$(ZS$,(2*AW-LEN(TP$))/2)+TP$:REM TITLE-PRINT PUZZLE
3020 PRINT#1:REM BLANK
3030 FOR Y=1 TO AH
3040 :FOR X=1 TO AW
3050 ::IF W$(X,Y)="-" THEN PRINT#1,CHR$(64+FN R(26));:GOTO 3070:REM RANDOM
3060 ::PRINT#1,W$(X,Y);:REM CORRECT LETTER
3070 ::PRINT#1," ";:REM SPACE AFTER LETTER
3080 :NEXT X
3090 :PRINT#1:REM CARRIAGE RETURN
3100 :PRINT#1:REM BLANK
3110 NEXT Y
3130 RETURN
3310 FOR Y=1 TO AH
3320 :FOR X=1 TO AW
3330 ::W$(X,Y)="-"
3340 :NEXT X
3350 NEXT Y:RETURN
3700 RUN
4000 PRINT#1,LEFT$(ZS$,(2*AW-LEN(TP$))/2)+TP$:REM TITLE-ANSWER KEY
4020 PRINT#1:REM BLANK
4030 FOR Y=1 TO AH
4040 :FOR X=1 TO AW
4050 ::PRINT#1,W$(X,Y);:REM PRINT LETTER
4060 ::PRINT#1," ";:REM SPACE AFTER LETTER
4070 :NEXT X
4080 :PRINT#1:REM CARRIAGE RETURN
4090 :PRINT#1:REM BLANK
4100 NEXT Y
4120 RETURN
4500 PRINT#1,"HERE ARE THE WORDS I'VE HIDDEN:"
4520 FOR Z=1 TO NW
4530 :PRINT#1,S$(Z):REM PRINT WORD
4540 NEXT Z
4560 RETURN
4700 XX$="[CLR]SAME SIZE?":GOSUB 10100:IF XA$="Y" THEN 470
4710 IF XA$<>"N" THEN 4700
4720 XQ$="[CLR]HOW WIDE (10-19)":GOSUB 10200:IF XA<10 OR XA>39 THEN 4720:REM L
INE150SETMA
4722 AW=XA
4730 XQ$="[CLR][DOWN]HOW HIGH (10-11)":GOSUB 10200:IF XA<10 OR XA>31 THEN 4730
```

(Continued on page 33)

NOTES ON A SPANISH BASIC

BY NIKI DELGADO & W. J. MORRISSEY

In the September-October 1979 issue of *Recreational Computing*, Jim Day suggested certain Spanish language words for possible use in a Spanish version of BASIC. (*These are reproduced below for reference – Ed.*) Mr. Day's idea is excellent and well founded.

The purpose of these notes is to offer some alternatives to Mr. Day's suggestions, which would be more linguistically and semantically correct.

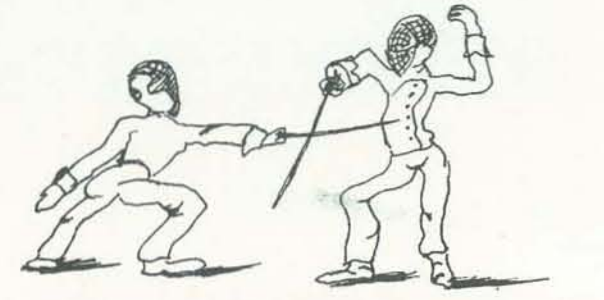
RC, September/October, 1979.

DIRECT COMMANDS

DEL	TACHA
LIST	LISTA
LOAD	PONE
RUN	ANDA
SAVE	SALVA

INDIRECT COMMANDS

DATA	DATOS
DEF FN	DEF FN
DIM	DIM
END	FIN
FOR . . . TO . . . STEP	DE . . . A . . . GRADA
GOSUB	VASUB
GOTO	VA A
IF . . . THEN	SI . . . LUEGO
INPUT	ENTRA
LET	HACE
NEXT	PROXIMO
ON . . . GOTO	POR . . . VA A
PRINT	TIPO
READ	LEE
REM	NOTA
RESTORE	RESTAURA
RETURN	RETORNO



ENGLISH	SPANISH
DEL	**BORRA** or **TACHA** – BORRA is much more commonly used than Day's TACHA, and would work as well.
FOR . . TO . . STEP	**POR . . A . . GRADO** instead of DE A GRADA. DE carries the meaning "of" except when used with an infinitive. POR much more closely translates the connotation of "for" used in this statement. GRADA refers to steps of the type found in bleachers, etc., whereas GRADO refers to degree – a much closer match for "increment".
PRINT	**IMPR** – Derived from imprime (imprimir) is used to indicate printing. When abbreviated, IMPR would be more easily recognized by a native speaker than TIPO which bears no resemblance to anything dealing with writing but instead refers to a manner of classifying as "kind". This is a good example of a false cognate.
ON . . GOTO	**EN . . VA A** gives the appropriate connotation for this statement. EN translates as "on" or "in" as "in case of". POR translates as either "for" or "to" and changes the meaning of the statement.
RESTORE	**REPONE** carries the same meaning as suggested by "restaura" but is shorter and more readily recognizable. Both terms translate as "reinstate".
RETURN	**VUELVA** is a more direct command indicating return to a previous place or state. RETORNO is a noun form which translates as "the return to" instead of the action of returning. RETORNA could be used to convey the same message but is less common and direct.

Computer language commands are designed to be practical and easily interpreted. Spanish mnemonics will work best if they convey the most accurate meaning possible rather than a derivation from approximate cognates. Cognates can be misleading.

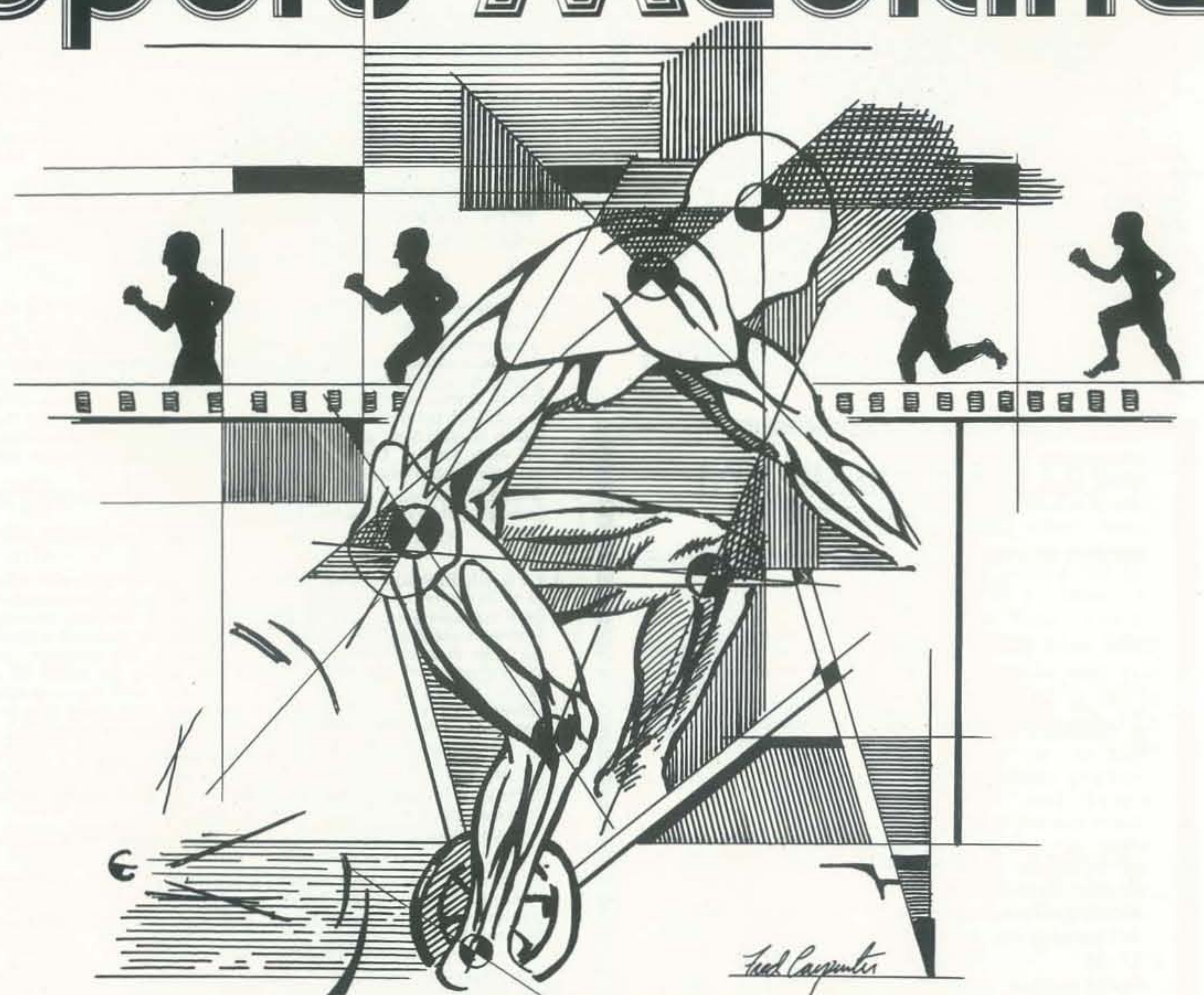
wordsearch

(Continued from page 31)

```
:REM LINE152SETMAX
 4732 AH=XA
 4750 GOTO 470
 5000 RC=RC+1:REM INCREMENT RND COUNTER-PICK RND START COORDINATE
 5020 IF RC>9 THEN PF=2:FC=0:RC=0:RETURN:REM ENOUGH RANDOM TRYS
 5030 X=FN R(W):REM X COORDINATE
 5040 Y=FN R(H):REM Y COORDINATE
 5050 RETURN
 5100 FC=FC+1:IF FC>W*H THEN PF=3:REM TRY ALL POINTS,3=NO ROOM
 5110 X=X+1:IF X>W THEN X=1:Y=Y+1:IF Y>H THEN Y=1:REM CIRCULAR NEXT SPOT
 5140 RETURN
10100 PRINT XX$;:REM GET ONE CHARACTER
10130 GET XA$:IF XA$>"" THEN 10130:REM CLEAR BUFFER
10140 GET XA$:IF XA$="" THEN 10140
10150 XA=VAL(XA$)
10199 XX$="":RETURN
10200 IF XP$="" THEN XP$="■":REM SET DEFAULT PROMPT,INPUT WITH PROMPT
10220 GET XA$:IF XA$<>"" THEN 10220:REM CLEAR BUFFER
10250 XB$=LEFT$("  "+XP$+ZL$,2*LEN(XP$)+4):REM SET UP FOR PROMPT
10260 PRINT XQ$;XB$;:REM PRINT QUESTION & PROPOSED ANSWER
10270 INPUT XA$
10280 XA=VAL(XA$)
10299 XP$="":XB$="":XQ$="":RETURN
10300 XF=0:REM INITIALIZE FLAG,EDIT 1 CHARACTER
10320 IF XM$="" THEN XM$="YN":REM DEFAULT
10330 FOR XX=1 TO LEN(XM$)
10340 :IF XA$=MID$(XM$,XX,1)THEN XF=XX
10350 NEXT
10399 XM$="":XX=0:RETURN

READY.
```

the end

Computers In Sports Medicine



BY ROBERT B. ARNOT, M.D.

In 1965, only 11 countries in the world recognized East Germany as a country. The Government and the National Olympic Committee formulated a policy which over the next 15 years brought this hard line East Bloc state from obscurity to international recognition. By 1976, the DDR (German Democratic Republic) had actually beaten the United States by winning 47 medals in the Olympics to our 40. At Lake Placid this year, they had double the number of US medals! The triumph is even more spectacular in light of East Germany's population, 17 million people—less than that of New York State.

This feat is now widely recognized to be due to a superb sports organizational system at the heart of which is sports medicine. The DDR can identify the motor genius at age three as a future gold medal winner. Future gymnasts, sprinters and swimmers may be identified before nursery school! Once selected for one of the 17 sports institutes, systems of physiologic pacing and biomechanics ensure their later Olympic triumph.

The United States, with superb basic sciences and technology, is meeting that challenge by harnessing biomedical principles with the cutting edge of American technology using the powerful minicomputer. The Sports Medicine Center at Lake Placid has used three computer-based systems over the past two years with the US Ski Team, the Luge Federation, and the Olympic Development Women's crew, as well as with cyclists, runners and figure skaters.

The Computer

For biological measurements only Hewlett-Packard and DEC have the commercially available interfaces for undertaking such a project. We selected a DEC 11/34 and will upgrade this year to the VAX or 11/60. It has 128 K core, and peripherals include a fully interactive graphics package, the VT-11, printer plotter (LXY-11), DUAL RK06 14 megabyte disks, RX02 floppy discs. Programs are written in assembly language for data collection and FORTRAN IV with extensions for data processing.

Physiologic Pacing: the Olympian's exercise prescription

Pulmonary gas exchange analysis or the measurement of expired respiratory gasses has been available literally since the discovery of molecular oxygen by Lavoisier. In the past, however, large "Douglas" bags collected the expired gasses which were chemically analyzed for oxygen and carbon dioxide content. Fast response gase analyzers and then mass spectrometers allowed accurate calculation of these values. The addition of a digital pneumotachograph which, using a hot wire anomometer, could calculate gas volumes, provided the instrumentation upon which a computer-based system could be constructed.

Using a fifty-channel 150 hz. analog digital convertor, oxygen and carbon dioxide concentrations, expired air flow rates and volumes, work load and electrocardiographic data are processed on line by the computer. These samples are measured each 1/60 second. The most difficult programming problem has been matching flow rates with the constantly changing gas concentrations. Delay time for gasses reaching the analyzers and the identification of the full respiratory cycle by the computer were the key challenges. The system processes all metabolic and cardiorespiratory data and displays fifteen-second averages during the test.

Computer driven ergometers allow the operator to deliver fixed system inputs to the athlete in order to determine what his potential is and how to make the most of it. The first of these is a ramp function by which the speed constantly increases. For a runner this would mean beginning a test at four miles an hour and continuing, if world class, to fifteen miles per hour. The computer printout would assess purely descriptive data: maximum lung and heart sizes, maximum amount of oxygen processed by the muscles, maximum heart rate. Far more important, however, a point occurs during the ramp where the abrupt increase of a substance called lactic acid is detected. This lactic acid makes our legs burn and engenders an acute feeling of shortness of breath. It is also the best pace for a marathoner to train at to increase his speed. By knowing both the speed and heart rate at which this occurs, we can then prescribe the most important part of his training program. Similar markers allow us to determine what long slow distance and interval training should be. For the US Cross Country Ski Team we used these two markers of "anaerobic threshold" for the increase in lactic acid and maximum oxygen consumed near the end of the test. Additionally, by adding differing amounts of arm and leg work it is possible to detect how effective summer training has been and tactically whether very much arm work can be added on hills without overwhelming the athlete. Further, by using a square wave where the athlete abruptly increases speed, one can detect how quickly the body can deliver oxygen on demand and with it, the effect of interval training. These tests may be repeated up to each six weeks to represcribe training programs and detect overtraining. For home computer systems this "anaerobic threshold," which is our most critical marker for any endurance sport, may be calculated easily by using an East German program and the values of lactic acid taken in a routine fitness, hospital or sports program. The microcomputer has as its input values the maximum lactic acid values taken after a workout at 70% and the 90% of race pace. This program accounts for the East Germans' successes in women's swimming, the marathon, cycling and crew.

Biomechanics

For thousands of years athletes were judged by comparison with those they compete against. Using a new computer technique, it is possible to compare any athlete against the best in the world and to further optimize his style for his given strengths and his anatomy. In this technique, two plane films are made of the world's best in any given sport at the Olympic Games or World Championships. Important landmarks, usually hip, knee, ankle and shoulder joints, are then "digitized" by projecting the film onto an xy digitizer and marking the joints with a cross hair. The computer indentifies that point and stores it. By projecting each individual frame, often at the rate of 100/second, it is possible to calculate centers of gravity, angular accelerations and limb velocities and linear displacements. From an analysis of the world's best, certain simple recommendations may be made. As an example, a ski jumper's hip extension should be 329 degrees per second at take off from the ninety-meter hill at Lake Placid. Most Americans do only 289 degrees per second. East European Lugers beat Americans off the start ramp only because of their high angular accelerations at the hip and shoulders. The key importance of the information for US athletes is that there is no longer a European mystique. We simply had not known of weight training programs that would produce these speeds. That's now in the works. Although these programs are usually run on DEC PDP 11 series or a large Data General System, at the University of Illinois Dr. Charles Dillman now has an Apple programmed to do these same calculations. The expense of that system will allow home users, schools and colleges to take advantage of these techniques.

In Lake Placid, however, we have a system that is literally on the cutting edge of US technology. Rather than digitize film, point by point, we are able to "predigitize" the joints using infrared light emitting diodes. A two-camera infrared sensing system then directly stores these points in the computer. The diodes show points accurate to 1 mm and 315 hz.

Further, by using an embedded three coordinate system, we mark each limb. The computer then calculates the instant center of motion of each joint in space. Since the knee is not a hinge at all, the conventional approach of marking a point on it is really quite innaccurate. The infrared system shows each joint fully three-dimensionally in space, processes the data on line, and gives an instant analysis after the motion is complete. We currently use this with a luge ramp in our laboratory to reengineer our athletes' starts. The hardware is made in Sweden by SELCOM and is interfaced to our 11/34 through a DR-11B which allows simultaneous rather than sequential direct memory access of data collected. The data acquisition rate is 16K/second and requires the storage space of an RK06 or larger for multi joint longer experiments. The program, developed at MIT, has been further optimized by a graduate student there, Eric Anthonson, so that it runs entirely on floppy discs. We expect this summer to have a smaller portable system. This 3D system obviates the need for gogiometers, accelerometers and, in many cases, strain gauges and allows a fully three-dimensional view of human activities.

Summary

Computers in sports medicine may be the single most important facet of developing winning Olympic teams. Without the speed and ease of data collection, the tremendous sophistication of US science and technology would be otherwise unavailable to the developing competitor. Look to Los Angeles, 1984, for the results! ●

PROGRAMMING PROBLEMS

© 1980, BY BOB ALBRECHT, DON ALBERS AND JIM CONLAN

PROBLEM #24 THREE DIGIT NIVEN NUMBERS

You may wish to solve, or at least read, PROBLEM #22 TWO-DIGIT NIVEN NUMBERS before you try this one.

A Niven number is a positive integer which is divisible by the sum of its digits.

- 720 is a Niven number because 7 + 2 + 0 = 9 is a divisor of 720.
- 123 is *not* a Niven number because 1 + 2 + 3 = 6 is *not* a divisor of 123.

Write a program to compute and print or display all three-digit Niven numbers. A three-digit Niven number is a Niven number in the range, 100 to 999, inclusive.

PROBLEM #25 NIVEN NUMBER TEST SUBROUTINE #1

A Niven number is a positive integer that is divisible by the sum of its digits.

- These are Niven numbers: 7, 24, 120, 1236
- These are not Niven numbers: 37, 123, 1234

Write a subroutine to find out if a number is a Niven number. We will call the number to be tested, NPT. Your subroutine should RETURN with FPT = 1 if NPT *is* a Niven number or FPT = 0 if NPT is *not* a Niven number.

NPT ——▶ [Your Subroutine] ——▶ FPT = { 0 if NPT is *not* a Niven number. / 1 if NPT is a Niven number.

Use this program, or a similar program, to check out your subroutine.

```
100  REM *** NIVEN NUMBER TESTER

200  REM *** ASK FOR NUMBER TO BE TESTED
210  CLS
220  PRINT : INPUT "ENTER AN INTEGER, 1 TO 999999" ; N
230  IF N<>INT (N) OR N<0 OR N>999999 THEN 220

300  REM *** USE NIVEN NUMBER TEST SUBROUTINE
310  NPT = N
320  GOSUB ________  ◀ line number of your subroutine

400  REM *** PRINT RESULT OF TEST
410  IF FPT = 0 THEN PRINT NPT "IS NOT A NIVEN NUMBER"
420  IF FPT = 1 THEN PRINT NPT "IS A NIVEN NUMBER"
430  GOTO 220

     (Your Subroutine)
```

```
ENTER AN INTEGER, 1 TO 999999?  123
 123 IS NOT A NIVEN NUMBER

ENTER AN INTEGER, 1 TO 999999?  24
 24 IS A NIVEN NUMBER

ENTER AN INTEGER, 1 TO 999999?  3.14 ◀ (Not an integer)

ENTER AN INTEGER, 1 TO 999999?  0 ◀ (Not a positive integer)

ENTER AN INTEGER, 1 TO 999999?  1000000 ◀ (Too big)

ENTER AN INTEGER, 1 TO 999999?  and so on.
```

Not an integer

Not a positive integer

Hang on to your subroutine. You will find it useful in future problems.

PROBLEM #26 RUNNING DICE TOTALS

A six-sided die is thrown repeatedly until the running total is more than 12.

THROW NO.	RESULT	TOTAL
1	4	4
2	1	5
3	2	7
4	1	8
5	4	12
6	3	15

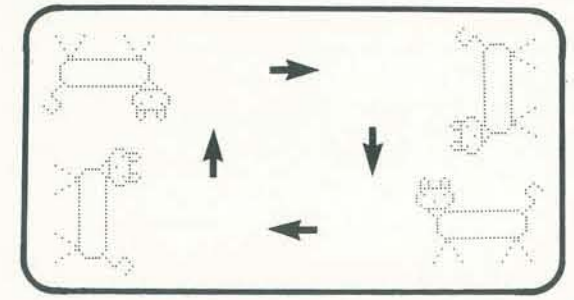
This time, we got 15. Possible final totals range from 13 to 18. Write a program to simulate this process N times. Count the number of times each possible final total (13 through 18) occurred. After N times, print or show the results in a table such as the one below. Two RUNs of our program are shown.

HOW MANY TIMES? 1000		HOW MANY TIMES? 10000	
FINAL TOTAL	NUMBER OF TIMES	FINAL TOTAL	NUMBER OF TIMES
12	310	13	2728
14	261	14	2448
15	162	15	1879
16	139	16	1499
17	87	17	960
18	41	18	486

How would we do this problem without a computer? For each possible final total, what odds might you give in a betting situation?

PROBLEM #27 FOUR CATS CHASING

Four cats are in the four corners of the TV screen of your computer (one cat per corner). Suddenly, the cats begin to chase each other. Each cat chases the cat which is in the nearest corner, clockwise from its corner.

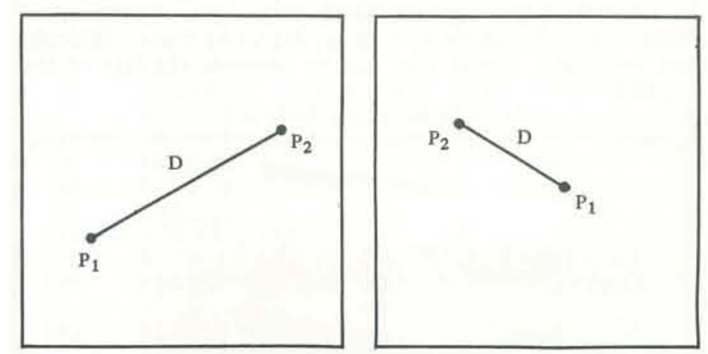


Write a program to make it happen on the screen of *your* computer. Will the cats ever meet? If so, where? Will their paths ever cross? If so, where? Will there be a cat fight?

You might want to leave a trail of cat-tracks (paw pads?) behind each cat. We also suggest that you put in a variable time delay so that you can easily slow down or speed up the action.

PROBLEM #28 SQUARISH POINT PICKING

Two points are chosen at random within a square of side 1. What is the probability that the points are more than one-half unit apart?



MORE THAN ½ THIS TIME — LESS THAN ½ THIS TIME

Do this by computer simulation. Write a program to simulate picking two points at random N times, where N is supplied in reponse to an INPUT statement. After each pair of points is picked, compute the distance between them. If the distance is greater than ½, add one to X, the success counter. At the end of N times, the simulated probability will be S/N.

Here is a RUN of our program.

```
HOW MANY TIMES?  100
DISTANCE GREATER THAN .5:  53 TIMES
SIMULATED PROBABILITY IS .53

HOW MANY TIMES?  1000
DISTANCE GREATER THAN .5:  497 TIMES
SIMULATED PROBABILITY IS .497

HOW MANY TIMES?  234
DISTANCE GREATER THAN .5:  122 TIMES
SIMULATED PROBABILITY IS .521367521
```

OK, someone. What is the theoretical probability?

PROBLEM #29 THREE-DIGIT PERMUTATION PRIMES

A *permutation prime number* is a prime number obtained by a permutation of the digits of another, *different* prime number.

- There are no one-digit permutation primes.
- There are four pairs of two-digit permutation primes, shown below. 13, 31 17, 71 37, 73 79, 97

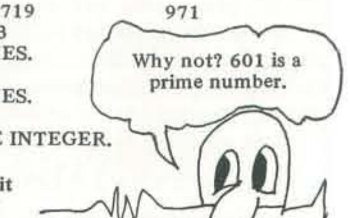
Let's look at 3-digit numbers. Here are some sets of 3-digit permutation primes.

179, 197, 719, 971
337, 373, 733
107, 701

For this problem, a three-digit number is an integer in the range 100 to 999, inclusive. Therefore, 017 and 071 are not *three-digit* prime numbers. Leading zeros don't count.

Write a program to compute and show all three-digit prime numbers which are permutations of the digits of any three-digit number entered by someone. For example. here is how a RUN might look:

```
3-DIGIT POSITIVE INTEGER?  791
PERMUTATION PRIMES ARE:
 179          197          719          971
3-DIGIT POSITIVE INTEGER?  123
SORRY, NO PERMUTATION PRIMES.
3-DIGIT POSTIVE INTEGER?  601
SORRY, NO PERMUTATION PRIMES.
3-DIGIT POSITIVE INTEGER?  37
THAT'S NOT A 3-DIGIT POSITIVE INTEGER.
```



Which three digits provide the greatest number of 3-digit permutation primes?

PROBLEM #30 THREE-DIGIT PERMUTATION, AGAIN

A three-digit permutation prime is a prime number in the range 100 to 999 which can be obtained by a permutation of the digits of another, *different* prime number in the range 100 to 999.

If you haven't done PROBLEM #29 THREE-DIGIT PERMUTATION PRIMES, we suggest that you at least read that problem before doing this one.

This time, write a program to compute and print a frequency distribution showing the number of 3-digit permutation primes which can be generated from 3-digit numbers. Show *only* those 3-digit numbers which actually do generate permutation primes.

- 100 doesn't.
- 101 doesn't, even though it is prime. *
- 102 doesn't.
- 103 doesn't, even though it is prime. *
- 104 doesn't, even though 401 is prime. *
- 105 doesn't.
- 106 doesn't, even though 601 is prime. *
- 107 does. Aha! Got one! 107 and 701 are prime.

The second 3-digit number which generates 3-digit permutation primes is 113 (113, 131 & 311). What is the third?

Side problem: There are exactly 900 three-digit numbers. How many of these numbers generate permutation primes?

*Why not? Read the definition of a three-digit permutation prime.

THREE-DIGIT PERMUTATION PRIMES

If we RUN your program, we expect it to begin like this:

3-DIGIT NUMBER	NUMBER OF PERMUTATION PRIMES
107	2
113	3

and so on.

PROBLEM #31 PERSISTENCE OF A NUMBER

Pick a number. Pick a positive integer. Let's pick 237.
Multiply the digits. 2 x 3 x 7 = 42
Multiply the digits of the above product: 4 x 2 = 8
Stop! The result is a one-digit number (8).
Again. This time we pick 397.

1. 3 x 9 x 7 = 189
2. 1 x 8 x 9 = 72
3. 7 x 1 = 14
4. 1 x 4 = 4 STOP!

Got it? Start with any positive integer. Multiply the digits to get another positive integer. Multiply the digits of that integer to get another . . . and so on, until the result is *one-digit* (0 to 9). Here is an example of a number which ends in zero (0).

Start with 12345
1. 1 x 2 x 3 x 4 x 5 = 120 2. 1 x 2 x 0 = 0

The *persistence* of a number (positive integer) is the number of times it takes to get to a one-digit number, 0 through 9.

- The persistence of 237 is 2.
- The persistence of 397 is 4.
- The persistence of 39 is 3.

Write a program to compute the persistence of a positive integer. Here is a sample of our program.

```
POSITIVE INTEGER, PLEASE? 123
THE PERSISTENCE OF 123 IS 1

POSITIVE INTEGER, PLEASE? 99
THE PERSISTENCE OF 99 IS 2

POSITIVE INTEGER, PLEASE? 123456789
THE PERSISTENCE OF 123456789 IS 2

POSITIVE INTEGER, PLEASE? 333333
THE PERSISTENCE OF 333333 IS 4
```

and so on.

This problem was inspired by the following article: N. J. A. Sloane, "The Persistence of a Number," *J. Recreational Math.*, Vol. 6, No. 2, Spring 1973, pp. 97-98.

PROBLEM #32 TRIPLE THREAT

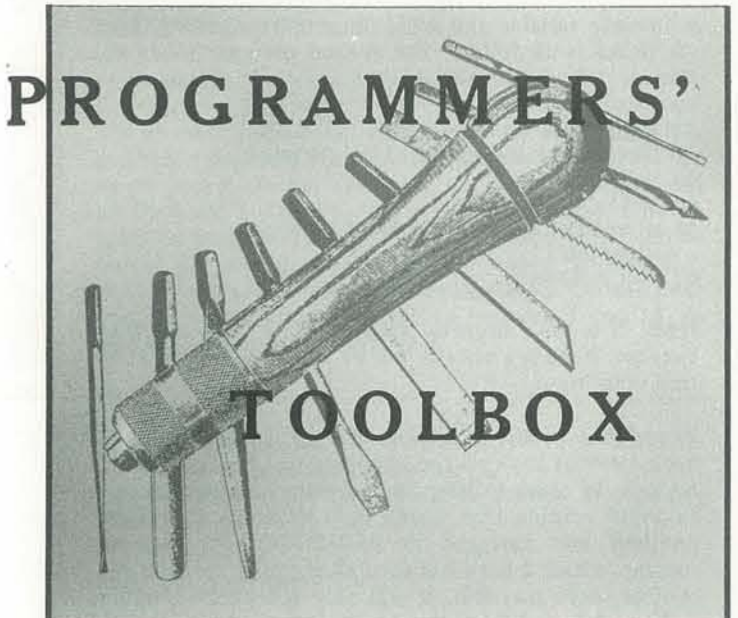
The triplets each got a new electric blanket for their birthday this year. The blankets are of the very latest design. They cool in the summer and heat in the winter. The blankets have a remarkable temperature range. Unfortunately, there was a catastrophe the very first night that the blankets were used. The controls got switched around. Arnold got the control for Bertrand's blanket, Bertrand got the control for Clem's blanket, and Clem got the control for Arnold's blanket. You can imagine the difficulties that followed. Things got worse and worse.

Arnold likes to be cool and sets his blanket at 71 degrees. Bertrand is more moderate and sets his blanket 72. Clem likes to be warm and sets his blanket at 73.

The temperature in the room was 72 degrees when they climbed into bed. Arnold was too warm at 72 so he decreased his control (Bertrand's blanket) one degree. Meanwhile Bertrand, being perfectly comfortable at 72, did not change his control (Clem's blanket.) Clem was feeling a bit cold at 72 and increased his control (Arnold's blanket) by one degree. Well, things got worse and worse.

After one minute had gone by, all the changes started to take effect. Arnold was now feeling warm at 73, Bertrand was feeling cold at 71, and Clem was feeling cold at 72. They all turned over and changed their controls by one degree. Well things got worse and worse.

After one minute had gone by, all the changes had started to take effect and it would take a computer to describe how things changed, minute by minute, degree by degree, throughout the night. Would you care to compute the fate of the triplets?



PT 23 : NAME INVERTER

People's names, when stored in a file, are most conveniently stored in the form

last name, first name

to simplify alphabetic sorting. However, when you're writing a letter or printing a mailing label, the order must be reversed. The routine below, written in TRS-80 DISK BASIC, accomplishes this task by placing the portion of the name following the comma first. If there is no comma, the name is printed without change. The subroutine in lines 110 through 130 accepts A$ (last name first) and generates B$ (first name first).

```
5    CLEAR 100
10   LINEINPUT "ENTER LAST NAME FOLLOWED
     BY COMMA THEN FIRST NAME."; A$
20   GOSUB 110
30   PRINT B$
40   GOTO 10
50   END
110  SIZE = LEN (A$)
120  N = INSTR(A$, ". ") : IF N = 0 B$ = A$ : RETURN
130  B$ = RIGHT$  (A$,  SIZE – N) + " " + LEFT$(A$,
     N – 1)
140  RETURN
```

CONTRIBUTED BY RICHARD M. GILMAN

PT 24 : DOLLAR EDIT

This program will print dollar values in the traditional financial format with a leading dollar sign and commas at intervals of $1000 and a leading zero if the amount is less than $1. It should work on most BASIC systems with string variables, but you should be wary about two things: the local definition of STR$ and the size of the local floating-point data representation. In the implementation used here, STR$ returns the string representation of a positive number without any leading blank; many systems return the number with a leading blank which will have to be removed by adding

```
9155  Z$ = RIGHT$( Z$ , LEN(Z$) – 1)
```

As to the other, the constant in line 9120 should be changed to reflect the local floating-point representation accuracy. A 23-bit mantissa allows accurate representation (that is, to the penny) of a maximum of $83886.08. That's small by today's financial standards.

```
  50 DIM Z$(40)
 100 INPUT Z
 200 GOSUB 9000
 300 PRINT Z$
 400 GOTO 100
9000 REM THIS SUBROUTINE PRODUCES A DOLLAR VALUE FOR THE
9010 REM VARIABLE  Z  IN Z$. THE VALUE OF Z IS DESTROYED
9020 REM IN THE PROCESS. THE RESULTS ARE SIMILAR TO THE
9030 REM LONGER AND MORE COMPLICATED ROUTINE BY MIKE
9040 REM DONAHUE PUBLISHED IN THE NOVEMBER 1979 KILOBAUD
9050 REM MICROCOMPUTING (PAGE 164).
9060 REM ONE MUST BE CAREFUL NOT TO CALL THIS ROUTINE
9070 REM EXPECTING ACCURATE RESULTS WHEN Z IS LARGER THAN
9080 REM THE MAXIMUM NUMBER OF CENTS WHICH CAN BE REPRESENTED
9090 REM EXACTLY. IN FINANCE ONE DOES QUIBBLE ABOUT FRACTIONS
9100 REM OF PENNIES . . .
9110 IF Z<0 THEN 9130
9120 IF Z<100000 THEN 9150
9130 Z$ = "$  *****"
9140 RETURN
9150 Z$ = STR$(INT(100*Z+0.5))
9160 IF LEN(Z$) >  2 THEN 9180
9170 Z$ = RIGHT$("000"+Z$,3)
9180 Z = 5
9190 IF LEN(Z$) <=Z THEN 9230
9200 Z4 = LEFT$(Z$, LEN(Z$)–Z) + "," + RIGHT$(Z$,Z)
9210 Z = Z + 4
9220 GOTO 9190
9230 Z$ = "$"+LEFT$(Z$,LEN(Z$) – 2) +"."+RIGHT$(Z$,2)
9240 RETURN
9250 END
```

? – 10.50	*Negative number*
$ *****	*Not allowed*
? 1	*one*
$1.00	*One dollar*
? 10.50	
$10.50	
? 99.345	*½ cent*
$99.35	*Rounded up*
? 9999.45	
$9,999.45	
? 99929292.9	*Too big?*
$ *****	*Yes!*
? 99999.99	
$99,999.99	*Note comma*

CONTRIBUTED BY DENNIS ALLISON

PT 25: CONTOUR PLOTS

When investigating functions of two variables, it's often useful to study a graphical representation. Here is a technique that allows one to easily generate a contour plot of a function of two variables on the terminal or on the line printer. For simplicity, we assume the function values are stored in an M by N array, A.

To form the plot, one first finds the maximum and minimum values of the function stored in the array. The string, R$, contains the characters to be printed for the various densities, beginning at the smallest. From the maximum and minimum values, one computes the increment per step, D,

D = (MX – MN) / (LEN (R$) – 1)

To make the plot, one then steps through the array, one element at a time, selecting and printing the right character. The expression

INT((A(I,J) – MN)/D) + 1

will take on values from 1 at the function's minimum to LEN(R$) at the function's maximum. Using this as an index in

```
MID$( R$, INT(A (I, J) – MN)/D + 1, 1)
```

selects the appropriate character to print. This choice of index truncates away from the maximum so only points which attain the maximum will be so displayed, with the maximum density character.

CONTRIBUTED BY BRUCE K. OPITZ

PT 26: YET ANOTHER STRING SQUEEZE

Here's a squeezed version of the squeeze program I wrote when I had problems getting PT8. Another String Squeeze appeared in the Sept/Oct 1979 *RC*.

```
90   CLS
100  INPUTAPT$:GOSUB200: PRINTGPT$: PRINT: CLEAR: GOTO100
200  BPT$ = "HI DRAGON": BPT = LEN(BPT$): APT = LEN(APT$): IFAPT = 0ORBPT = 0THENRETURN
210  FORJPT = 1TOAPT: UPT$ = MID$(APT$, JPT,1): FORKPT = 1TOBPT: VPT$ = MID$(BPT$, KPT, 1): IFUP
      T$ = VPT$THENGPT$ = GPT$ + UPT$
220  NEXT: NEXT: RETURN
```

CONTRIBUTED BY W. T. BURNHAM

PT 27 : TRS-80 INPUT USING INKEY$

Here's a simple little program which demonstrates the use of INKEY$ on the TRS-80. INKEY$ inputs are particularly useful in program environments where you want complete control of the input; one such case in real-time games where the keyboard input controls the play.

```
5    REM -- A LITTLE ROUTINE FOR AN INPUT
             STATEMENT WITHOUT USING "INPUT".
             AS A FUN BONUS, IT OUTPUTS BACK-
             WARDS. NOTE: INPUT IS LIMITED TO
             15 CHARACTERS UNLESS YOU CLEAR
             MORE STRING SPACE.
10   CLEAR: CLS: PRINT "WHAT IS YOUR NAME?":GOSUB 100
20   PRINT @ 384, CHR$(30): PRINT @400, C$: FOR T = 1 TO 1000: NEXT T: GOTO 10
100  A$ = INKEY$:IF A$ = "" THEN 100
110  B$ = A$ + B$: C$ = C$ + A$
120  IF ASC(A$) = 13 THEN RETURN
130  PRINT@400, B$: GOTO 100
```

CONTRIBUTED BY W. T. BURNHAM

PT 28: INPUT WITHOUT TEARS. HOW TO AVOID READY WHEN PEOPLE MESS UP ON INPUT.

I teach elementary school, and the biggest problem with programs is the kids' ability to get out of RUN mode on INPUT or GET statements. Since it takes teacher time to reRUN or often reLOAD programs, I have searched for ways to protect INPUT statements from curious program users.

The most common problem is simply hitting RETURN before data, especially in programs calling for speed. A very simple way to solve this is shown below:

```
10 INPUT"→→→$←←←";A$
20 IF A$="$" THEN PRINT "↑↑": GOTO 10
30 your program
```

In this sample program, $ is any character of your choice. If you want it to look like the cursor, type shifted SPACE. I use $ for string entries. A$, of course, could be any string variable. PRINT"↑↑" in line 20 moves the cursor up to reprint line 10 in the same place on the screen. ↑ is cursor up, ← is left cursor, and → is right cursor.

For numeric input, you can use the same method, but you have to add lines to change your string variable into a numeric variable and avoid the problem that VAL(A$) =0 if A$ is all letters. The revised program looks like this:

```
10 INPUT"→→→$←←←";A$
20 IF A$="$" THEN PRINT "↑↑": GOTO 10
30 N=VAL(A$)
40 IF A$="0" GOTO 60
50 IF N=0 THEN PRINT"PLEASE ENTER A
   NUMBER!↑↑↑": GOTO 10
60 your program
```

Here, N is your numeric variable. Line 40 is necessary because, if A$ is a zero, N will be zero and line 50 would trap your input.

Experience in the classroom has shown that kids will hack away at keys on the keyboard just to see what will happen. In order to keep the program running, I needed an input routine that would only recognize letters and numbers, and disregard the RUN/STOP key. The next routine, which I have put into all my programs as a subroutine, does just that. It will only accept the numbers 0 through 9 and the letters A through Z, or the shift of those characters. Line 9020 uses modulo arithmetic to convert the shifted characters into unshifted ones.

```
9000 ZY$="" : POKE 537, 136
9010 GET ZZ$: IF ZZ$="" THEN 9010
9020 IF ZZ$= CHR$(13) THEN 9100
9030 ZZ= ASC(ZZ$)- INT(ASC(ZZ$)/128)*128
9040 IF ZZ<48 OR ZZ>90 THEN 9010
9050 IF ZZ>57 AND ZZ<65 THEN 9010
9060 PRINT CHR$(ZZ); : ZY$=ZY$+CHR$(ZZ): GOTO 9010
9100 PRINT: POKE 537, 133
9110 RETURN
```

For new-ROM PETs, lines 9000 and 9100 should have POKE 144,49 and POKE 144,46 instead of the values given above.

Line 9000 sets ZY$, which will accumulate characters to form the word, to a null string and disables the RUN/STOP key. This also stops the timer. Line 9010 gets a character from the keyboard. The next line exits the routine if RETURN has been pressed. Lines 9040 and 9050 check to see that the character is a number or a letter; all other characters are simply ignored by returning to the GET statement. If a letter or number has been entered, the character is printed and added to the string ZY$, which makes up the entered word.

I appreciate the assistance of Brian Howell and Peter McCloud in designing the last routine. You have to try this routine to believe it; you can whack away at cursor, shifted keys, anything you want, and nothing happens. When you type letters or keys, they appear on the screen just as if there was an INPUT statement.

CONTRIBUTED BY GLENN FISHER

PT 29: RANDOMIZING ROUTINE

This routine makes all elements of a list appear in random order before any element is used a second time. It is written to move list elements as little as possible.

At each pass, the range of the random number function decreases by one. The element of the list chosen by the random number function is placed at the end to the active list, which are those elements still in the range of the random number function.

For example, if there are 20 elements in the list, the random number can be between 1 and 20 on the first pass. The element picked will be put in place #20 in the list, and other elements will be moved up one place to fill the empty space. On the next pass, the random number will be between 1 and 19. Let's say the random number function picks element 5. Elements 6 through 19 will be moved up one space (element 6 is moved to place #5, etc.). Element 5 is then put in place #19. On the next pass, the random number will be between 1 and 18, so element 5 will not be used again.

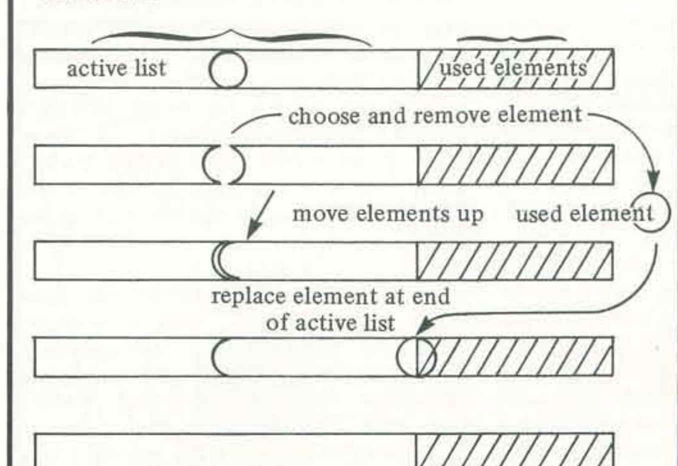
DEFINITIONS:

K is the random number.
N is the number of elements in the list.
T counts the number of times through the routine.
W$(K) is the data in the list.

PROGRAM LISTING

```
 10  T = 0                          (initialize counter)
 20  T = T + 1                      (increment counter)
100  K = INT(RND(1) * (N–T+1) + 1)  (takes random number between 1
                                     and the last unused element)
110  PRINT W$(K)                    (your use for the element)
200  IF K >(N – T) THEN 300         (don't reshuffle if last element)
210  Y$ = W$(K)                     (save element)
220  FOR I=K TO (N–T)               (move following elements up one place)
230  W$(I)=W$(I+1)
240  NEXT I
250  W$(I)=Y$                       (put element at end of active list)
300  GO TO 20                       (next pass)
399  END
```

This illustration shows how the program moves elements in the list:



CONTRIBUTED BY GLENN FISHER

PT 30 : A SIMPLE CARD SHUFFLING PROGRAM

Here is a very simple card shuffling program I wrote one day after seeing a friend's abysmal attempt. This is wrtten in Apple Integer BASIC but can easily be adapted to other BASICs. Different amounts of cards (or whatever) can also be easily accommodated.

```
10   REM LOAD ARRAY
20   DIM CARDS (52)          It is up to the user to
30   FOR I = 1 TO 52          interpret CARD(X) as being
                               a certain card
40   CARD(I) = I
50   NEXT I
60   REM SHUFFLE
70   FOR I = 1 to 52
80   J = RND(52)             Pick a card and . . .
90   TEMP = CARD(J)             . . . switch . . .
100  CARD(J) = CARD(I)             . . . the two . . .
110  CARD(I) = TEMP                   . . . cards.
120  NEXT I
  .     rest of program
  .
```

This program segment takes each card in order and selects a random card for it to change places with. On occasion, I will equal J, but that is acceptable since this is a random process.

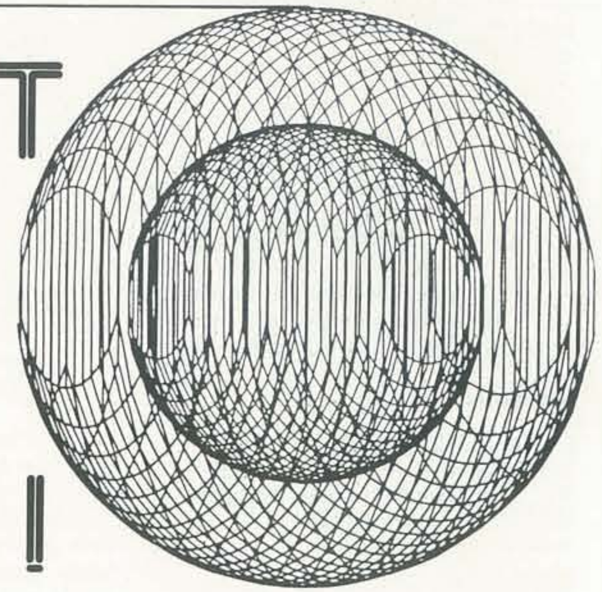
The Apple Integer RND(X) function will supply a number between 1 and X, which can make life nice, but for those with the more general RND function which supplies a real number between 0 and 1, line 80 can be replaced with

```
80  J = INT( (RND(1)*52)+1)
```

This is a very fast shuffler, so if you want increased randomness, you can perform the shuffle two or three times.

CONTRIBUTED BY PHIL REED

LET'S HEAR IT FOR THE SPHERE!



BY THE OLD SOLDIER

In the fall of 1975, if you wanted a personal computer you bought an ALTAIR or you rolled your own. Other machines were on the market, but the only one that gave promise of any real effort at full system availability was the big "A," grinning at you with its face full of toggle switches and flashing lights. Keyboard, CRT, printer, and other peripherals were optional extras available "Soon."

This was the situation when I, the proud owner of a secondhand DEC PDP-8L and a superannuated Teletype, journeyed to the 1975 WESCON.

Parked across the street from the convention center was a motor home emblazoned "SPHERE – LOWEST COST COMPUTER SYSTEM." Inside I saw a device resembling a CRT terminal, with display and keyboard included, and – wonder of wonders – a full computer inside! No flashing lights or switches!

I was introduced to the marvels of the resident monitor on PROM, a working BASIC, at an assembled cost under $2000. Never one to make hasty decisions, I took the literature home to think things over. My PDP-8L was an able computer, but it suffered from the limitations of a restricted and unexpandable memory, the necessity of toggling in a bootstrap loader program, and the agony of a 3-6 minute load time for BASIC. Magnetic tape and disks were available, but their price was (and still is) commensurate with the original $25,000 price of the computer.

The SPHERE was the answer to a maiden's prayer. In late September I made my deposit, paying the balance by mid-October. Soon after, beguiled by SPHERE ads, I sent money for additional memory and a printer. So far, lots of money out and nothing to show for it. I was learning the joys of purchasing a product in the early days of the production cycle. Was I nervous? You can bet your ASCII I was.

Sometime in January of '76 the great day came. My SPHERE arrived. Disappointing news #1, no cassette interface yet since the implementation of the "Kansas City Cassette Standard" was not finished. Bad news #2, garbage on the screen when I turned the SPHERE on.

Several phone calls to the SPHERE plant in Bountiful, Utah, and I finally got a blinking curser and the capability of typing in programs in HEX. When the cassette board finally arrived in late January, I loaded a BASIC that had to be the slowest BASIC extant. Rumor was that it was an "Emulator" BASIC. This meant that a program was read into the SPHERE to make it look like another computer, then that computer's BASIC was fed in. The overhead for all this was so great that benchmark programs running 20 to 30 seconds on other machines would take from 3 to 5 minutes on the SPHERE. To make the situation more aggravating, this BASIC would not support cassette storage of programs or printer operation.

During the subsequent months, after hundreds of phone calls, one trip to the factory and a half dozen shippings of the computer or suspect boards back to the factory, I finally had reasonably reliable operation, but the promised BASIC was always "Just a few days from complete."

At about this time I might have chucked it all had not "Programma Consultants" come on the scene. Headed by Mel Norell, a SPHERE dealer and probably the world's first personal computer store owner, Programma was independently developing SPHERE software, as well as serving as an information center for other SPHERE owners developing software. In short order I had an excellent integer BASIC with cassette utilities and printer driver. My SPHERE finally started earning its keep.

The SPHERE Corporation was still there through 1976, with a sporadic newsletter and frequent new product announcements, along with recurring assurances that the promised BASIC was still "Just around the corner." In spite of their past track record, I was still sold enough on their future prospects to invest in a disk drive. Shortly after that, in early 1977, SPHERE went under. Several attempts at reorganization and/or acquisition went for naught. I owned an orphan.

The utility of my SPHERE increased slowly until June of 1977, when I finally located and purchased an independently developed floating point disk BASIC with all the bells and whistles. I finally possessed a computer that, for about $6,000, delivered what I wanted – two years after I thought I had purchased it.

In the next two years I developed enough software to make the SPHERE an indispensable adjunct to my enginerering practice. I began to worry what would happen if my essentially unsupported system were to die on me. Of all the computers available in early 1979, the one least likely to become an orphan was the TRS-80; so I bought one and proceeded to start moving my software across. My SPHERE is still functioning perfectly, though, with not one cent spent for maintainance in the past year and a half. For my SPHERE, I have six different BASICs, PL/S, FORTH, text editor and formatter, and a dual cassette system that has never – repeat NEVER – failed to read or write successfully. Eat your heart out, Radio Shack.

During my SPHERE ownership, I learned a great deal more about the internal workings of computers than I had ever wanted to know. My initial pique at that noble crew from Bountiful, Utah, for their many broken promises has been replaced with admiration for what they did accomplish. SPHERE was two years ahead of the pack in their development of the package configuration that is now the industry standard for microcomputers.

I can't help wondering if – if some far-sighted financier had plugged in enough to get that new BASIC out, or if one of the bailout operations had succeeded, or perhaps even if the editors had held off a while longer, Bountiful, Utah, would be up there with Ft. Worth, Texas, and Silicon Valley as a major computer center.

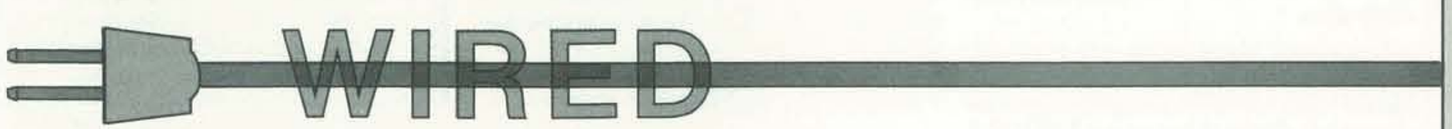
Perhaps a thousand SPHERES were sold. There still exists a users group of around 100. Via a monthly newsletter, software and hardware hints are traded. The assets of SPHERE were purchased by an East Coast company with the intention of developing a turnkey business system; so perhaps the SPHERE marque will again see the marketplace.

If I had waited until everything I wanted in a computer was available off the shelf, I would have saved myself quite a bit of aggravation. On the other hand, I did learn a great deal about computers that has stood me in good stead. Even with the two-year delay getting fully on-line, I was still up at least a year before other companies were offering systems functionally equivalent to my SPHERE at anywhere near the price. I count myself fortunate to be one of the privileged few to own a genuine 1975 SPHERE. I think I'll keep it: it's paying its way.

WIRED By Mark Singer

Not every hobbyist keeps his TRS-80 on his bedside table, but then few of us live out our electronic fantasies with the élan of Pierre Schwob. Mr. Schwob has five micros in his Manhattan apartment; in addition to the TRS-80, there's an Apple II, a Sol, a Sorcerer, and an IMSAI – not to mention four dual-disc drives, a high-speed printer, a modem, and . . . the list gets staggering.

A writer from The New Yorker *visited Mr. Schwob last spring, and his report, which appeared in the magazine's "Talk of the Town" section on May 21, follows. We dedicate our reprint to all searchers after the perfect home brew.*



Pierre Schwob hates to lose touch. When the temperature is in the seventies and the sun is shining, he enjoys nothing more than spending the day indoors, in his apartment on the West Side, running up his electric bill. It is an entirely adequate apartment – eat-in kitchen, roomy foyer, living room with park view, two baths, two bedrooms. In one of the bedrooms, Pierre Schwob sleeps and plays; in the other, he works and plays. When we dropped by for a visit the other day, a portable table next to his bed was bare. Usually, Pierre Schwob, a slender, elegant, dark-haired man in his early thirties, keeps a TRS-80 microcomputer on this bedside table. Because we had come to see the TRS-80, which Schwob calls his "man-in-the-street computer," the news that he had lent it to a man in the street disappointed us. Before we could feel too let down, however, he pointed out that right there at the foot of the bed was another entertaining piece of electronic equipment. This turned out to be an Apple II microcomputer. It prints its output on a Sony Trinitron television set that sits on top of an adjacent table. ("My first television," Schwob said, in a sentimental way.) A storage rack loaded with video cassettes of several recent popular movies sits on top of the television set. The room also contains two video-cassette tape recorders, one of which can be programmed a week ahead of time; two cable-television channel selectors, one with a Home Box Office hookup; an Advent VideoBeam 710 television projector and an Advent VideoBeam five-foot-diagonal television screen; a digital alarm clock; a stereo radio; an electric blanket; a Sanyo answering machine and a Pageboy II receiver (in Schwob's pocket); and, side by side on the floor near the bed, a red touch-tone telephone and a white rotary-dial telephone. One of the telephones rang. Schwob knelt down and felt each of them. "The red one is for business, and the white one is my private number," he said. "When one rings, I feel them to see which one is vibrating. I guess I should have lights put on them." microwave oven and another cable-television channel selector with Home Box Office. "I didn't really know anything about computers until August, 1977," Schwob told us. "Then I read several books about them and talked to a lot of people, and three months later I bought the kit to build the Sol. It was so well designed that the first time I hooked it up, it came to life."

Schwob's affection for the Sol hasn't diminished his fondness for some of the other things in the living room: a combination turntable, AM/FM tuner, and amplifier; an eight-track tape player; two cassette recorders; a reel-to-reel tape recorder; a switching box; a stereo frequency equalizer; four speakers; a Beam Box indoor FM antenna; and two boom microphones, which he uses to record himself playing the piano. The piano stands in the foyer.

In this instance, the red phone had been ringing. It stopped when his secretary, Judy Silberman, who was standing next to her desk in the living room, answered the call. Her telephone is connected to a console with thirty-two buttons, which houses, among other things, the telephone's memory. If Miss Silberman wants to call, say, Schwob's lawyers, banker, accountant, or doctor, the police, or Schwob's Pageboy II receiver, she pushes the appropriate button on the console. In the course of a year, she saves at least fifteen minutes not looking up numbers and another fifteen not punching them out on the touch-tone phone. At her desk, she also has an I.B.M. self-correcting Selectric II electric typewriter, which functions as the printer for a microcomputer – a Sol, from Processor Technology – that sits on another part of her desk. The Sol also has a video monitor and a keyboard. Schwob feels the same way about the Sol that he does about the Sony Trinitron television set in his bedroom. "My first microcomputer," he said as we entered the living room, having walked through the foyer and past the kitchen, where there is a programmable

A repairman sat at Miss Silberman's desk and tampered with the I.B.M. typewriter. He held a hammer in one hand and a long, narrow tool in the other. With the hammer he struck the butt end of the tool. Schwob saw this and winced. "You sure you know what you're up to, hitting my I.B.M. with a hammer?" he asked, smiling. "What are you doing, anyway – chiselling your initials in there?" He watched some more, and then said to us, "Why don't we go to the computer room?"

We had already seen two computers and the spot where a third one usually stands, but we had not yet, it seemed, been to Pierre Schwob's computer room. Now he led us there. The computer room used to be the second bedroom. A bed could no longer fit inside. There are, around the room, seven hundred and eighty-five switches and buttons, belonging, in part, to two computers, a Sorcerer, from Exidy, and an IMSAI (which Schwob calls "the Cadillac of microcomputers"), and there are two keyboards; three video monitors; four dual-disc drives for feeding data to the computers; a dual-trace oscilloscope; an L/C/R bridge, for measuring inductance, capacitance, and resistance; two digital multimeters, for measuring voltage, current, and resistance; a signal tracer; a radio-frequency generator; a transistor tester; an integrated-circuit tester; a universal-frequency counter; a modem (in effect, a translator that allows two computers to talk to each other over the phone); a high-speed paper-tape reader; a high-speed printer; two "joysticks," for performing computer graphics; six cassette tape recorders; two stereo speakers; a monaural computer speaker; a third cable-television channel selector with Home Box Office; a digital clock radio; a wristband radio; a television camera; a 35-mm. camera; several pocket calculators; a photocopier; an alarm system that emits low levels of microwave radiation; five or six soldering irons; a wall-to-wall, floor-to-ceiling bookcase jammed with volumes in French and English; a laser; and a seven-horsepower outboard motor.

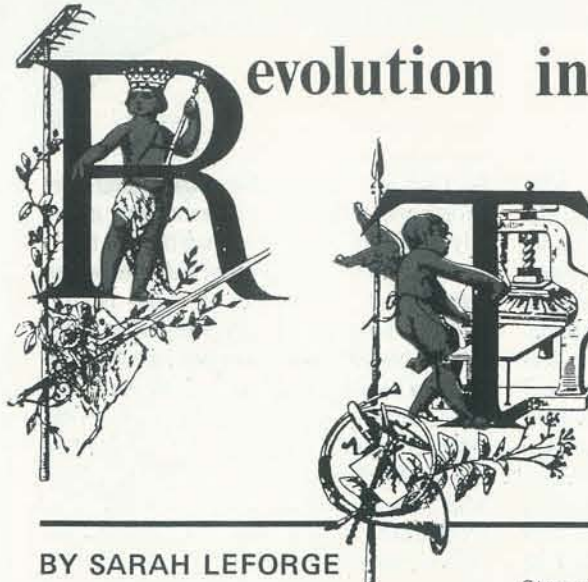
"I occasionally fantasize about aiming the laser out the window and scaring the wits out of the people on the street," Schwob said. "I actually use it, though, to do holography. I use the outboard motor on a little dinghy." His computer-room telephone, which has its own thirty-two-button console and its own memory, rang. After a brief conversation, he hung up and said, "Would you like to play a game?"

He inserted a thin plastic disc in one of the disc readers on the IMSAI computer and typed in some instructions on the keyboard, whereupon the rules for a game called Target flashed on a video monitor. After a few more instructions, blips began to float horizontally across the monitor. "The blips are airships," Schwob said. "The smaller blips are worth two hundred points and the larger ones are worth one hundred points. You aim with these arrows." He pointed to two keys marked ">" and "<." "You fire with the space bar." We watched the blips float by, and we watched Schwob, casually tapping the space bar, track and destroy them with ease. Occasionally, a pair of blips collided, and when that occurred a third blip descended on a vertical line. "This is a particularly cruel game," he said. "When two airships strike each other, a parachutist ejects, and if you shoot a parachutist, it's worth six hundred points. Here, we'll sit and watch for one." A few moments later, blips collided, the parachutist ejected, Schwob obliterated him, and a shower of simulated debris rained down the screen.

Most of the equipment in the computer room began to find its way there when Schwob realized, after building the Sol, that his fascination with computers had grown. His initial interest developed from an interest in electronic calculators. He had started collecting them (he has winnowed his collection down to half a dozen) when the first models came on the market, ten years ago. In 1975, he published a book titled "How to Use Pocket Calculators." His oeuvre has since expanded to include "The Chess Tutor; Opening Moves" and a book, which he collaborated on with the Austrian historian Friedrich Heer, titled "Great Documents of the World: Milestones of Human Thought" (selections from the Code of Hammurabi, the Laws of Solon, Plato's "Republic," Paul's Epistle to the Romans, the Koran, "Pacem in Terris," the plaque aboard Pioneer 10, and so on). Schwob has ideas for a few other books, including one to be called "The Encyclopedia of Basic Knowledge," but they will have to wait, because he has been preoccupied recently with his duties as president and sole owner of a young company called PRS – The Program of the Month Corporation. Although he is the only person on his block who owns five microcomputers, he feels confident that in time most households will have at least one. "Today, most of the people who own microcomputers are hobbyists," he told us. "Unless they write their own programs, whatever programs are generally available to them are written so badly that you have to have a Ph.D. to understand them. I happen to come from a consumer-oriented background. My father owned Contis-Frawa, a chain of women's-clothing stores in Switzerland. My company will write and sell intelligible computer programs to the average consumer. Pretty soon, programs will be sold in bookstores. I think computers are great tools – but, still, only tools. My goal is cheap software."

With the cheap, and easy, software, Schwob, who seems to think that the energy crisis will end any day now, plans to help the owner of the garden-variety home computer ask his hardware to turn out the lights, cook dinner, pay the bills, order groceries, feed the tropical fish, turn on the lawn sprinkler, play games, and keep track of friends. "*My* friends," Schwob announced, like a herald, as he inserted another disc and saluted the keyboard. His address book flashed on the video monitor. Most of the entries were women's names. "I'm a bachelor," he said. "The list is dynamic." He pointed out that the list also included the addresses and phone numbers of his most trusted computer dealers, his favorite restaurants, and the garage where he parks his car. "On the video monitor – that's 'soft' copy," he said. "The printer will give 'hard' copy – a sheet of paper. I can print the names and addresses and take the paper with me in my car. I have a telephone in the car. If I don't have a dinner reservation, I can look at the list and quickly call and make a reservation. I don't have a computer in my car. Not yet. I need to work on that."

There was a pause. "And now I must show you one more thing," he said. He stood up and walked out of the computer room, and we followed. "Just in case all systems fail," he said, pointing to an object on the wall. It was an ebony abacus, made in Taiwan.



Revolution in Typography?

BY SARAH LEFORGE

How many times in the course of history has the axiom "Necessity is the mother of invention" proven true? A recent example involves Professor Donald E. Knuth of Stanford University's Computer Science Department. "Necessity" is a system of high quality printing for mathematical books, to replace the rapidly disappearing monotype process using hot lead typeface. "Invention" is Knuth's development of a mathematical typography applying mathematical concepts to program the design of page formats and individual characters. With his new systems, TEX and METAFONT, Knuth explains to a computer where to place characters on a page and how to draw them.

When the second edition of volume two of Knuth's book, *The Art of Computer Programming*, was being printed, he learned that an entirely photographic process was being used. The results were much less satisfactory than those of the lead type and photography process used for the first edition. Knuth realized that the decline in quality meant real trouble for him and everyone publishing mathematical books in the future. Seeking alternatives, he examined the experimental systems of computer typesetting already available and concluded that their developers lacked sufficient knowledge about printing and the mathematics needed for printing. He reasoned that the problem was not one of programming a computer to copy existing fonts, as had been done up to this time, but of designing fonts for the new equipment. He asked himself how the great type designers of the past would do this.

Intrigued by the notion that if he "could find a purely mathematical way to define the letters and convert them to discrete raster patterns" he would have solved the problem once and for all, Knuth canceled his sabbatical to South America and began to work.

Step one—research on the history of type design at the Stanford Library. Knuth found that the idea of defining letters mathematically dates back to the fifteenth century. Subsequent work in letter design convinced calligraphers that too strict adherence to the rules of mathematics, a rigid use of compass and ruler, failed to produce letters of "calligraphic grace." Strictly geometric letter forms were not pleasing. (Knuth's METAFONT system includes a randomization feature whereby he avoids the too perfect letter.)

Informed and encouraged by his research, Knuth began developing his own systems. In answer to the problem of representing two-dimensional formulas as a one-dimensional sequence of instructions for transmission to the computer, Knuth designed the language he calls TEX (pronounced "tech"). The language is simple, clear, and unambiguous. A great advantage of TEX is that it shows an author exactly how his pages will look when printed. He can make changes or corrections immediately, if desired, without the worry of second party error or misinterpretation.

TEX is designed to produce a unified system which blends in the mathematical features with the word-processing routines. "The main idea of TEX is to construct what I call *boxes*. A character of type by itself is a box, as is a solid black rectangle, and we use such 'atoms' to construct more complex boxes analogous to 'molecules,' by forming horizontal or vertical lists of boxes . . . . A mathematical formula breaks down into boxes in a natural way; for example, the numerator and denominator of a fraction are boxes, and so is the bar line between them (since it is a thin but solid black rectangle). The elements of a rectangular matrix are boxes, and so on."

Describing the new printing equipment used for digital typography, Knuth says that it essentially treats each page of a book as a huge matrix of 0's (positions to be left blank) and 1's (positions to be inked). "The total job of a system like TEX now becomes one of converting an author's manuscript into a gigantic bit matrix." There are clear indications that "discrete raster-based printing devices will soon make the other machines obsolete for nearly all publishing activity . . . . The ultimately relevant thing will be mathematics; the mathematics of matrices of 0's and 1's!"

While TEX programs the page design, METAFONT programs the individual characters and symbols. Fig. 1 shows the digits 0 to 9 drawn by METAFONT.

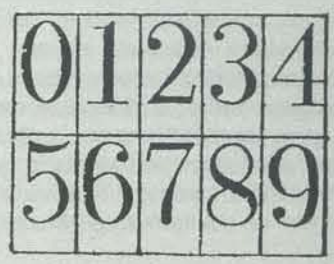


Figure 1.
Digits 0 to 9 drawn by the METAFONT programs. (Further refinements to these characters will be made before the font has its final form.)

The computer program for drawing the numeral 3 directs, "First draw a dot whose left boundary is 1/6 of the way from the left edge to the right edge of the type and whose bottom boundary is 3/4 of the way from the top to the bottom of the desired final shape. Then take a hairline pen and, starting at the left of the dot, draw the upward arc of an ellipse; after reaching the top, the pen begins to grow in width, and it proceeds downward in another ellipse in such a way that the maximum width occurs on the axis of the ellipse, with the right edge of the pen 8/9 of the way from the left edge to the right edge of the type. Then the pen width begins to decrease to its original size again as the pen traverses another ellipse taking it down to a position 48% of the way from the top to the bottom of the desired final change . . . ."

This program, Knuth explains, ". . . describes the curve traveled by the *center* of the *pen*, and the shape of this pen is allowed to vary as the pen moves. The main advantage of this approach is that the same definition readily yields a family of infinitely many related fonts of type, each font being internally consistent."

The numeral 4 illustrates another feature of METAFONT, the eraser. This was used to cut the top left of the thick line at an angle.

Using the techniques of METAFONT, a complete font of 128 characters can be produced in about two months. In the past it has taken designers months, sometimes even years, to create a font.

Not all symbols lend themselves readily to a computer program. Special problems arose for Knuth in the formation of S. Sleepless nights and assistance from his wife at length gave him an acceptable S. "I finally came up with a satisfactory solution [the middle S in Fig. 2], somewhat like those used in the sixteenth century but generalized to ellipses. Each boundary of each arc of my S curve is composed of an ellipse and a straight line, determined by (i) the locations of the beginning and ending points, (ii) the slope of the straight line, and (iii) the desired left extremity of the curve."

METAFONT can be extended to create any number of fonts by changing the parameters of the program. Each program has about twenty parameters which specify features such as the sizes and proportions of various parts of the letters. A change in the parameters produces a new font all of whose characters are consistent and harmoniously related. Fig. 3 shows the effects of varying parameters to METAFONT.

Figure 2.
Different S's obtained by varying the slope in the middle. (This shows 1/2, 2/3, 3/4, 1, 4/3, 3/2, and 2 times the "correct" slope.)

Anyone interested in the printed word cannot help being excited by the promise of TEX and METAFONT. As a final note, these systems, the offspring of sophisticated equipment and Professor Donald Knuth's mathematics, will not eliminate the need for artists to design type face. They will provide a challenging new medium.

The first edition of Knuth's book, *The Art of Computer Programming*, won for him the National Medal of Science. The second edition started him on a quest that may signify a new age in printing and win him a place in history beside Johannes Gutenberg, the inventor of movable type.

Note: For further information about Knuth's work see "Mathematical Typography" by Donald E. Knuth, *Dr. Dobb's Journal*, Mar. '80, pp. 5-20 and the Stanford University News Service release of 2/26/80.

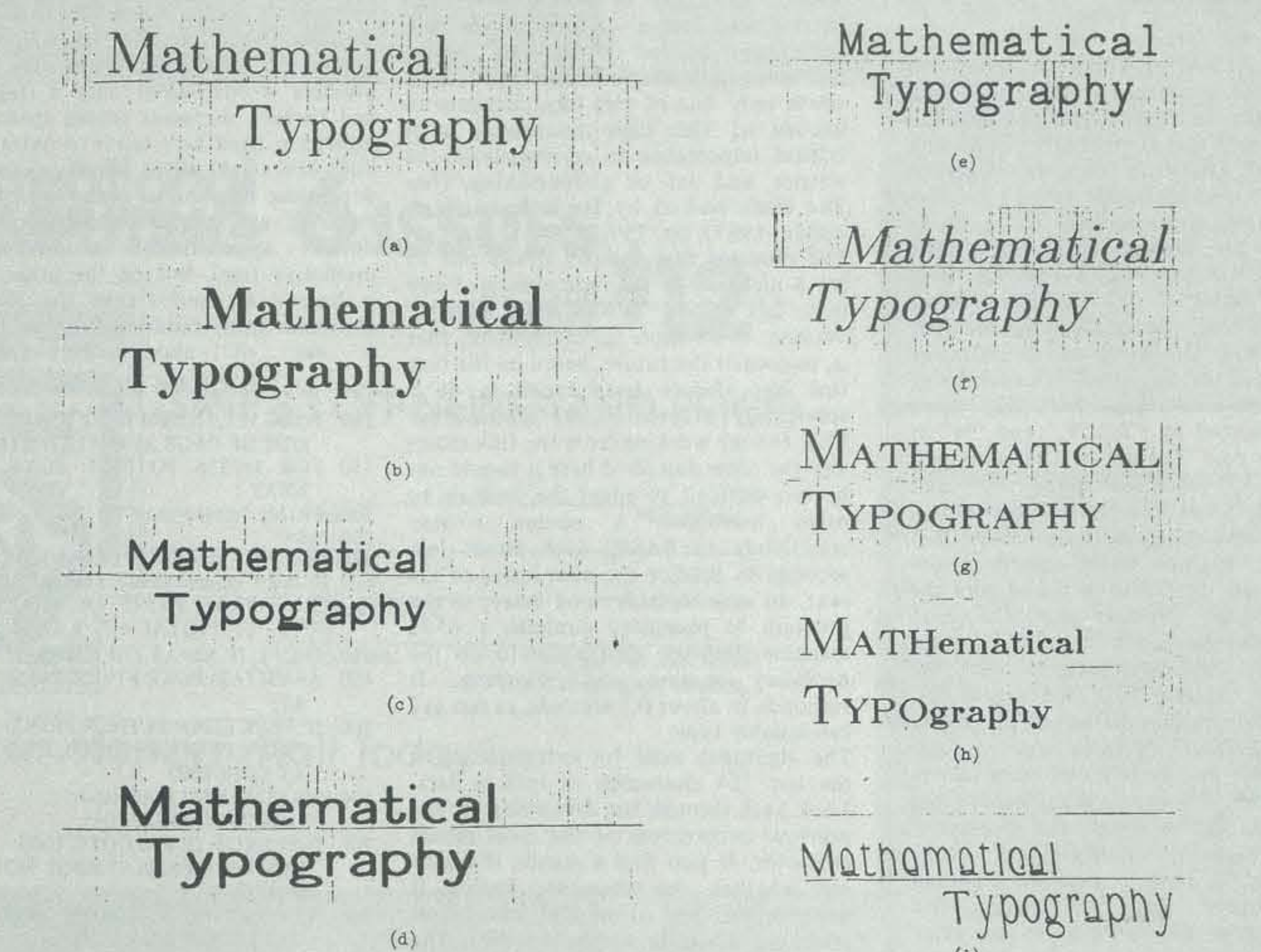


Figure 3.
Different styles of type obtained by varying the parameters to METAFONT:
(a) Computer Modern Roman; (b) Computer Modern Bold; (c) Computer Modern Sans-serif; (d) Computer Modern Sans-serif Bold; (e) Computer Modern Typewriter; (f) Computer Modern Slanted Roman; (g) Computer Modern Roman with small caps; (h) Computer Modern Roman with small caps and "small lower case"; (i) Computer Modern Funny.

Textrapolation

BY MARK ZIMMERMANN

Concerning the algorithm used in "Textrapolation":

The program TEXTRAPOLATION attempts to extrapolate on text recently typed into it, in order to guess what is coming up next. It does this by following a very simple algorithm, one which will work for any language input. In order to guess the next character, TEXTRAPOLATION looks at the latest characters typed, and then scans backwards trying to find the "best" match among the preceding text. (Actually, only the preceding 255 characters are saved.) For example, suppose that the input up to the present has been: "THE PROGRAM TEXTRAPOLATION ATTEMPTS TO EXTRA." What will the program guess for the next letter? The most recent character typed in was "A." Looking back, one sees that "A" has been followed by "T," "P," and "M" in the past. The last two letters typed, "RA," occurred before "P" and "M," and the last three letters, "TRA," occurred only before "P." So, a three letter match is the "best" that can be found, and the program predicts that the next letter input will be a "P." (Actually, this match extends back for five letters; "EXTRA" occurred before, within the word "TEXTRAPOLATION.") In the case of a tie, where two matches of equal length occur, the most recent occurrence wins. Spaces count, and upper-case and lower-case letters are distinguished. Thus, if a period is always followed by two spaces, the program will catch on to that after the first time and never get it wrong again.

English is quite a redundant language. One mathematical way to measure that redundancy is to take a message up to a certain point and then to try to guess what comes next. For example, in the previous sentence, if one just looked at the fragment "One mathem," most English speakers would have no difficulty guessing that the next few letters were "a," then "t," then "i," etc.

In fact, tests have shown that a person can guess about *75%* correctly for text written using the 26 letters plus punctuation and spaces – a surprisingly high percentage. Instead of requiring five or six bits per character, most text really needs only one of two bits/character to encode it! This high redundancy is of critical importance in cryptanalysis, the science and art of codebreaking. (See *The Codebreakers* by David Kahn (Macmillan, 1967), pp. 759-762.)

The program that follows allows one to get a measure of the redundancy of any text. It's called "TEXTRAPOLATION" because it attempts to extrapolate, that is, to predict the future, based on the text that has already been typed in. It is specifically adapted to the Commodore PET, though working from the flowcharts and the ideas described here it should not be too difficult to adapt the program to other machines. A version written completely in BASIC took about two seconds to predict the next letter of the text, an unacceptably long delay, so the program as presented contains a 6502 machine-language subroutine to do the necessary sorting and searching. It responds in about 0.1 seconds, as fast as I can reliably type.

The algorithm used for extrapolation is the last 254 characters of text as data. Look back through the data string for the previous occurrences of the most recent character. If you find a match, check to see whether the character before it matches also, and so on. Find the longest matching substring within the data. The predicted character is the character that followed that substring.

An example may make that clearer. Take the data string

ABABCBCABACABCB . . .

What character do you predict comes next? Looking back for matches with the last character, "B," one finds five possibilities, but only one of those matches the next-to-last character "C" (and in addition, matches the last four characters). The matching sections are underlined here: ABABCBCABACABCB . . . Since the character following the best previous matching substring is "C," the program guesses that "C" will be next in the input text. In the data string "FOURSCORE AND SEVEN YEARS," the best matching substring is the underlined "RS" in "FOURSCORE," so that program predicts "C" (instead of a space).

Clearly the input need not be in English; the program as written starts out completely ignorant and learns as it receives text. Small changes in scorekeeping and output (like suppression of the computer's guess until *after* the next symbol is typed) would make this extrapolation scheme into a good strategy for playing "rock-paper-scissors" or "odd-even" or other such games where predicting the opponent's choice is essential to winning. Perhaps it could also be adapted to making an "intelligent" typewriter, which predicts words, given only a fragment, and thereby increases typing speed. The algorithm used here tends to extrapolate 20%-30% of the input correctly, depending on how often words are repeated and other stylistic factors. It doesn't approach the human's 75% predictive level, but on the other hand, it does a *lot* better than the 3% level expected from random guessing out of a set of about 30 symbols!

PROGRAM LISTING

```
100  POKE 135,31:REM KEEP BASIC
       OUT OF PAGE 31 IN PET
150  FOR I=7936 TO 8191: POKE I,0:
       NEXT I
200  PRINT "clear screen"
250  A$=""
300  GET A$ : IF A$=" " GOTO 300
320  IF A$="reverse-field" THEN PRINT:
     PRINT:PRINT:PRINT R; "RIGHT
     OF"; C; "TOTAL = "; R/C: GOTO250
340  C=C+1: IF N$=A$ THEN R=R+1
400  A=ASC(A$):POKE 8191,A:PRINT
       A$;
500  IF PEEK (226)=39 THEN PRINT:
       PRINT:PRINT:REM SPACE DOWN
       AT LINE END
600 SYS (826):REM MACHINE
       LANGUAGE ROUTINE
700  P=PEEK(2): IF P=0 GOTO 1000:
       REM P=0 MEANS SYMBOL NOT
       FOUND
800  N$=CHR$ (PEEK(7936+P)): REM
       GET PREDICTION
900  PRINT "down, reverse-field";N$;"off-
       reverse-field, back, up";: GOTO 250
1000 PRINT "down, null-symbol, back, up"
       ;:N$="":GOTO 250
```

Note: items in lower-case within quotes refer to PET cursor control character.

(continued from page 27)

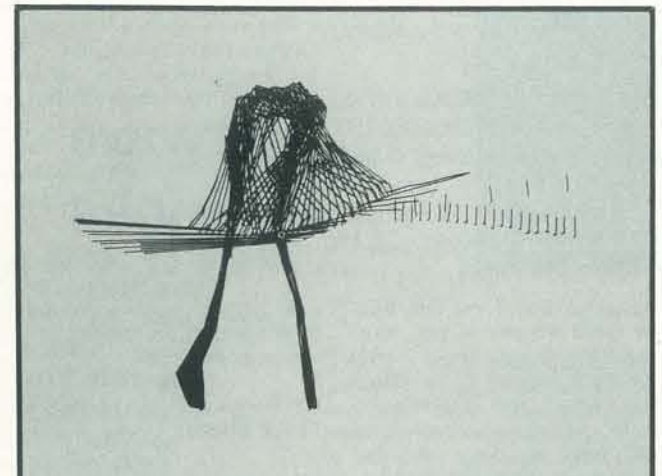


Figure 6. Jimmy Connors' Tennis Backhand

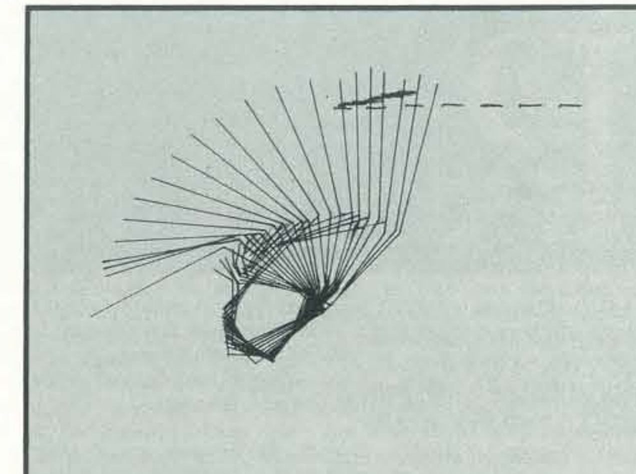
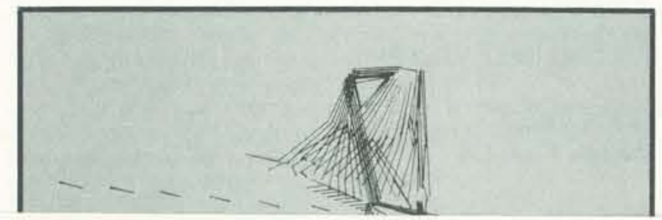
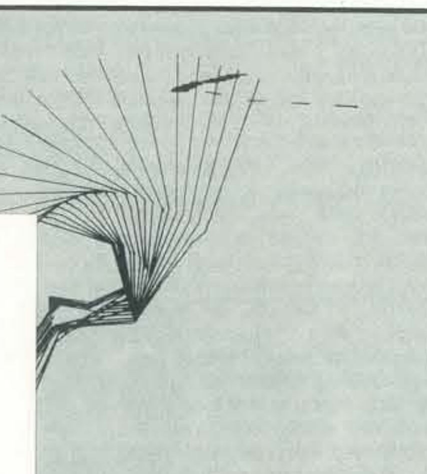


Figure 7. Chris Evert's Tennis Backhand (topview)





...stase's Tennis Backhand (topview)

Figure 10. Jack Nicklaus' Golf Drive

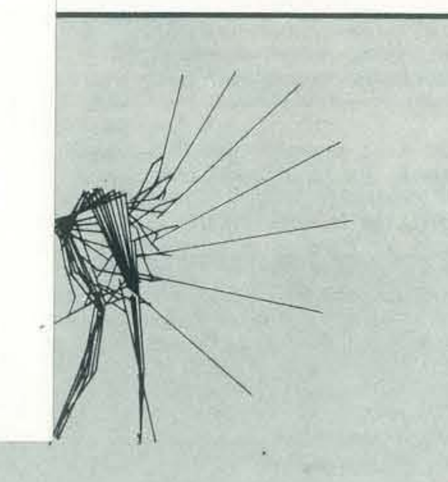


Figure 11. Gerald R. Ford's Golf Drive

Textrapolation

BY MARK ZIMMERMANN

Concerning the algorithm used in "Textrapolation":

The program TEXTRAPOLATION attempts to extrapolate on text recently typed into it, in order to guess what is coming up next. It does this by following a very simple algorithm, one which will work for any language input. In order to guess the next character, TEXTRAPOLATION looks at the latest characters typed, and then scans backwards trying to find the "best" match among the preceding text. (Actually, only the preceding 255 characters are saved.) For example, suppose that the input up to the present has been: "THE PROGRAM TEXTRAPOLATION ATTEMPTS TO EXTRA." What will the program guess for the next letter? The most recent character typed in was "A." Looking back, one sees that "A" has been followed by "T," "P," and "M" in the past. The last two letters typed, "RA," occurred before "P" and "M," and the last three letters, "TRA," occurred only before "P." So, a three letter match is the "best" that can be found, and the program predicts that the next letter input will be a "P." (Actually, this match extends back for five letters; "EXTRA" occurred before, within the word "TEXTRAPOLATION.") In the case of a tie, where two matches of equal length occur, the most recent occurrence wins. Spaces count, and upper-case and lower-case letters are distinguished. Thus, if a period is always followed by two spaces, the program will catch on to that after the first time and never get it wrong again.

English is quite a redundant language. One mathematical way to measure that redundancy is to take a message up to a certain point and then to try to guess what comes next. For example, in the previous sentence, if one just looked at the fragment "One mathem," most English speakers would have no difficulty guessing that the next few letters were "a," then "t," then "i," etc.

In fact, tests have shown that a person can guess about *75%* correctly for text written using the 26 letters plus punctuation and spaces – a surprisingly high percentage. Instead of requiring five or six bits per character, most text really

matches also, and so on. Find the longest matching substring within the data. The predicted character is the character that followed that substring.

An example may make that clearer. Take the data string

ABABCBCABACABCB . . .

What character do you predict comes next? Looking back for matches with the last character, "B," one finds five possibilities, but only one of those matches the next-to-last character "C" (and in addition, matches the last four characters). The matching sections are underlined here: ABABCBCABACABCB . . . Since the character following the best previous matching substring is "C," the program guesses that "C" will be next in the input text. In the data string "FOURSCORE AND SEVEN YEARS," the best matching substring is the underlined "RS" in "FOURSCORE," so that program predicts "C" (instead of a space).

Clearly the input need not be in English; the program as written starts out completely ignorant and learns as it receives text. Small changes in scorekeeping and output (like suppression of the computer's guess until *after* the next symbol is typed) would make this extrapolation scheme into a good strategy for playing "rock-paper-scissors" or "odd-even" or other such games where predicting the opponent's choice is essential to winning. Perhaps it could also be adapted to making an "intelligent" typewriter, which

```
      FOUND
800  N$=CHR$ (PEEK(7936+P)): REM
       GET PREDICTION
900  PRINT "down, reverse-field";N$;"off-
       reverse-field, back, up";: GOTO 250
1000 PRINT "down, null-symbol, back, up"
       ;:N$="":GOTO 250
```

Note: items in lower-case within quotes refer to PET cursor control character.

(continued from page 27)

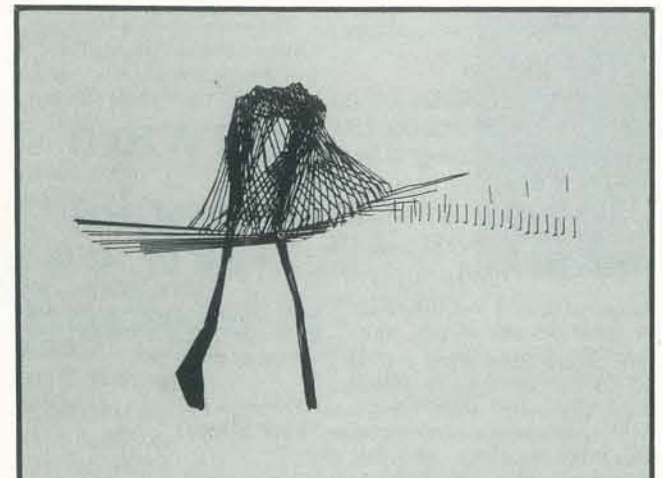


Figure 6. Jimmy Connors' Tennis Backhand

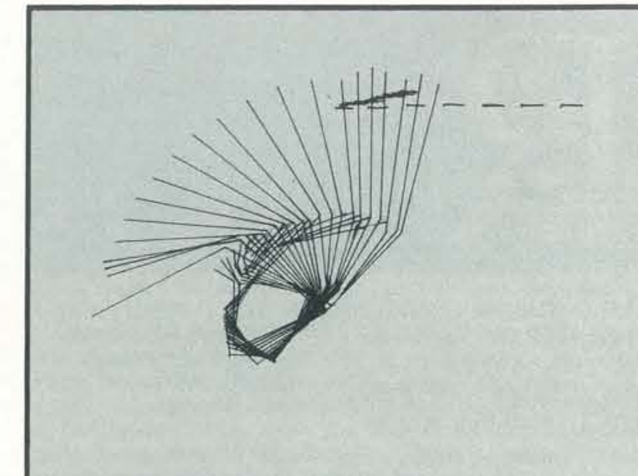


Figure 7. Chris Evert's Tennis Backhand (topview)

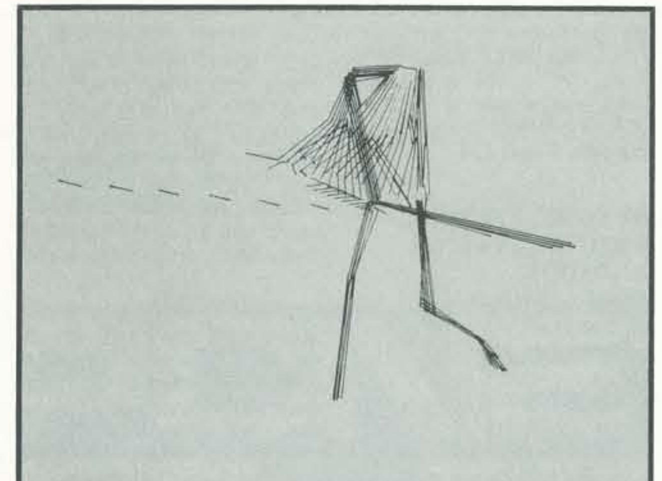


Figure 8. Chris Evert's Tennis Backhand (sideview)

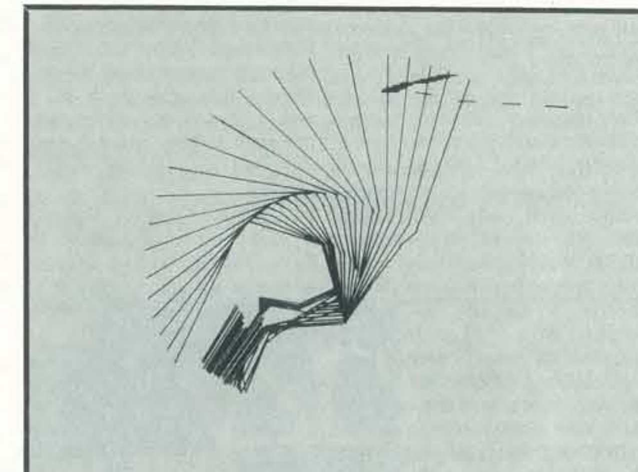


Figure 9. Elie Nastase's Tennis Backhand (topview)

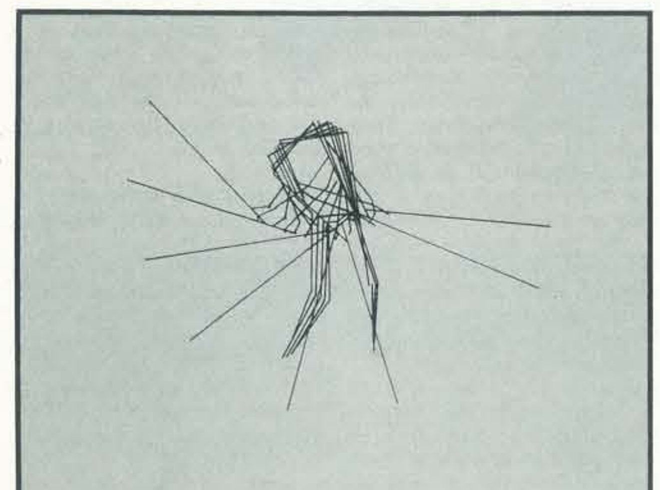


Figure 10. Jack Nicklaus' Golf Drive

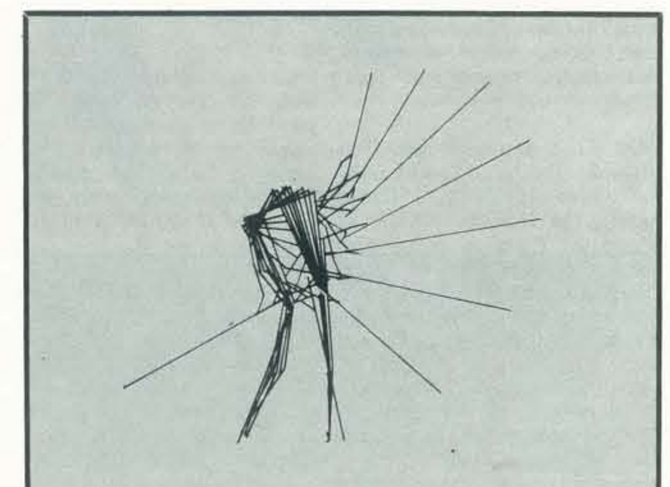


Figure 11. Gerald R. Ford's Golf Drive

REVIEWS

Software & Books

RADIO SHACK COMMUNICATIONS PACKAGE #26-1146 $29.95
PROGRAMMA INTERNATIONAL TIELINE $24.95
3400 Wilshire Blvd., Los Angeles, CA 90010
A.C. S. S. DOS 3.0 COMMUNICATIONS FEATURE $49.95
Computer World
625 Main St., Nashville, TN 37206

After spending piles of money to obtain stand-alone computing capability, why go to more trouble just to talk to other computers? Well, for one thing, there are dozens of computer bulletin boards out there where some interesting information is available. There are also several computer data bases accessible by phone, offering everything from newsheadlines to scientific journal searches. Some timesharing services make available FORTRAN, COBOL and other languages that are not readily available to micros. One big selling point to me was the possibility of program transfer between machines with different storage media. I dumped some old paper tape pgrograms into the local timeshare, then milked them out to disk storage on my 80. Beats typing them in again from listings.

DOS 3.0 I purchased this DOS primarily for its advertised communication capability. Unfortunately, the documentation is so sparse, or I am so dense, that I was never able to go on line in the communicaton mode. I have seen comments on several bulletin boards indicating I am not alone in my failure. There is an enhanced documentation package in preparation, so perhaps someday I will get to use some of the promised goodies.

RADIO SHACK COMMUNICATIONS PACKAGE PROGRAMMA TIELINE Both of these packages are readily-loaded, menu-driven operations permitting such desirable functions as on-line baud rate changes, terminal operation, program and file exchange, and remote operation of your computer.

Documentation on both programs is easy to follow, and both seem to perform all advertised functions. The PROGRAMMA program seems to have a smoother "flow" to it, and it has become my standard, primarily because the on-screen prompting of TIELINE reduces the need to reference paper documentation. When you are on line with both Ma Bell and a timeshare service, several minutes leafing through an instruction manual can make you nervous.

If you have an expansion interface, RS232, and a modem, a whole new world waits out there for you.

The Old Soldier
Mountain View, CA

DISK DRIVE TIMER (DDT)
Tape $14.95, Disk $19.95
Morton Technology Inc.,
Box 11129, Santa Rosa, CA 95406. Available for TRS-80 and Apple.

I have had my share of problems with my TRS-80 disk drives. Two out of three needed swopping within the warranty period. Annoying for me, but disastrous for the one-drive system. When I saw the ad for DDT, I jumped at it.

The tape version (TRS-80 only) loaded the first try, unusal in itself, and the menu came up immediately. I was given the choice of checking speed of all drives, checking speed of a single drive, or a graphic display of a single drive speed to guide adjustment.

The directions for speed adjustment were adequate, but the problems inherent in getting to the potentiometer to make those adjustments may deter some who fear dismantling. I was able to adjust speed on one of my more intractible drives, but I would have been happier if the adjustment were more accessible. I would, however, recommend DDT even for those who fear screwdrivers, since the ability to diagnose a problem before sending a unit in for repairs, or even the ability to remove blame from disk drives can be valuable tools of computer system management.

DDT. Don't leave home without it.

The Old Soldier
Mountain View, CA

THE COMPLETE MICROCOMPUTER SYSTEMS HANDBOOK
Edward L. Safford, Jr.
Tab Books
$9.95, 324 pages.

THE COMPLETE MICROCOMPUTER SYSTEMS HANDBOOK is a general guide to microcomputers and how to use them, and is probably best employed as a reference manual for the novice. There are discussions on an array of topics, from programming concepts to electronic and mechanical diagnostics, as well as definitions and general explanations of microcomputer terminology covering the spectrum from the basics to bubble memory. Background material is included wherever appropriate. It is unfortunate that the usefulness of the many photographs is diminished by the poor quality of reproduction, but the illustrations and drawings are very informative. A detailed table of contents and the index are important aids in finding specific topics.

The microcomputer field is a very broad subject. To touch upon so many parts of it required some sacrifice in depth of coverage. However, the *HANDBOOK* supplies enough information on enough facets of microcomputers to help the beginner know what questions to ask. It could be a great help in defining the specific needs of a prospective microcomputer purchaser.

Reviewed by,
Susan Bowers

THE MOST POPULAR SUBROUTINES IN BASIC
Ken Tracton
Tab Books
$5.95 184 pages

THE MOST POPULAR SUBROUTINES IN BASIC is a collection of well over 100 short subroutines in 15 categories, designed for use as "building modules" in BASIC programming. Most are less than ten lines in length and all are presented in a businesslike, no-nonsense manner.

Chapter One briefly explains what subroutines are and how to use them. The subroutines themselves are found in the next fifteen chapters. Included in the categories are annuities, conversions, graphs, mathematics, physics, sequencing and trajectories. The chapter on conversions is especially good, with 52 frequently used conversions in chemistry, physics, mathematics and mechanics. Those most useful to electronic hobbyists are given special attention. Chapter Seventeen provides programming examples using some of the subroutines, and is followed by a short but complete index.

Each routine is accompanied by documentation which includes a statement of its purpose, variables needed, variables altered, variables returned, and equations used, in addition to the source code and a test run. When explanations are necessary they are clear and very concise. Equations are defined so that programmers with little mathematics may still understand them.

Inexperienced hobbyists, especially those without extensive mathematics background, may find *THE MOST POPULAR SUBROUTINES IN BASIC* to be exactly what they have needed to help them become more efficient programmers. More experienced and professional programmers will also find it very helpful.

Reviewed by,
Susan Bowers–University of Wisconsin

THE GUIDE TO SIMULATIONS/GAMES FOR EDUCATION & TRAINING 4th Edition.
Robert E. Horn and Anne Cleves (eds)
Sage Publications,
275 Beverly Drive,
Beverly Hills CA 90212
$49.95 692 pages

A serious, educator-oriented compendium of tutorial articles, resource listings, and capsule descriptions of simulation games. There's information here of interest and use to the gamesmith, but very little of it relates directly to computer games. They don't mention Dungeons & Dragons or any of the other fantasy games either.

Reviewed by,
Dennis Allison

KATIE AND THE COMPUTER
Fred D'Ignazio
Illustrated by Stan Gilliam
Creative Computing Press
PO Box 789-M,
Morristown N. J. 07960
36 pages

This is a kid's book written for parents. There's an allegorical storyline where characters who personify the various functions of a computer act out for Katie how the computer works. The whole effect is rather strained and uncompelling. When I tried it out on my daughters, they were left confused and bored. It would still have redeeming value had it amused me, the parent: it did not. Daughter Katie appreciates it, though it sits on her shelf unread; it is nice to have *your* name up there in lights.

Reviewed by,
Dennis Allison

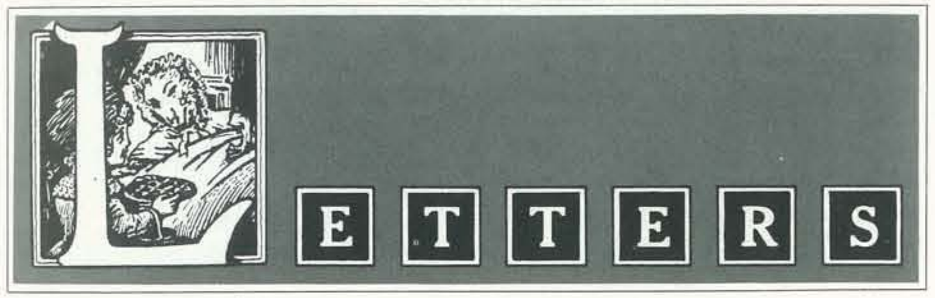
THE DELPHI CALCULUS
Maury Green
Dell Books
$2.25

Advertized as a book about a supercomputer with "millions of microchips" which make it all-powerful and give it the potential to be the greatest blackmailer of all time, this novel provides recreational reading for the computer buff. An intriguing concept of computer control of the U.S. forms the basis of this light, but entertaining, story. Set during the Presidential campaign of 1980 and the first months of the new President's tenure in 1981, this story is timely, to say the least.

In the story, a "think tank" determines that the construction and covert use of a massive computing system provides the only way to assure the security of the U.S. and convinces the Presidential front-runner that it should be implemented in secrecy. The genius who develops the system falls victim to his own love of power and creates a very credible blackmail scheme using Delphi, the computer, to control other computer sytsems and, through them, transportation, finance, communications, in order to threaten the continued existence of the United States.

Although people with characters drawn from recent history, Watergate, the CIA, national politics, etc.) and containing the obligatory racial and sexual vignettes, this is an enjoyable little book that can help pass the time for anyone even fainlty interested in computers. The threat may well be real, but for the more serious reader, Alan Drury in his counterpoint novels *Promise of Joy* and *Come Ninevah, Come Tyre* presents in much greater depth a far more literary examination of potential national and international political threats. Still, it is a good sign to see "microchips" on a popular novel's cover–maybe the world is beginning to learn about us after all! You'll probably see a lot of this litte book in airports, train stations, and on the subway this year.

Reviewed by,
Harold Kinne, Ph.D.,



MORE ABOUT WIZARD'S CASTLE

If any of you were ADVENTUREous enough to try adapting Wizard's Castle II for your system, you have realized that this is a non-trivial task. This follow-up is meant to make the job a little easier by giving you more information about the program. If you have any questions, feel free to write (I enjoy mail). Enclose a self-addressed stamped envelope and tell me about yourself (age, interests, equipment owned, etc).

The biggest problems people are going to have will all deal with that bizarre line 0. The reason that is in the program is because the Exidy Sorcerer's BASIC has no RANDOM command. For machines that have such a command, delete the REM, the POKEs to 260 and 261, and the USR call. For Sorcerer owners, type the line as 0 REM----------. Leave BASIC with the BYE command and ENter the following at 01DA: ED 5F 28 FC 32 FF F7 C9 / /. Return to BASIC and the line is as it should be. This machine language subroutine moves the contents of the Z-80's refresh register to the last location on the screen via the accumulator. For non Z-80 owners, you will have to come up with some other method of randomizing the initial seed.

Another source of grief for a lot of people is going to be the extensive use that was made of string arrays. If your BASIC has long strings instead of string arrays some major changes must be made.. Your best bet would be to write either Kevin Williams, 3250 Vermont S.W., Grandville, MI 49418 or Dana Kaempen, 46 W. 56th St., Westmont, IL 60559 because they have both managed to get versions running on HP systems that have the long strings.

To make life as easy as possible, I am also going to make the following offer: Send me a blank tape, a SASE with about 80 cents postage, and $1.00 and I will make you a copy of either the Sorcerer or TRS-80 version (instructions not included).

Best wishes.

Joseph R. Power,
124 Cedar Street #5
E. Lansing, MI 48823

LEARN WITH THIS

Good morning!

Here is an interesting little diversion that my class in beginning programming enjoyed. I have not seen it published anywhere, so it might even be original!

1. Begin a sequence by choosing any two integers 0 thru 9.
2. Add the two integers.
3. If the sum is less than 10, write the sum as the next number of the sequence; otherwise, add the 2 digits of the sum and write that total as the next number of the sequence.
4. Add the new number to the number preceding it in the sequence.
5. Go to step 3.

Example: Start with 3 and 6. Add them and get 9. The sequence so far is 3, 6, 9. Add 6 and 9 to get 15. Since this is more than 10, we add the 1 and the 5 to get 6. The sequence now is 3, 6, 9, 6. Add the 9 and 6 to get 15, which once again becomes 6 so that the sequence now is 3, 6, 9, 6, 6. Continuing in this manner we find that the complete sequence is 3, 6, 9, 6, 6, 3, 9, 3, 3, 6 . . . and it has begun to repeat.

Questions:

1. Try two other starting numbers. Do they give a sequence that repeats?
2. Would *any* two numbers eventually give a repeating sequence? Can you prove it?
3. Write a computer program to carry out this process.

Some comments on the process are listed below, published upside down so as not to spoil the fun.

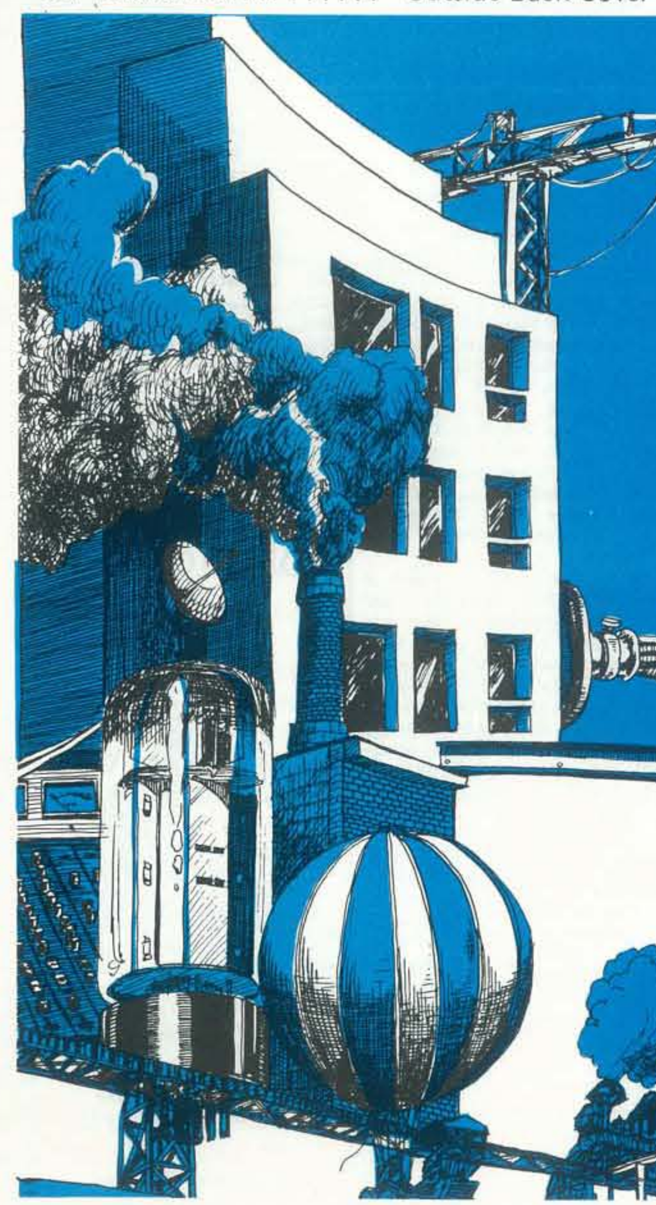
1. There are 81 possible combinations of 2 single-digit numbers that can be used to start the sequence—both numbers may be the same, as 5, 5, . . .
2. Every choice of 2 numbers leads to a repeating sequence. The shortest sequence is the one that begins 9, 9, . . . There are three sequences that have 24 terms before repeating and one (which was used in the example) that repeats on the 9th term.
3. The program is quite simple, but can be fancied up with all sorts of interesting ways to print the results.

Sincerely,
Douglas E. Scott

A Plea to Cryptarithms Fans

For those of you who have enjoyed the many mathematical puzzles that have appeared in Jack (Nine Hex) Crehore's column, it is obvious that he has spent much time and effort in producing them. Now Jack needs some of your time and effort. Jack would appreciate someone lending him a helping hand by typing and organizing his material. If you can spare some much needed time, please contact him at P.O. Box 96, Charlotte Court House, Virginia 23923, Tel. 804-542-5930.

ADVERTISING INDEX



ANNOUNCEMENTS

Unless the U.S. Exploits its advantages in computer software and higher education, the 1980s may be remembered as the decade when technological progress was "invented in America (but) made in Japan," warns William F. Miller, Stanford computer scientist and president of SRI International, an independent nonprofit firm in nearby Menlo Park.

While Japan already has 17,000 of the 22,000 working industrial robots in the world, the U.S. remains supreme in the sophisticated software concepts needed for the extensive application of robotics, Miller said. The same holds true in computer-based text editing and grpahics, as well as the design automation necessary for very large systems integration (VLSI).

American universities produce "innovative and concept-oriented graduates" in computer science and engineering, a feature the Japanese hope to import, Miller adds. Moreover, American industries remain more receptive to such graduates than the Japanese or German systems "although they certainly will change over time."

But America should not underestimate its foreign competitors. "We are in a global competition," Miller warns. "Where the Japanese are playing a global game, we are playing a national one . . ."

"We can no longer structure our processes, our regulations, our approach to the (relations) between government and business as though we had only to deal with the American marketplace."

Where the U.S. had more than half the world's productive capability at the end of World War II, today its share is about 25 percent. Without computer science, this country would have been in "perpetual recession" during the past decade, Miller said.

America should "be eclectic" in selecting the best ideas from other countries as well as our own, he suggests. Among the key areas:

Human Resource Development: The Japanese emphasize life-long employment, retaining and improving the expertise of employees in a single firm, and providing managers with a variety of opportunities to work with government officials.

Capital Structures: These should be changed to favor innovation and technological development.

Business-Government Relations: Given global competition, this should be marked by greater cooperation and a less "adversarial" attitude.

Strategies Planning: "The most important single factor in Japanese success is the fact that they have a national strategies plan," Miller says.

A Month-Old Baby is being tended by a computer called ORCA III in London. ORCA soothes the baby when she cries, tells her bedtime stories and will teach her English, French and German when she starts to talk. ORCA can also do household chores: turn on lights and open the garage door, for instance, as well as guarding the house against thieves. Starting with an ITT 2020, 28-year-old Richard Zawadski expanded the computer's capabilities to include speech synthesis and voice recognition. He and his wife, an engineer, rewired the house and set up microphones in the nursery. The computer switches on the instant the baby starts to cry, and talks back in a soothing tone which reproduces both his and his wife's voice. ORCA makes up bedtime stories which start with "Once upon a time," and end up with "They lived happily ever after."

The Terrapin Turtle is now available with a plugable interface from any standard RS-232 port to the Turtle. That means they can be run from nearly any computer or even remotely via modem and acoustic coupler. Initial pricing will be $150 for the interface or $125 with a turtle. For more information contact Terrapin, Inc., 678 Massachusetts Ave. #205, Cambridge MA 02139 (617) 482-1033.

Lightpen for APPLE II is a self-contained light pen which plugs directly into the Apple. Other versions are available for the PET and the TRS-80. Using the light pen, programs can use menu selection to acquire data. The 3-G light pen is completely assembled and ready to plug into the Apple game paddle port. A demonstration game cassette, sample program, and complete instructions are included with the pen. The cost is $32.95 plus $1.50 US postage ($6 foreign) from 3-G Company, Incorporated, Rt. 3 Box 28A, Gaston OR 97009. (503) 662-4492.

The PET Toolkit, a PET compatible printer, and a large keyboard are but a few of the PET support hardware items available from Skyles Electric Works, 231 E. South Whisman Road, Mountain View, CA 94041. (415) 965-1735.

The RESET E-X-T-E-N-D-E-R (tm) is a device that will help TRS-80 (tm) owners who are having trouble accessing the little RESET button in the back of the keyboard after adding an Expansion Interface. Most TRS-80 (tm) users are currently sticking a pencil through the hole in the Expansion Interface connector hood to get to the RESET button. A better solution is the RESET E-X-T-E-N-D-E-R, complete with push button, spring, grommet, rivet and instructions, is available for $3.99 from Emmanuel B. Garcia, Jr. & Associates –203 N. Wabash, Rm. 2102, Chicago, Illinois 60601 (312) 782-9750.

Charles Mann & Associates has announced the release of a new programmable Business Data Base System for the TRS-80 computer. The system allows the user to define and build data bases for such purposes as inventory control, general ledger accounting, accounts receivable and accounts payable. The cost of the sytem is $89.95. Additional information is available from CM-&A, 7594 San Remo Trail, Yucca Valley, CA 92284; 714-365-9718.

TRS-80 LEVEL II BASIC, a self-teaching guide for learning to program and use a Level II TRS-80 microcomputer system, is now available from Radio Shack. According to Radio Shack, no matter what your level of experience with computers is, this 351-page manual includes all the information needed to make the computer work for you. The book is written by Bob Albrecht, Don Inman and Ramon Zamora (Bob and Ramon are former *RC* editors, and all three are frequent contributors). Priced at $9.95, the book is available at all Radio Shack dealers.

ComputerTown, USA! began in April 1979. Bob Albrecht and Ramon Zamora started the project by going to local places such as a pizza parlor and a popular bookstore a few nights each month. They brought along their personal computers, a bit of software and magazines, and made the equipment available to anybody who was interested. ComputerTown USA! has given more than 1000 people—kids and adults—the opportunity to use a microcomputer, has been the subject of many news stories and magazine articles, and is spreading across the land. If you would like more information about starting a ComputerTown in your community, write to ComputerTown USA!, P.O. Box E, Menlo Park, CA 94025, and include a stamped, self-addressed envelope.

Teaching Programming is a newly-started newsletter directed at teachers of computer programming. The subscription fee is $8 for 12 issues (one year). If you want to see it first, the editor, Craig Nansen, will send you an issue free. Write to Craig Nansen, 1112 Glacial Drive, Minot, ND 58701.

JAX Area PET Society offers PET software at $1.50 per program. Send a stamped, self-addressed envelope to them for a list of what they have available. Write: PET Library, 401 Monument Road, Jacksonville, FL 32211.

MOUNTAIN COMPUTER INC. announces Music System for Apple II computers. this 16 voice digital synthesizer is said to set new standards for computer generated music. Its capabilities permit the creation of the sounds of real musical instruments utilizing the principle of additive synthesis. The generation of sounds is accomplished through fully programmable waveforms, envelopes, and amplitudes for each musical "voice." Provided with the hardware system is software for editing and playing of musical compositions. The Editor program permits graphical input of sheet music utilizing standard music notation. The Player program permits polyphonic performance of musical compositions. Stereo output is to users' stereo amplifier and speakers, or directly off card with stereo headphones. For further information, contact: Mountain Computer Inc., 300 Harvey West Blvd., Santa Cruz, CA 95060 Phone (408) 429-8600

BACK ISSUES

PCC NEWSPAPERS

Half Off Sale!
All 12 available back issues of PEOPLE'S COMPUTER COMPANY NEWSPAPER for only $12.

- #3 **May-June 1973** ZINGO/"Going to Bid"/How to Buy an Edusystem/BASIC MUSIC CHOMP. $2.00
- #10 **July-Aug 1974** A Fantasy of Future Forms/Low-Cost Home School Microprocessor Systems/INCHWORM/ MAZE/Analysis of Teaser/Computer Illiteracy Problem by Peter Lykos. $2.00
- #13 **Jan-Feb 1975** Build your own BASIC/Mysterious RND/More INCHWORM/Computers & Art/BIOSIN Son of Hardware. $2.00
- #18 **Nov-Dec 1975** NUMBER/Tiny BASIC/Analog to Digital Conversion/Minuet in G Major. $2.00
- #19 **Jan-Feb 1976** Space Games/Tiny BASIC Soloworks/ DIDDLE/STTR/More Music. $2.00
- #20 **Mar-Apr 1976** Games for the Pocket Calculator/ SQUARES/Make Believe Computers/STRTRK/ Huntington Computer Project. $2.00
- #21 **May-June 1976** Biofeedback/POUNCE/"We're Building Our Own Computer"/Programmer's Toolbox/CCC/TV as Terminal and Game Center. $2.00
- #22 **July-Aug 1976** Low Cost Software/Tiny BASIC, Tiny Trek/Your Brain is a Hologram. $2.00
- #23 **Sept-Oct 1976** Dungeons & Dragons/HATS/One on One/PLANETS/The Positive of Power Thinking $2.00
- #24 **Nov-Dec 1976** STORY/SNAKE/More Build Your Own Computer/Introducing PILOT/FROGS. $2.00
- #25 **Jan-Feb 1977** REVERSE/Robots/Tiny PILOT/ Space & Computers/Conversational Programming. $2.00
- #26 **Mar-Apr 1977** Z-80 PILOT/6502 Assembly Programming/Tiny BASIC for Beginners. $2.00

PEOPLE'S COMPUTERS

- #27 **May-June 1977** Graphics, Computers & Copyright Laws, PILOT & Tiny BASIC, Home Computers for Beginners, Women & Computers. $2.50
- #28 **July-August 1977** Heathkit's 8080 & LSI-11. Do-It-Yourself CAI, Robot PETs. $2.50
- #29 **Sept-Oct 1977** Computer Networks, the $595 PET, More Tiny Languages, $2.50
- #30 **Nov-Dec 1977** Oz Graphics, Bio-feedback & Micros, Our PET's First Steps. $2.50
- #31 **Jan-Feb 1978** Robots, Video Discs, PET Update & Programs, PASCAL & COMAL. $2.50
- #32 **March-Apr 1978** Epic Computer Games, TRS-80 Review. Micros for the Handicapped. $2.50
- #34 **July-Aug 1978** Computer Whiz Kids, Public Access to Computers, Man-made Minds. $2.50
- #35 **Sept-Oct 1978** Computer & Museums, Kingdom, APL, Sorcerer of Exidy. $2.50
- #36 **Nov-Dec 1978** The Return of the Dragons, APL, Animated Games for TRS-80. $3.00

RECREATIONAL COMPUTING

- #37 **Jan-Feb 1979** Artifical Intelligence, A fantasy by Jules Verne, TRS-80 Personal Software, The Apple Corps is with Us. $3.00
- #38 **March-Apr 1979** Artificial Intelligence, Calculator Comics, Chess Reconsidered, Lord of the Rings. $3.00
- #39 **May-June 1979** Now Peter Can Read, Game of Life, TRS-80 Programs, Detective Game Program. $3.00
- #40 **July-Aug 1979** ComputerTown, USA!, Cryptarithms, DP Games Playing, FORTE Music Programming Language. $3.00
- #42 **Nov-Dec 1979** Shogi Games, Creating New Characters on the Ti 99/4, Music for Atari, TRS-80 & PET. $3.00
- #43 **Jan-Feb 1980** Computing for Health and Equaity (PET). Ti Graphics & Animation Pt. 2, More Music for Atari. $3.00
- #44 **Mar-Apr 1980** Special Games Issue! Recreation Apple II Hi-Res Graphics, Galaxy II, TRS-80 Tiny Trek. $3.00
- #45 **May-June 1980** Intro to Computer Music, CBBS Phone Numbers, 6502 Machine Language, Five New Personal Computers. $3.00
- #46 **July-Aug 1980** Fantasy Games! Write Your Own; Wizard's Castle, Star Trek, Wonderful World of Eamon, The End of Fortran Man $3.50

Only the issues listed above are available. Price includes issue, handling, and shipment by second class or foreign surface mail. Within the U.S., please allow 6-9 weeks to process your order second class. Outside the U.S., surface mail can take 2-4 months. For faster service within the U.S. we'll ship UPS if you *add* $1.00 for 1-2 issues and $.50 for *each* issue thereafter. We need a street address, not a PO Box. Outside the U.S., *add* $1.50 *per* issue requested for airmail shipment.

Circle your choices.

3 10 13 18 19 20 21 22 23 24 25 26 27 28 29 30
31 32 34 35 36 37 38 39 40 42 43 44 45 46

Please send me the issues circled.

☐ Yes, send me #**3-26** for only **$12**

☐ I'm ordering $20 or more and I want a 10% discount.

I enclose $ ____________ (U.S. check or money order). Foreign add $.50 *per* issue (this includes postage and handling).

Please send in this form or facsimile to: Recreational Computing, 1263 El Camino Real, Box E, Menlo Park CA 94025.

My Name ____________________

Address ____________________

City/State/Zip ____________________