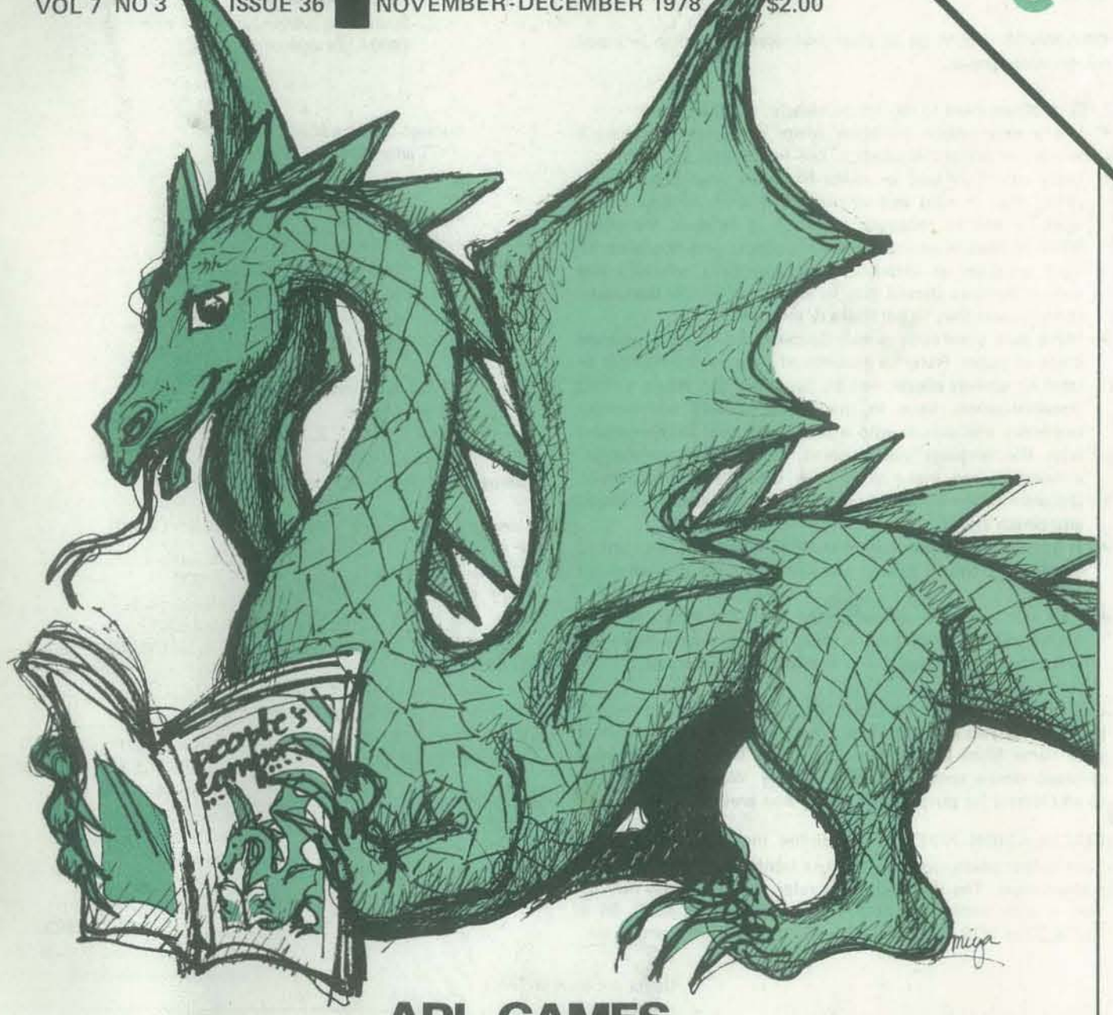
people's computers

VOL 7 NO 3 ISSUE 36 NOVEMBER-DECEMBER 1978 $2.00

SOON TO BE Recreational COMPUTING MAGAZINE



APL GAMES

THE RETURN OF THE DRAGONS!

ANIMATED GAMES FOR TRS·80

SUBMITTING ITEMS FOR PUBLICATION

LABEL everything please, your name, address and the *date*

TYPE text if at all possible, double-spaced, on 8½ x 11 inch white paper.

DRAWINGS should be as clear and neat as possible in black ink on white paper.

LISTINGS are hard to reproduce clearly, so please note:

- Use a new ribbon on plain white paper when making a listing; we prefer roll paper or fan-fold paper.
- Send copies of one or more RUNs of your program, to verify that it runs and to provide a sense of how things work – and to motivate more of us to read the code. RUNs should illustrate the main purpose and operation of your program as clearly as possible. Bells, whistles and special features should just be described in the documentation unless they're particularly relevant.
- Make sure your code is well documented – use a separate sheet of paper. Refer to portions of code by line number or label or address please, not by page number. When writing documentation, keep in mind that readers will include beginners and people who may be relatively inexperienced with the language you're using. Helpful documentation/annotation can make your code useful to more people. Documentation should discuss just which cases are covered and which aren't.
- If you send us a program to publish, we reserve the right to annotate it (don't worry, we won't publish it if we don't like it).
- Last but not least, please try to limit the width of your listings: 50-60 characters is ideal. Narrow widths mean less reduction, better readability and better use of space.

LETTERS are always welcome; we assume it's OK to publish them unless you ask us not to. Upon request we will withhold your name from a published letter, but we will not publish correspondence sent to us anonymously. We reserve the right to edit letters for purposes of clarity and brevity.

CIRCULATION NOTE: To decipher the expiration date of your subscription, look at the top right hand corner of your address label. The last two digits refer to a code/issue number that is your expiration date. Hence, read 35 as 9/78, 36 as 11/78, 37 as 1/79, 38 as 3/79, 39 as 5/79, 40 as 7/79, and so on.

SUBSCRIPTIONS

U. S. Subscriptions
- ☐ $10/1 yr. (6 issues)
- ☐ Retaining subscription @ $25 ($15 tax deductible)
- ☐ Sustaining subscription @ $100+ ($90+ tax deductible)

Foreign Surface Mail
- ☐ add $4/yr.

Foreign Airmail
- ☐ add $8/yr. for Canada
- ☐ add $11/yr. for Europe
- ☐ add $14/yr. elsewhere

Payment must be in U.S. dollars drawn on a U.S. bank.

These back issues are available at $2.50 each:
Vol 5, No 6
Vol 6, Nos 1, 2, 3, 4, 5
Vol 7, Nos 1, 2

Foreign Distributors of *People's Computers*

Vincent Coen
LP Enterprises
313 Kingston Road
Ilford IG1 1PJ
Essex, UK

Rudi Hoess
Electronic Concepts PTY Ltd
Ground Floor Cambridge House
52-58 Clarence St
Sydney NSW 2000

ASCII Publishing
305 HI TORIO
5-6-7 Minami Aoyama
Minato-Ku, Tokyo 107
JAPAN

Eastern Canada

Liz Janik
RS-232
186 Queen St W
Suite 232
Toronto ON M5V 1Z1

Western Canada

Brian Wiebe
Kitronic Audio, Digital & Electronic Components
26236 26th Ave RR5
Aldergrove
BC VOX1A0
Canada

Integrated Computer
Greenhills PO Box 483
San Juan, Metro Manila
PHILIPPINES 3113

Cover art by Ann Miya

People's Computers 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, independent, non-profit corporation, and donations are tax-deductible.
Second class postage paid at Menlo Park, California, and additional entry points.

people's computers

SOON TO BE Recreational COMPUTING

VOL 7 NO 3 NOV-DEC 1978

SPECIAL FEATURES

ARTICLES

GAMES & STUFF

DEPARTMENTS

LETTERS

EDITOR'S NOTES

Bob Kahn, your editor of the past two issues, has moved onward and upward. Bob is designing computer games and interactions for museums and other places that let people use and enjoy computers. Watch these pages for "Further Adventures of Bob Kahn."

So, the dragons have returned to PCC. In fact, three dragons—or, if you prefer, a three headed dragon. This time, Albrecht, Bradley and Zamora—editors from A to Z! (Oops, sorry about that.)

What next? Well, People's Computers, *the* oldest *home/school/personal computing periodical (once an outrageous newspaper called PCC), will soon become the* newest *periodical, called* Recreational Computing. *In this issue, you will see much evidence of the process of change to the new name and content. In this and future issues, watch for these things.*

- ★ *Serious Games: games with a purpose; games to learn by; games that teach; worlds of* IF *for people to explore.*
- ★ *Pure enjoyment: fun games, graphics, art and music.*
- ★ *Fantasy and Science Fiction: fantasy adventure games; science fiction games; whimsical odds and ends; and our swashbuckling hero—FORTRAN Man!*
- ★ *Free Software: lots of it! Old games rewritten for APPLE, PET, SOL, TRS-80 and others, featuring animation and creative use of graphics;* new *games—more than any other magazine.*
- ★ *Languages: regular stuff in APL, BASIC and PILOT, and a language forum to discuss future languages.*
- ★ *Challenges: puzzles; teasers; problems for* you *to solve; programs for* you *to write.*
- ★ *And Still More: how-to-do-it tutorials; reviews; interviews; letters; and surprises.*

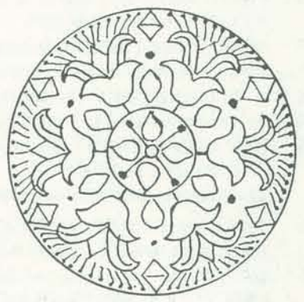
Did we say letters? Write one today to the three-headed dragon!

—Bob Albrecht
—Chuck Bradley
—Ramon Zamora

PILOT for the APPLE

Dear People,

For the past couple of months, I have been enjoying an Applesoft II version of PILOT. So the articles you've published on PILOT have been especially interesting. The version I have now (written by N. (Arley) Dealey using logic by C. Shapiro) lacks the USE and COMPUTE commands and numerical variables. But, I was able to do a pretty good job on a PILOT version of Story.



Arley is now writing an extended version of PILOT for the Apple Disc II. This will include capability to save programs on the disc. And, it uses an interpreter to link all the Applesoft BASIC commands, statements and functions to PILOT. Floating point math and graphics capability will be available to the more experienced PILOT programmers. (PILOT's structural simplicity remains intact for the neophyte programmer.)

I plan to use PILOT Programs for computer aided assembly instructions. This will be part of a more comprehensive computer aided manufacturing task. I also plan to develop PILOT programs for elementary electronics instruction. Animated Apple graphics will provide visual effectiveness for this application.

I would like to share the results of my efforts and experiences with *People's Computers* and anyone else that's interested.

C.R. (Chuck) Carpenter
2228 Montclair Pl.
Carrollton, TX 75006

Watch for more PILOT articles in future issues, especially games and educational stuff. And Chuck, how about an article from you? *– The Dragon*

PILOT LANDS IN THE OLD COUNTRY

Dear People,

Pilot has just reached us in England. To aid the introduction and development of the language the Pilot Users Group has been formed. The aim of this group is to create a library of the software available in Pilot and to distribute this to its members. It is our feeling that the language will not become popular unless there already exists a large number of packages to run in Pilot.

There is not a large amount of money available in education for this purpose in this country. The distribution of such packages should therefore be free of charge, or at least cost only a minimal amount.

To start us off, we would be interested in collecting material from the U.S.A. We could then adapt this to our own needs and build on this. We would therefore be most grateful if you would supply us with copies of anything which you have available. Perhaps you would invoice any cost to myself.

In addition to this, perhaps you could supply me with information of things available from other sources, including the commercial ones.

We hope to have at least some material ready for the start of the next school term in September. May I thank you in anticipation for your help, and I hope that eventually we may be able to cooperate on a more equal basis.

P.P. Couzens
Computer Department, Oundle School,
OUNDLE, Peterborough. PE8 4EN.
England.

The central aerodrome for PILOTs in the U.S. is: The PILOT Information Exchange, c/o Earl Keyser, 22 Clover Lane, Mason City, IA 50401, U.S.A. Cheers and good luck! – The Dragon

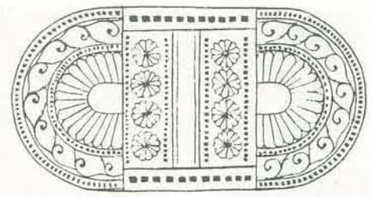
HP BASIC → TRS-80 BASIC?

Dear Sirs:

I presently own a Radio Shack TRS-80 Microcomputer, 16K RAM BASIC, Level II. I also own a copy of your excellent book, *What To Do After You Hit Return*. However, as you may know, Radio Shack Basic and Hewlett-Packard BASIC are not precisely compatible. I would like to purchase a book which explains in detail all the terminology and programming language of the particular version of BASIC contained in your book, so that I may make the necessary modifications in the book's programs.

Tindaro J. Peti
Monterey Park, CA

The games in What To Do After You Hit Return *are written in HP 2000 F BASIC which is cleverly and concisely explained in* HP 2000 F:A Guide to Time-Shared BASIC. *For info on how to get it, contact Louise Watkins, Hewlett-Packard, General Systems Division, 5303 Stevens Creek Blvd., Santa Clara CA 95050. And watch this magazine for* lots *of games for TRS-80, PET, APPLE, SOL and other home/school/personal computers. – The Dragon*



A DIFFICULT TASK FOR THE THREE HEADED DRAGON

Dear Stupids,

When I last wrote you (see copy of that letter) I accused you of not playing with a full deck. I now have revised my opinion, and I think that it's possible that I'm dealing with a band of pettifogging footpads!

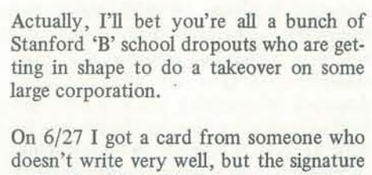
I paid for a sub last Spring, I received one issue, and now a letter telling me that my sub has expired!!

I think that the thing that lulled me into a false sense of security was your name. *People's Computer Company* brings to my mind certain things: guys with beards and acne, girls who strike ballet poses for no reason—and can't type, rolfing, getting clear, looking for one's space (I can't think of any more cliches).

Actually, I'll bet you're all a bunch of Stanford 'B' school dropouts who are getting in shape to do a takeover on some large corporation.

On 6/27 I got a card from someone who doesn't write very well, but the signature looks like Michael Mady, who told me that A) My sub was entered and would run thru Vol 8, Issue 2. and B) He was forwarding my missing issues. No doubt he is full of (bleep).

You have a fairly simple choice: send me the mag and stop harassing me with threatening letters, or return my eight bucks and go (bleep) yourselves. [*We have turned this suggestion over to the Dragon in charge of our department of unisexual reproduction.*]



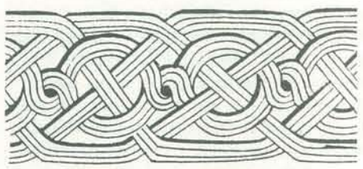
Norman Gold
73 Bennett Avenue
Huntington Station, NY 11746

I found your letter a beautiful thing to read. Your use of English and invective are superb to behold. But all in all, I am truly sorry that we caused you so much distress. I'm glad that you have enough of a sense of humor to keep laughing through your tears. I hope by now that everything is straightened out, because what you've asked us to do is very difficult. Regards.
Willard Holden.

P.S. Here are some additional cliches for you:

Doing your own thing
Getting your (bleep) together
Relating to people
Getting your head straight
Jacobsen's progressive relaxation
Finding your center
Tripping off in theta space

P.P.S. Michael Madaj adds that, in his opinion, there is reasonable doubt that he is full of (bleep).

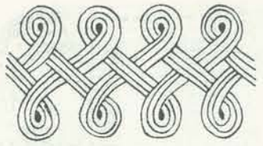


THOUGHTS ON TINY LANGUAGES

I've been thinking off and on about the Tiny Languages Extravaganza since it was announced. I am still of several minds about the whole thing, and my comments are really on several levels. However, I feel it is time to put in my two cents' worth. In order to give some pattern to my comments I will respond to the published articles in sequence.
Comments on Design Criteria:
1. *Good for Tiny BASIC and Tiny PILOT type problems.*
To me this means a basically numerical language with some string handling. So

far, so good. [PILOT is basically non-numerical with some number handling.]
2. *Most useful to elementary school kids and also useful to their parents and teachers.*
Now we have a conflict. To my mind a language for kids is not going to be basically numerical. (This is substantiated by the fact that Logo, PILOT, and Smalltalk are not basically numerical.)
3. *Can be implemented in about 4K bytes of ROM.*
This seems unnecessarily small to me. Surely memory prices have gotten to the point where we can afford to use 8K bytes [or even 16K!]. The increase should certainly enable us to have a more powerful language.

4. *Uses a color TV with simple graphics.*
Sounds good, but what exactly do we mean by 'simple graphics'? Also, if someone is paying the extra for a color TV they can surely afford the extra to have an 8K language rather than a 4K.
5. *Superficially, it looks kind of like Tiny BASIC.* Hold on a minute. Let's not restrict our options too soon. BASIC-like structure is not necessarily the way to go either syntactically or semantically. While the semantic structure of BASIC (i.e., commands) is probably the most common form, there are other forms, such as function (as in LISP) and actors (as in Smalltalk). There are other possibilities for syntax too. For example, compare BASIC, PILOT, and ALGOL. The differences are not trivial.
6. *All variables are string variables*
While this proposal has advantages in uniformity I think it loses out on the grounds of efficiency. But only if we are having a basically numerical language. The point is how do you reconcile the concept of being good for BASIC type problems with this criterion?
7. *We also have one-dimensional string arrays.*
Why stop there? We should have multi-dimensional arrays and general structures (as in C for example).



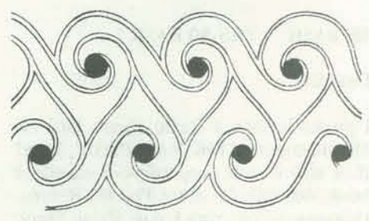
8. *A rich set of built-in functions.*
I agree, but I feel the ability to add functions is even more important.
9. *Substantially better than BASIC.*
I feel this is the most important criterion. Unless, we produce something 'better' than BASIC we might as well not begin.

Comments on Reader Feedback:
I don't want to make comments on any of the specific suggestions. I feel that starting from features is the wrong way to go about designing a language. After all, we can generate features almost ad infinitum. I think we should design in more of a top-down way by first dealing with pragmatics, then semantics, and finally syntactics. Pragmatics deals with the relations between the users and the language. This means asking questions like: "Who are going to be the users of the language?" and "What sort of things will they be trying to do?" Once we have a clear idea of what we want the language to do we can work on the semantic level. Semantics deals with the basic concepts of the language. This means deciding how we want the user to think about the things they want to do. For example, do we allow them to use unrestricted control transfers (goto's) or only more restricted concepts (if's, while's, etc.). At the last level is syntactics. Here we decide exactly how we write a program in the langauge. Admittedly, the actual design process will involve switches between levels and backtracking, rather than a straight line march.

Some Specific Suggestions:
Pragmatics: The language is to be most useful to kids, so we know who the users are. What they want to do is more difficult to establish. I would argue for a general-purpose language to avoid restricting the language to a particular age and/or interest group. The semantic style also needs to be decided. In spite of my limited experience with it, I would argue for using actor semantics, as in Smalltalk. It seems to work very nicely.
Semantics: We have to decide what our primitive actors will be. This is not easy, so I will do no more than sketch an example. The primitive for graphics could be a *pen*. It would have a location, an up/downness, a color, and possibly a direction and a width. We could then say things like:

pen at 20 30,point 0, down, do 4
(move 10, turn 90)

and draw a 10 by 10 square.
Syntactics: The actual form of a language is very important. It is also very difficult to establish a 'goodness' metric for. Consequently, I make no proposals for it.



Conclusion:
I feel a new language is a good idea. We should continually try to improve our tools for working with computers and language is perhaps the most important tool we use. But we should make a 'better' language each time. Better to stay with a bad language than to switch to a different bad language. The former may be disheartening, but the latter is idiotic.

Eryk Vershen
Stanford, California

Editor's Comment: Last time, Bob Kahn described the Exidy Sorceror™, a personal computer that will accept plug-in 16K ROM Pacs. Hmmm . . . sure would be nice to plug in a not-so-tiny language for kids—a language designed for very young children to use for games, graphics, music and other things of interest to kids.
— The Dragon

Annie & friends at the front desk.

Plenty of room to play.

MARIN COMPUTER CENTER

BY ANNIE FOX

For over a year now, David and Annie Fox have successfully proven a new concept in computer education. It's the Marin Computer Center, a non-profit organization, where anyone can walk-in and learn about computers in a safe, non-intimidating environment. Their main goal is to bring the wonders of advanced technology to the people. The Center was created to give large numbers of people easy access to computers as well as an understanding about how they work.

MCC offers direct "hands-on" experience with personal computers through programming classes, or reasonable hourly rental of the equipment for whatever you want to do. For more information write to:
Marin Computer Center
70 Skyview Terrace, Room 301
San Rafael, California 94903

This article relates some of Annie's experiences at the Center. Also, you'll find STARWARS HODGE, a game written in PILOT by David Fox. – CB

There's hardly any age barrier to learning PILOT. It's easy to use, and kids as young as 5 or 6 have learned to program this non-math oriented computer langauge.

People's Computers has run a number of articles on PILOT. A really great introduction by Phyllis Cole appears in the May-June 1977 issue of *PC*. A number of PILOT programs by Ellen Nold and Sallie Cannon of Stanford University are printed in the Sept-Oct and Nov-Dec issues of *PC*. We're so happy that you've kept all our back issues!

Processor Technology supports PILOT on their SOL and may be obtained by writing to:
Processor Technology
(415) 829-2600
PILOT 8080, Version 2.2
7100 Johnson Industrial Drive
Pleasanton, CA 94566

Dean Brown has a Z-80 version of PILOT which runs on a MCZ system. Contact:
Dean Brown
(408) 446-4666
Zilog, Inc.
10460 Bubb Road
Cupertino, CA 95014

You're also invited to join the PILOT Information Exchange. Write to Earl Keyser for information:
Earl Keyser
(515) 424-5548
22 Clover Lane
Mason City, Iowa 50401

It works this way Mommy!

The Marin Computer Center was created for the purpose of giving computer knowledge to "the people." We feel that the biggest threat of the 1984 "Big Brotherism" is not in the proliferation of computers but rather in the lack of public computer awareness. Therefore, back in September 1977, we set out to take the knowledge, which had heretofore been so carefully guarded in the laboratories and the computer science departments of universities, and spread it around. With a real sense of fun and irreverence, (which computers need so that their egos don't swell out of proportion) we have been teaching people about computers. And through our computer programming classes, "utterly baffled" people of all ages and backgrounds have discovered: (1) computers are fun to play with; (2) computers are diligent servants but rather limited in their imaginations; (3) that the frightening myths about computers, such as "computers are taking over the world; we are all their slaves" were obviously started and perpetrated by individuals who have never worked with them.

We've learned that when you put someone in front of a computer to teach them programming it really doesn't matter whether they are ten or sixty. In fact, other than touch typing skills, which can be a decided advantage, age has very little effect on the learning of "computerese". In some instances, (actually, most of them, when I stop and think about it) the kids seem to have the learning advantage. You see, thanks to "Star Trek", "Star Wars", and the consumer video games of the past few years, kids between the ages of 10 to 15 LOVE computers. They've never had the negative computer associated experiences that their parents grumble about. They've never received a computerized bill, pay check or magazine subscription. They've never had a bank error on their monthly statement, or been "tripped up" or "found out" by the IRS's computer. They've never experienced any of that kind of stuff—and in their naivete they are open and willing to learn.

When your image of a computer is a super-intelligent, anti-human, show-off machine, (as it is to one degree or another for many adults) then learning its language and operations would seem like an insurmountable task. "Oh I could never learn that! I was an English major!"

Let me tell you something. Before I got into computers I wasn't exactly a mechanical engineer! (Though I did run a nursery school which may not be all that different.) I was one of the "Artiste-Elite" who writes children's stories, plays the guitar and does astrological gardening. Do you think I ever, in my most bizarre psychic visions, pictured myself not only working with, but playing with, as well as teaching others about computers? Never! I could have more readily envisioned myself running for school board on the "Lecithin for Lunch" Ticket!

So how did this transformation of interest come about? Well, I've got to admit, it was not my idea to get into the business of computer education. It was my husband's—his and a few of his nutty friends. For three years he had been doing one-to-one counseling, and after all those hours of helping people unravel their lives, the figured he had finally learned enough about himself to quit. He wanted a change, and since he has always had an affinity for machines and tinkering with them, the idea of a community computer center was born.

David is a futurist, (he's got Aquarius Rising) and he enjoys, admires, and relishes in technology. But unlike many pro-techonology folks, he sees technology as valuable only as it relates to serving people. He saw computers, especially micro-computers as a new technology with a tremendous amount to offer in the servicing of people. And yet knowing the way most of the world feels about computers, there was a great need for public acceptance which he felt certain would come once people had an opportunity to interact with them in a "safe" environment.

And so that's the kind of environment we created—spacious and yet warm. Comfortable and calm. Even when the place is filled with people playing games and writing programs, there still reigns a certain animated serenity in the space which makes for a wonderful place to learn about something you might be slighty scared of.

What's the young computer scientist watching?

What do people do here? The Marin Computer Center has 13 microcomputers, which are the kind you've heard about as "home" or "personal" computers. For those of you who are familiar with different systems, we've got nine SOL 20's by Processor Technology (one with a Micromation Dual Floppy with CP/M and a Diablo Hyterm Printer, and one with North Star Drives); one Equinox-100 by Parasitic Engineering; one PET by Commodore; one APPLE-II; 16K Level II Radio Shack TRS-80; and an IMSAI VDP-40. We also have about 150 programs (mostly games) in our library.
So people come in and rent computer time ($2.25/hr. for non-members and $1.50/hr. for members—a membership costs $12.50 a year). If they know programming, they can use our systems to program in BASIC, ASSEMBLY, PILOT or FORTRAN. If they don't know programming and would like to learn, we offer the following classes: PLAYING WITH PILOT, BEGINNING BASIC, and BASIC II (for those who are still hungry after the first course).

Of all of the classes I've taught, the most incredible things have happened in my classes for kids—specifically in the classes that are learning PILOT. Since April, I've been teaching computer programming in PILOT to kids as young as 3rd grade and the results have been startling! PILOT is a language that does not hang kids up with a lot of syntactical barbs the way BASIC does. Instead, it is easy to manipulate and master. After one class session, I have seen 8 year old kids programming stories and letters which are beautifully formatted and centered on the screen. After three sessions they are writing fairly sophisticated interactive games. Because PILOT is so simple to understand and to use, the kids in my six-week class are moving at a velocity easily twice the speed of my earlier BASIC classes for kids. Since the computer logic in both languages is the same, I would have to attribute the difference to the flexibility of PILOT. When a kid wants to have a computer print out" Hello, Jane" in BASIC, he has to do it like this:

```
10 PRINT "Hello, Jane"
```

In PILOT it's:

```
T: Hello, Jane
```

Kids don't like having to worry about those line numbers or little quotation marks (nobody writes like that). For one thing, it's hard to remember to put the quotes on both ends

He's getting a blast out of the big screen.

of your message, and for another, they're tricky to type since they require the coordination of the shift key.

Kids love computer games and we have used games very successfully to introduce kids (of all ages) to the wonders of micro computers. Now those of you who have spent any time with computer games know that it's a very rare game that deserves a second or third go around with most bright kids. So we've found that giving kids the tools with which they can create their own games has them captivated totally. At that point they are creatively computing. And the novelty of the experience is then only limited to their creative imagination.

When kids who have been through our classes come back to the Computer Center, they often spend more of their time programming games than playing games. After they know how to program, they feel that programming is more fun; with PILOT, they really do get the sense that they "know how."

For people with business applications in mind, we've got a TEXT EDITOR, as well as specific programs that can do MAILING LISTS, CHECK BALANCING, and LOAN AMMORTIZATION.

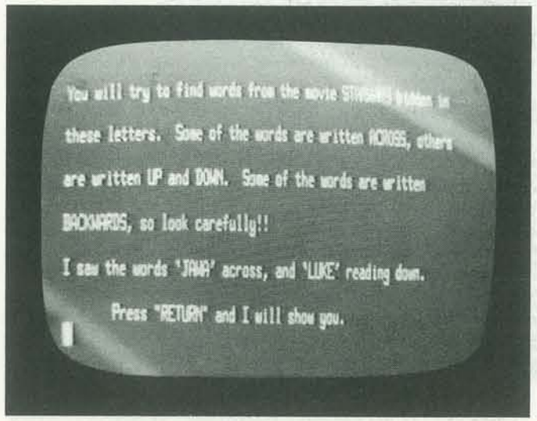
If they just want to "have fun," they can choose from the GAMES listing: they can try to capture the Romulans with their blasted cloaking device, manage the economy of SETATS DETINU (a large nation of Middle Earth), hunt for buried treasure on an island surrounded by sharks who are always looking for a handout (or any other part of you!).

You can have all of these adventures and many more. And while you are using the computer as an electronic vehicle to new realms of reality, you will undoubtedly find that your attitude about computers has been altered drastically. They no longer represent something very large and overwhelmingly powerful and "anti-personal." Rather, a close encounter of the Third Kind with a micro computer will leave you with the afterglow of having discovered a new way to exercise your mind, stimulate your creativity and serve your interest. And there's nothing more "personal" than that!

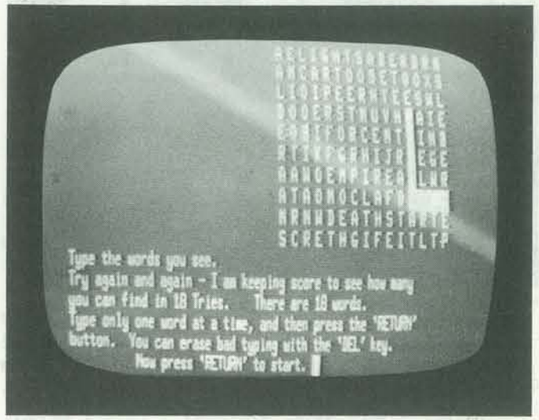
Here is a listing of a program called STARWARS HODGE written in Processor Technology's new PILOT. The original version of the game was written by Keith Anderson. This version was written at Marin Computer Center by David Fox (a 27 year old kid).

```
    R: STARWARS HODGE BY DAVID FOX
    R:  BASED ON HODGE BY S.WILLIAMS & K.ANDERSON
    R: MOD 5/75, 10/77, 5/78
*START
CH:
CA:6,10
T: * * *  S T A R W A R S   H O D G E  * * *
FOOT:
CH:
C:A=0
C:B=0
C:C=0
C:D=0
C:E=0
C:F=0
C:H=0
C:H=0
C:I=0
C:J=0
C:L=1
C:X=0
U(P):*DISPLAY
J(P):*PASS
C:P=1
   CA:2,48
T:
   CA:5
T:This program will show you rows of letters.
T:
T:Press the `RETURN' button to see them.
  INMAX:1
  A:
U:*DISPLAY
FOOT:
   CH:
   CA:3
T:You will try to find words from the movie STARWARS hidden in
T:
T:these letters.  Some of the words are written ACROSS, others
T:
T:are written UP and DOWN.  Some of the words are written
T:
T:BACKWARDS, so look carefully!!
T:
T:I saw the words `JAWA' across, and `LUKE' reading down.
T:
T:       Press "RETURN" and I will show you.
A:
C:Y=8
U:*DISPLAY
U:*BLINK
*PASS
   CA:11
T:Type the words you see.
```

```
T:Try again and again - I am keeping score to see how many
T:you can find in 18 Tries.    There are 18 words.
T:Type only one word at a time, and then press the 'RETURN'
T:button.  You can erase bad typing with the 'DEL' key.
T:          Now press 'RETURN' to start.
   CA:16,40
A:
U:*DISPLAY
   CA:11
   CE:

*ANS
   CA:12
   CL:
INMAX:64
A:
C:A=A+1
U:*TEST
C:K=A-17
J(K):*DONE
J:*ANS

*DONE
   CA:11
   CE:
T:You have had 18 tries, press "RETURN" and
T:I will show you your score.
A:
   CA:11
   CE:
T:Your score is #X out of a possible 18.
TH:Press "\" to quit, or "RETURN" to try again.
INMAX:1
A:
   CH:
J:*START

*DISPLAY
   CH:
   CA:1,34
T:A E L I G H T S A B E R D R N
   CA:2,34
T:A N C A R T O O D E T O O X S
   CA:3,34
T:L I O I P E E R H T E E S W L
   CA:4,34
T:D O O E R S T N U V H L A I E
   CA:5,34
T:E O B I F O R C E N T U I N B
   CA:6,34
T:R T I K F G R H I J R K E G E
   CA:7,34
T:A A W O E M P I R E A E L W R
   CA:8,34
T:A T A O N O C L A F D J A W A
   CA:9,34
T:N R N W D E A T H S T A R T E
   CA:10,34
T:S C R E T H G I F E I T L T P
  R: WOOKIE - DEATHSTAR - TATOOINE - DARTH - XWING - FORCE
   R: ARTOODETOO - SEETHREEPIO - LEIA - LUKE - JAWA - EMPIRE
   R: REBELS - OBIWAN - ALDERAAN - FALCON - LIGHTSABER
   R: TIEFIGHTER
E:
```



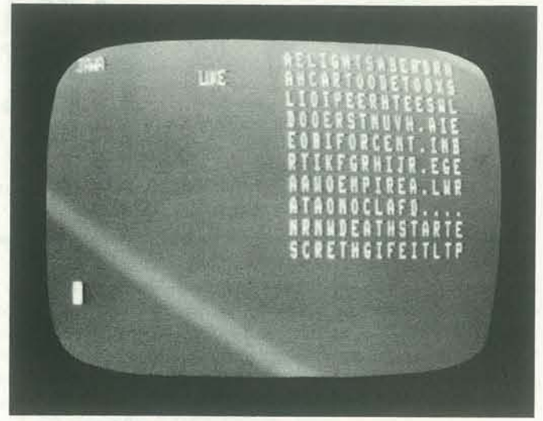

```
*BLINK
   CA:8,56
T:. . . .
   CA:4,56
T:.
   CA:5,56
T:.
   CA:6,56
T:.
   CA:7,56
T:.
U:*NULL
   CA:8,56
T:J A W A
   CA:4,56
T:L
   CA:5,56
T:U
   CA:6,56
T:K
   CA:7,56
T:E
U:*NULL
C:Y=Y-1
J(Y):*BLINK
E:
*TEST
*M1 M:LIGHTSABER
JN:*M2
   CA:1,38
T:. . . . . .
J(B):*M1A
C:B=L
```



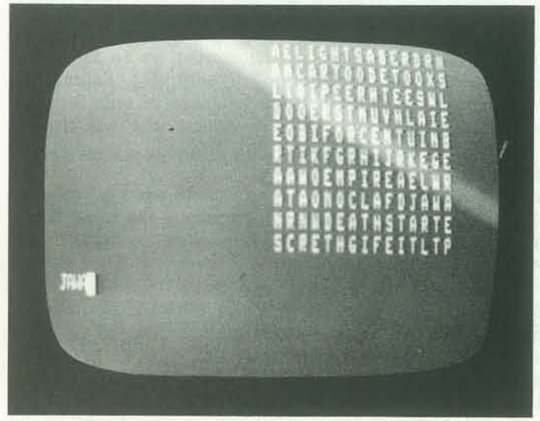

```
C:L=L+1
*M1A CA:B
T:LIGHTSABER
J:*EOT

*M2 M:ARTOODETOO
JN:*M3
   CA:2,40
T:. . . . . .
J(C):*M2A
C:C=L
C:L=L+1
*M2A CA:C
T:ARTOODETOO
J:*EOT

*M3 M:SEETHREEPIO
JN:*M4
   CA:3,38
TY:. . . . . .
J(D):*M3A
C:D=L
C:L=L+1
*M3A CA:D
T:SEETHREEPIO
J:*EOT

*M4 M: FORCE
JN:*M5
   CA:5,42
T:. .
J(E):*M4A
C:E=L
C:L=L+1

*M4A CA:E
T:FORCE
J:*EOT

*M5 M:EMPIRE
JN:*M6
   CA:7,42
T:. . . . . .
J(F):*M5A
C:F=L
C:L=L+1
*M5A CA:F
T:EMPIRE
J:*EOT

*M6 M:FALCON
JN:*M7
   CA:8,42
T:. . . . . .
J(G):*M6A
C:G=L
C:L=L+1
*M6A CA:G
T:FALCON
J:*EOT

*M7 M:DEATHSTAR
JN:*M8
   CA:9,42
T:. . . . . .
J(H):*M7A
C:H=L
C:L=L+1

*M7A CA:H
T:DEATHSTAR
J:*EOT

*M8 M:TIEFIGHTER
JN:*M9
   CA:10,39
T:. . . . . .
J(I):*M8A
C:I=L
C:L=L+1
*M8A CA:I
T:TIEFIGHTER
J:*EOT

*M9 M:JAWA
JN:*M10
   CA:8,56
T:. . . .
J(J):*M9A
C:J=L
C:L=L+1
*M9A CA:J
T:JAWA
J:*EOT

*M10

*MD1 M:ALDERAAN
JN:*MD2
   CA:2,34
T:.
   CA:3,34
```



```
T:.
   CA:4,34
T:.
   CA:5,34
T:.
   CA:6,34
T:.
   CA:7,34
T:.
   CA:8,34
T:.
   CA:9,34
T:.
J(B):*MD1A
C:B=L
C:L=L+1
*MD1A CA:B,20
T:ALDERAAN
J:*EOT

*MD2 M:TATOOINE
JN:*MD3
   CA:1,36
T:.
   CA:2,36
T:.
   CA:3,36
T:.
   CA:4,36
T:.
   CA:5,36
T:.
   CA:6,36
T:.
   CA:7,36
T:.
   CA:8,36
T:.
J(C):*MD2A
C:C=L
C:L=L+1
*MD2A CA:C,20
T:TATOOINE
J:*EOT

*MD3 M: OBIWAN
JN:*MD4
   CA:4,38
T:.
   CA:5,38
T:.
   CA:6,38
T:.
   CA:7,38
T:.
   CA:8,38
T:.
   CA:9,38
T:.
J(D):*MD3A
C:D=L
C:L=L+1
*MD3A CA:D,20
T:OBIWAN
J:*EOT
*MD4 M:WOOKIE
JN:*MD5
   CA:4,40
T:.
   CA:5,40
T:.
   CA:6,40
T:.
   CA:7,40
T:.
   CA:8,40
T:.
   CA:9,40
T:.
J(E):*MD4A
C:E=L
C:L=L+1
*MD4A CA:E,20
T:WOOKIE
J:*EOT

*MD5 M:DARTH
JN:*MD6
   CA:4,54
T:.
   CA:5,54
T:.
   CA:6,54
T:.
   CA:7,54
T:.
   CA:8,54
T:.
J(F):*MD5A
C:F=L
C:L=L+1
*MD5A CA:F,20
T:DARTH
J:*EOT

*MD6 M:LUKE
JN:*MD7
   CA:4,56
T:.
   CA:5,56
T:.
   CA:6,56
T:.
   CA:7,56
T:.
J(G):*MD6A
C:G=L
C:L=L+1
*MD6A CA:G,20
T:LUKE
J:*EOT

*MD7 M:LEIA
JN:*MD8
   CA:4,58
T:.
   CA:5,58
T:.
   CA:6,58
T:.
   CA:7,58
T:.
```




```
J(H):*MD7A
C:H=L
C:L=L+1
*MD7A CA:H,20
T:LEIA
J:*EOT

*MD8 M:XWING
JN:*MD9
   CA:2,60
T:.
   CA:3,60
T:.
   CA:4,60
T:.
   CA:5,60
T:.
   CA:6,60
T:.
J(I):*MD8A
C:I=L
C:L=L+1
*MD8A CA:I,20
T:XWING
J:*EOT

*MD9 M:REBELS
JN:*MD10
   CA:2,62
T:.
   CA:3,62
T:.
   CA:4,62
T:.
   CA:5,62
T:.
   CA:6,62
T:.
   CA:7,62
T:.
J(J):*MD9A
C:J=L
C:L=L+1
*MD9A CA:J,20
T:REBELS
J:*EOT
*MD10     R: NO MATCH FOUND */
E:

*EOT      R: A MATCH WAS FOUND */
C:X=X+1
E:

*NULL E:
20
E:

Press "RETURN" to go on

Immediate commands are:
LOAD  GET   SAVE    COPY   READ
RUN   EDIT  INFO    CLEAR  SCRATCH
```



All in the MIND

BY WARWICK SLOUGH

"Belial came last, than whom a Spirit more lewd
Fell not from heaven . . . " Milton, *Paradise Lost* (Book I)

Warwick Slough lives in Twickenham, Middlesex, U.K. The only things we know about Warwick must be inferred from his story. He has read Milton, is familiar with computers, parapsychology and perhaps the plans of the lower worlds—but read on . . . —*RZ*

The parapsychological department of the university was a small compact arrangement of rooms on the third floor of a characteristically modern, glass and concrete, oblong box. The money to maintain the random letter distribution (RLD) research work going on under Dr. Robert Sinclair was grudgingly donated by a number of foundations. Some part of this reluctance was probably due to the fact that Sinclair was a convinced sceptic about the reality of psi phenomena, and therefore frequently found himself in the curious position of seeking financial support to carry out work which was intended to rationalise away the necessity for invoking paranormal processes at all.

The most recent investigations involved the random selection of sequences of letters of the alphabet by designed electronic circuitry, which were displayed as green, glowing discharges in a small dark grey unit. The intention had been to find subjects who could influence the order in which letters were produced by exerting their wishes in some separate room. It had however taken rather longer than anticipated to set up the electronic equipment and to get it running in an acceptable manner. On this particular afternoon Sinclair was once again engaged in testing the experimental set-up. It was midwinter and the sky had already turned to a leaden grey layer, shot with crimson and mauve, as the sun disappeared below the horizon. In the laboratory the reduction in light intensity made the glowing green letters stand out from their background and Sinclair made no attempt to introduce any artificial illumination. **E, A, P, B** formed and disappeared. As he recorded the sequence a shiver of cold ran down his back and he wondered if he was about to succumb to the influenza virus which had recently been running through the population. **D, E, A, T** flashed the display unit—seemed as though there was an inbuilt preference for E and A he thought—**H** added the machine, as if through an afterthought. Oh fine—DEATH—if the circuitry was going to prefer arrangements of letters what the hell was wrong with SEX? Outside it was almost dark now and a flurry of snow whipped against the window pane. He started another random selection run, becoming absorbed in the task at hand. Had he glanced up he might have seen, gazing in, an apparently disembodied head, floating three stories up and showing some interest in his endeavours. The head possessed a malevolently evil face which was contorted into a fierce grin. Since however it was quite obviously not within conventional experience his brain might not have registered it as a recordable image, such was the intensity of his determination to fit all phenomena into the framework of everyday experience.

However the head showed no sign of requiring recognition, and continued to watch with amused interest the flickering electronic displays taking place at the bench. Presently Sinclair with a sigh took up a screwdriver and removed a plate from one of the units. He then examined various wire connections inside the assembly. A sudden unpleasant tingling in his fingers told him that the current was not quietly following its preordained pathway. He returned once again to the flashing letters. **B-E-L-I-A-L** spelled the machine. Sounded familiar he thought—where in hell had he heard BELIAL? As he attempted to rezero the machine his finger contact with the buttons seemed elusive. He attempted to rise from his chair and found surprisingly that he could look down at his own figure from a point slightly above its head. Furthermore by focussing his attention on any part of the room he appeared to be in that part of the room. He focussed on the window and noticed the snow which was now beginning to form a thin white layer on roads and paths. For the first time he now also noticed the head. Rationalising to himself he reasoned that no head could exist some thirty feet from the ground without sign of support—must be hallucinatory he thought. A very vivid experience it certainly was for the head now grinned wickedly at him and parts of a body were faintly visible as an electric blue fluorescence. A talon-like hand seemed to be beckoning him. Where in hell was it inviting him? Where in Hell?

Somewhere in the immensity of space-time the Prince of Darkness sighed. Another behavioural psychologist to join an already overstaffed team whose aim was to rationalise away the existence of himself and his influence. That idea of evolving consciousness in man had been very productive initially (consider the bloodshed and cruelty engendered by differing devotees of THE TRUTH), but now it was producing some boring souls.

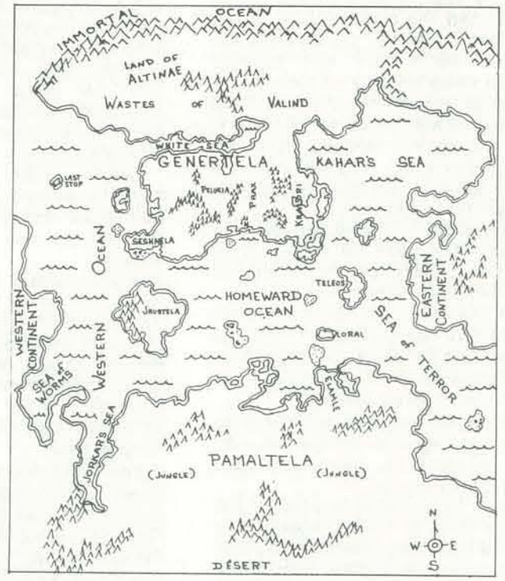
It would probably be necessary to submit a supplementary estimate for time spent in the current era to the ACEE (Advisory Council for Engendering Evil). Better have a word with Streicher (or was it Eichmann who was chairing these days?)

Didn't help of course when all the Cosmologists were joining the other side—working on the technology of the 'Big-Bang.'

His reverie was disturbed as a circular silver vehicle sped by close to the speed of light. Blast Kirk and his Enterprise; his voyages seemed eternal in time as well as space. □

RUNEQUEST

BY STEVE PERRIN AND FRIENDS



Runequest *is a role-playing fantasy adventure game (see Dragonsmoke,* People's Computers, *Volume 7, no. 2, Sep/Oct 1978). In this and future issues of* PC *(soon to be* Recreational Computing*), we will publish excerpts from* Runequest *and begin building computer programs to (1) assist a gamemaster conduct a game of* Runequest *or (2) implement a simple form of* Runequest *as a computer game. If you wish to play along, we suggest you get your very own copy of* Runequest. *$8 from The Chaosium, P.O. Box 6302, Dept. P, Albany CA 94706.*

Here are some excerpts from Runequest.

HISTORY

Glorontha was created by its deities from the Primal Void of Chaos. At first, there was no history, for the initial creation formed the period of magic and timeless simultaneity called Godtime. During this time, all the world was populated with the beings and races of the Golden Age.

The birth into this world of Umath, the Primal Air, shattered the period of peace with crowding, confusion and fear. This escalated into the War of the Gods, resulting in the death of the Sun, the Great Darkness and Chaos let loose upon the world. Total Destruction seemed near.

But Glorontha was not destroyed. Spurred on by Hope and Chance, the Seven Lightbringers sought through the chaos-ridden cosmos and, after harrowing adventures, liberated the life-giving Sun God. The joyful god leapt to his rightful place in the sky and the dark was banished.

To keep entropic Chaos from coming to their realm again, the gods swore great oaths and compacts, resulting in the Great Compromise. Men call this Time. By altering the world, the gods saved it, and History began.

There followed the Dawn Ages. This was a time of peace and growing, as the survivors rebuilt and taught themselves the way of the world inside Time. The Ancient Races, Elves, Dragonewts, Dwarves, and Trolls, were as healthy and proud as Humans. In

Dragon Pass, on the continent of Genertela, the First Council was a balanced organization including humans and non-humans, men and women, mortals and divinities.

This age of innocence could not last. Ancient feuds were remembered and conflicts grew with peoples outside the Council. Inside dissension and outside enemies forced the formation of the Second Council, a warlike empire, which grew in arrogance and power until they dreamed of bringing back the Godtime. the experiment ended in the birth of Gbaji, the Chaos god, whose reign of terror kept Glorontha at war with itself for 75 years. This was the death knell of the inhuman races, which have never had the same stature in the world since then. This was the end of the Dawn Ages.

Out of this shattered world grew new political entities. Among the most powerful was the state which called itself the Empire of the Wyrms Friends. This empire, centered in Dragon Pass, lasted for 500 years, proud of and famous for their friendship with the Dragons and Wyrms of the Pass. The peoples and places thrived, made contact with two other continents, and delved deep into spiritual byways.

After several hundred years the Empire was replaced by a ruling body of men and gods called the Third Council. Legends relate that there was no telling the men from the gods in the Council chambers.

But the magic of the Council could not counter the miseries of its worshippers, or control the swords of the rebels who did not sacrifice to them. Foreign gods gained power and prestige as the provinces of the Third Council revolted or were overrun by invaders.

At last the Council turned its energies to defending its worshippers. Epic battles raged across the land. At last the Dragonewts, dormant for centuries, rose against the Council and slew them all.

Seventy-eight years later the humans who had aided in the destruction of the Empire turned against the Dragonewts. With few allies, they seemed helpless before the destroying might of the hordes. Nests were scoured with wildfire, planted with seeds laid by Chaotic parasites, sealed by diseases unnamable, and cursed by fear and loathing for any that would mourn them.

Heartless and relentless, the humans crushed and plundered, but when they closed in upon Dragon Pass the assembled human armies met those who would mourn the loss of Dragonewts. The Dragons from all across Time and Space assembled in their ancestral home to preserve the purity of their birthplace. The Dragonkill War got its name from what the Dragons did, not what they suffered. Humans have feared the Dragons since that time, despite draconic indifference, and none now seek to smash the eggs of the Dragonewts, even though they might plunder the nest city.

Dragon Pass was abandoned by humanity. Elsewhere, old empires shook and the seas were utterly closed to human crossing. New Kingdoms rose to fill the power vacuums and new magics and deities broke free from ancient shackles.

North of Dragon Pass, in the region called Peloria, there arose the Red Mood Goddess. In her were balanced Constancy and Change, life and death, love and indifference, and all the dichotomies of the Universe, including a touch of Chaos. Her arrival changed the face of the land.

After living on Glorontha a short time she ascended to the heavens where she remains in her cyclical beauty, viewing the land which she left to her family below. The ever-reincarnating Red Emperor of the Lunar Empire is her son and her pride.

The Lunar Empire expanded from its founding in 1220. When defeated it rebounded stronger than ever. They went North to the icy wastes, East to the bounds of the Horse barbarian lands, and West until they were halted by the magical Syndics Ban. But in the South the Empire continued to grow and grow.

When they reached Dragon Pass they found it populated again. Moreover, it was owned by the Kingdom of Sartar, ruled by a wise dynasty cautious to maintain the great and magical laws which gave it its strength.

Despite initial defeats, the Emperor eventually marched almost unhindered into the Sartar capital of Boldhome. He defiled the Temple of Sartar and ravaged the city. So far, it was the typical Lunar conquest.

But the spirits of Sartar Temple incited the natives to rebellion. When the Lunar Priestesses attempted to construct a Temple of the Reaching Moon on Wind Top to extend the Glowline, a Dragon unearthed itself under the foundation, devouring the priestess there and half of the attendant armies. Shortly afterward, rebellion in the outpost city of Pavis spelled a Lunar defeat, and the Victorious barbarian warlord led his army towards Dragon Pass.

The warlord was Argath Dragontooth, member of a minor Sartar household and refugee from Lunar Justice. He had grown famous amongst the tribes of Prax and now claimed heirdom to Sartar's realm. He defeated Lunar forces in a military victory and relit the fire in Sartar's Temple with a command. Thus, the Empire was thrown back again.

The bravery and glory of Sartar's fight for independence attracted thousands of volunteers, and people from all about Glorontha became Adventurers in order to build up their skills to take their places in the ranks. The period was known as the Hero Wars, and the fighting around Dragon Pass drew the greatest collection of Heroes and Super-Heroes the world had ever seen in one place.

This is the game of that period. In these pages one learns how to start to become a Hero, to take one's place in the Hero wars.

TECHNOLOGICAL BASE

Glorontha is a Bronze Age world. Bronze is common, and can be mined directly from the bones of the gods who died in the Gods' War. These bones provide a ready source of the metal.

Unalloyed, or pure, metals, such as iron, lead, tin, and copper, prevent a person from using magic unless he is "sealed" to the Rune connected with that metal. See Chapter X, for a list of what metals are aligned to which Runes. Note that all coins are alloyed, as are gold and silver ornaments.

SOCIOLOGICAL BASE

Glorontha is an Ancient Period and early Dark Ages world. It has far more to do with Mesopotamia, Ancient China, Hyboria, and Lankhmar than it does with Medieval Europe, Le Mort' D'Arthur, or the Carolingian Cycle. Its heroes are Conans, Grey Mousers, and Rustums, not Lancelots, Percivals, and Rolands.

Unlike the worlds in other role playing games, there is no Alignment, as such. People have allegiances to nations, cities, religions, and tribes, not to abstract concepts. It is also possible for people within the game to survive quite well with no allegiances whatever except to themselves.

In Glorontha, the gods, in the forms of their followers and cults, play an active and important part in most major events. However, most gods are complementary, and rarely oppose each other directly. Only the gods of Power are actively antagonistic, and even then only within their own spheres of interest.

MONETARY BASE

The coinage of Glorontha is based on silver. While both gold and copper are used as coins, silver is by far the most common monetary metal. Silver coinage was first introduced into Glorontha by the Lunar Empire. In the Empire, the basic silver coin is referred to as an Imperial. In Sartar it is called a Sovereign, and in the city of Corflu, run by various guilds, it is called a Guilder. All of these coins are roughly equivalent in value. The generic term for silver coins used over the continent is the Lunar, in honor of the Lunar Empire. Note that a Lunar, referred to in the game as L, is worth about one pre- WW II English Pound, or five Dollars.

Gold was the first coinage of the world, brought to the people by the enigmatic Sun-Wheel Dancers. In their honor, Gold coins are still called Wheels. Gold, however, is scarce and very valuable. One golden Wheel equals twenty Lunars of silver. Gold is still mainly used as a means of settling debts between nations rather than individuals.

Copper coins were invented by the Dwarves. As is usual with any innovation brought out by that most inventive race, humans shrink from acknowledging the contribution. The copper coin is called a Clack, or often just a Copper. It takes ten Clacks to equal one Lunar.

PURPOSE OF THE GAME

The title of the game, RUNEQUEST, describes its goal. The player creates one or more characters, known as Adventurers, and plays them in various scenarios designed by a Referee. The Adventurer has the use of combat, magic, and other skills to survive and gain glory, advancement in his skills, and treasure. The Referee has the use of assorted monsters, traps, and his own wicked imagination to keep the Adventurer from his goal within the rules of the game. A surviving Adventurer gains experience in fighting, magic, and other skills, as well as money to purchase further training.

The Adventurer progresses in this way until he is so proficient that he comes to the attention of the High Priests, Sages, and Gods. At this point he has the option to join a Rune Cult. Joining such a cult gives him many advantages, not the least of which is aid from the god of the cult.

Acquiring a Rune by joining such a cult is the goal of the game, for only in gathering a Rune may a character take the next step, up into the ranks of Hero, and perhaps Superhero.

EQUIPMENT NEEDED TO PLAY

Besides this set of rules, RUNEQUEST players will need the following materials:

DICE

Dice of many shapes and sides are needed for this game. They are usually available at any game store. If there are no game stores in your area, addresses of suppliers are printed in the appendix. The minimum mix needed for play is as follows:

TYPE	ABBREVIATION
1 pair of 20 sided percentile dice	D100
1 20 sided die marked from 1 to 20	D20
1 12 sided die	D12
1 8 sided die	D8
3 6 sided dice	D6
1 4 sided die	D4

Within these rules, dice will be referred to using the above abbreviations. Thus, if it is necessary to roll two twelve sided dice, it will be written as 2D12. Occasionally, it will be necessary to roll one three sided die, D3. Simply roll a D6 and divide by two, always rounding up. Thus, 1-2 = 1, 3-4 = 2, 5-6 = 3.

OTHER PLAYING AIDS

RUNEQUEST does not need a playing board. The players' imaginations provide the stage on which the characters act. As assistance in this drama, you'll find the following helpful:

PAPER, blank, lined, and quadrille ruled (graph paper)

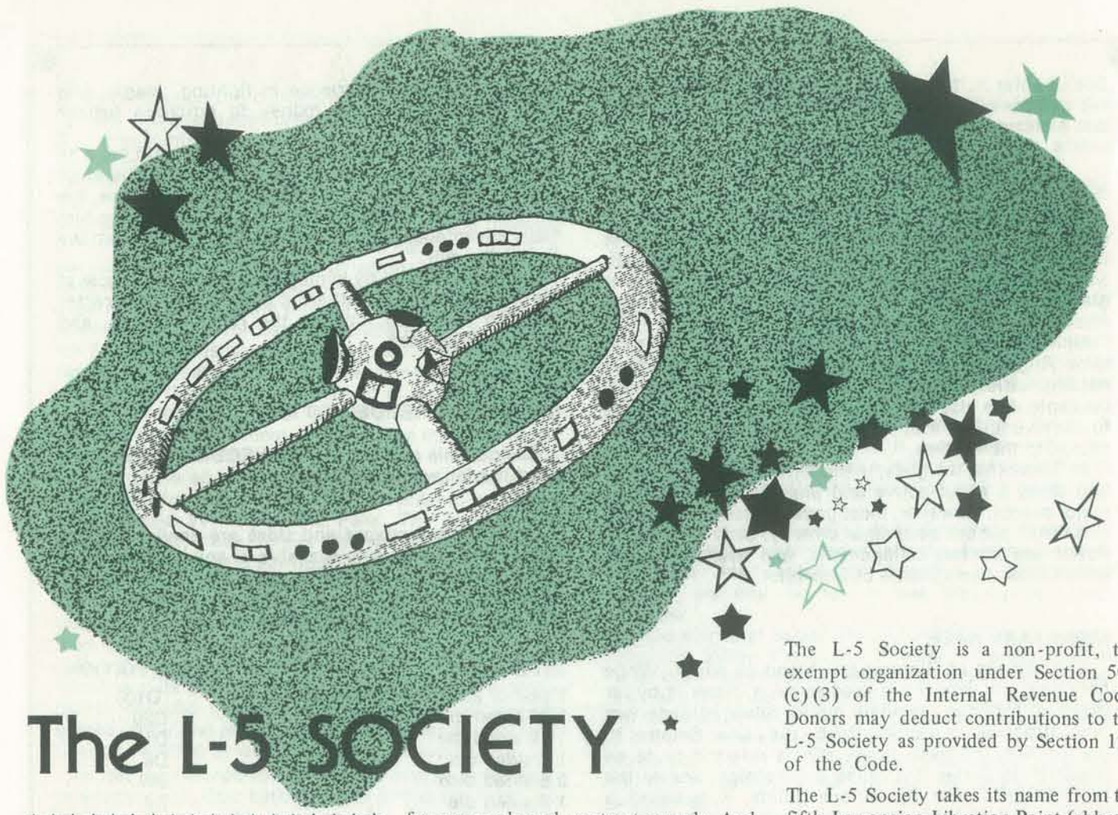
PENCILS AND PENS

TIN/LEAD OR PLASTIC FIGURINES (these are optional, but give the play some focus and help settle arguments over who was where. We recommend 25mm military minatures as the best all around size.)

TIME (ranging from a couple of hours for a quick scenario to years for a long-running campaign)

GLASS OR PLEXIGLASS SHEET AND GREASE PENCIL (for sketching in the outlines of rooms and corridors)

IMAGINATION □



The L-5 SOCIETY

For a list of L-5 Society Chapters, see page 41.

BY CAROLYN HENSON

Here is information from and about the L-5 Society. If you are interested in the colonization of near space, try the L-5 Society. There are more interesting things to do in the galaxy than shooting up Klingons. Hmmm . . . maybe the Don Quixote Starship will *happen. – The Dragon*

If you want to keep well informed about developments affecting space settlements, industries and solar power satellites, you'll want to stay up to date with the *L-5 News*. The *L-5 News*, the monthly magazine of the L-5 Society, carries fast-breaking news, in-depth articles by major workers in these fields, reviews of pertinent books, articles and papers, announcements of job opportunities, requests for proposals and contract awards. And, perhaps most important, the *L-5 News* provides a forum for discussing what history may judge to be the most significant issues of our century.

The L-5 Society sponsors several technical conferences each year. Next on the schedule is "The Future United States Space Program," cosponsored by the American Astronautical Society, to be held in Houston October 30 through November 2. Society members who wish to attend will receive a discount on conference fees.

If you want to influence legislation affecting space settlements, industries and solar power satellites, you may wish to sign up for the Society's Legislative Information Service (LIS). This service, available at no extra cost to Society members who request it, provides frequent first-class mailings covering space related bills, appropriations, Presidential and executive branch studies and policy statements. LIS gives subscribers the opportunity to participate in Department of Energy hearings and space-related public events.

The L-5 Society is a non-profit, tax exempt organization under Section 501 (c) (3) of the Internal Revenue Code. Donors may deduct contributions to the L-5 Society as provided by Section 170 of the Code.

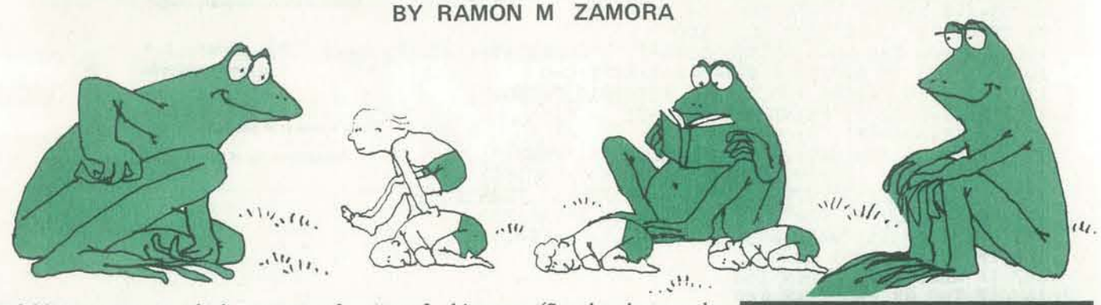
The L-5 Society takes its name from the fifth Lagrangian Libration Point (abbreviated L-5). In 1772, the French mathematician Joseph Lagrange determined that five libration points exist in the Earth/Moon system. Three of them lie on a line connecting the Earth and Moon; these are L-1, L-2, and L-3. They are unstable; a body placed there and moved slightly will tend to move away, though it will not crash directly onto the Earth or Moon. The other two, L-4 and L-5, lie at an equal distance from the Earth and Moon, in the Moon's orbit, thus forming equilateral triangles with Earth and Moon. Due to the sun's disturbing influence, a colony could be placed not directly at L-4 or L-5, but rather in an orbit around one of these points.

However, if we can't have an L-5 colony, or even one at L-4, we'll settle for less interesting orbits. Creating tens of thousands of jobs and homes for us in space before the end of the century is the goal of the L-5 Society.

For more information contact Carolyn Henson, L-5 Society, 1620 N. Park Ave., Tucson AZ 85719 (602) 622-6351. □

TRS-80: FROGS!

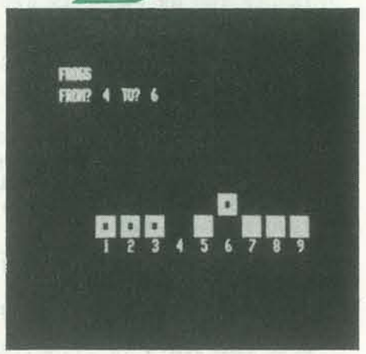
BY RAMON M ZAMORA

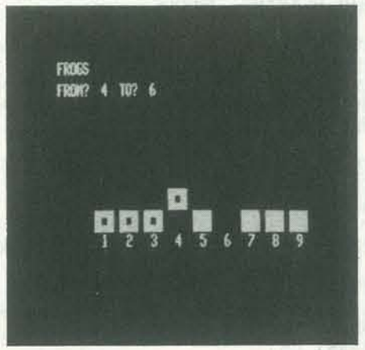


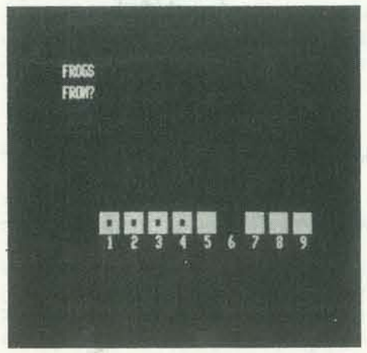
FROGS is a game in which two sets of four objects are placed on a nine position board. One set placed to the left; the other to the right. This layout leaves a vacant position in the center of the board. The game consists of reversing the two sets of objects by either sliding one object to an empty position or having one object jump one other object into a vacant place. No other moves are considered legal. This game was presented by Mac Ogelsby for the SR-52 in the October 1977 issue of *Calculators/Computers* (Vol. 1, No. 2, pp. 5–8).*

About a month ago, the Dragon (Bob Albrecht), gave me a Level II 16K TRS-80 and asked me to look at some games such as FROGS. I started with FROGS since it allowed me to take the TRS-80 and see if I could really make the frogs 'jump'. The accompanying program, which utilizes the TRS-80 PRINT@ feature, provides a simple animated version of this game. (See the photographs of the screen during a 'jump.')

I also experimented with two other features on the TRS-80: the INKEY$ keyboard strobe function and the character construction capability. The INKEY$ feature allows the user to enter all program input data without having to use the ENTER key. The string handling features of the language allows the concatenation of special graphics codes to create a set of characters to represent the frogs.





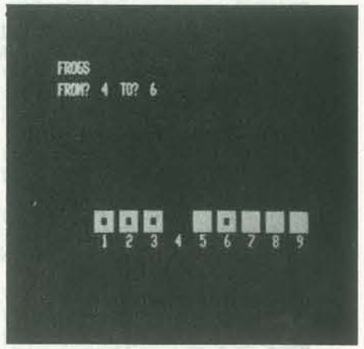


The combination of these capabilities of the TRS-80 demonstrates the potential of this machine for developing interesting variations of games and educational programs. (Look for a TRS-80 animated version of REVERSE elsewhere in this issue.)

For the readers who wish to run FROGS and would like to minimize the amount of effort needed to type in the program:

Do not type statements 95 to 97
and statements 7000-8010.

These program lines contain the instructions for running FROGS and will not affect the general program execution. Have fun with FROGS!!



* MAC also had a teletype version of this game in *PCC*, Vol. 5, No. 3, Nov-Dec,1976

```
85 REM  ** FRØGS  16K LEVEL II BASIC TRS-80  R. ZAMØRA  9/9/78  **
90 DIM P(9): A$="△△△△"
91 B$= CHR$(191) + CHR$(179) + CHR$(191)                    Construct graphic characters.
92 C$= CHR$(191): C$= C$+C$+C$
95 PRINT"FRØGS GAME--DØ YØU NEED INSTRUCTIØNS?";            Check if instructions are needed.
96 D$=INKEY$: IF D$="" THEN 96 ELSE 97
97 IF D$="Y" THEN 7000 ELSE 100
100 CLS:FØR I=0 TØ 8:PRINT@528+4*I,"△"+CHR$(49+I)+"△△";:NEXT   Print characters 1–9
105 FØR I=1 TØ 4:P(I)=1:P(I+5)=-1:NEXT:C=0                      in proper positions.
110 PRINT@10,STRING$(30,"△");: PRINT@10,"FRØGS";
120 FØR I=0 TØ 3: PRINT@464+4*I,B$;                         Print symbols to be used as 'frogs.'
130 PRINT@484+4*I,C$;:NEXT
140 PRINT@74,STRING$(20,"△");:PRINT@74,"FRØM?";             Accept user input and validate.
142 F$=INKEY$: IF F$="" THEN 142 ELSE F=VAL(F$)
143 PRINT@80,F;:PRINT@84,"TØ?";: IF F$="S" THEN 1032
144 F$=INKEY$: IF F$="" THEN 144 ELSE T=VAL(F$)
145 PRINT@88,T;:PRINT@20,STRING$(23,"△");:C=C+1
146 IF F$="S" THEN 1032
147 IF ABS(T-F)>2 THEN 160
148 IF T=0 ØR F=0 THEN 160
150 IF T-F=0 ØR P(T)=1 ØR P(T)=-1 THEN 160 ELSE 170
160 C=C-1:PRINT@20,"ILLEGAL MØVE--TRY AGAIN";:GØTØ140
170 IF ABS(T-F)>1 THEN 180
172 S=F:X=464:GØSUB 2000: S=T:GØSUB 3500                    Move is a slide.
174 GØTØ 1000
180 S=F:X=464:GØSUB 2000: X=400:GØSUB 3500
190 GØSUB 2000: S=T:GØSUB 3500                              Move is a jump.
200 GØSUB 2000: X=464:GØSUB 3500
1000 P(T)=P(F): P(F)=0: S=0: T=0
1005 FØR I=1 TØ 4:S=S+P(I):T=T+P(I+5):NEXT
1010 IF S=-4 AND T=4 THEN PRINT@74,"YOU DID IT IN ",C," MØVES";ELSE 140
1020 PRINT@20,"WØULD YØU LIKE TØ PLAY ANØTHER GAME?";
1022 D$=INKEY$: IF D$="" THEN 1022 ELSE 1030
1030 IF "Y"=D$ THEN 100                                     Check for end of game and replay.
1032 PRINT@128,"THANK YØU FØR PLAYING--BYE";
1040 END
2000 PRINT@X+4*(S-1),A$;:RETURN                             Blanks screen where move was made.
3000 FØR J=1 TØ 300:NEXT:RETURN                             Delay so jump is visible.
3500 IF P(F)=1 THEN 4000 ELSE 6000
4000 PRINT@X+4*(S-1),B$;:GØSUB 3000:RETURN                  Prints symbol with hole in center.
6000 PRINT@X+4*(S-1),C$;:GØSUB 3000:RETURN                  Prints solid symbol.
7000 GØSUB 8000                                             Instructions.
7020 PRINT"FRØGS BEGINS WITH 8 SYMBØLS ØN A SPACE FØR 9 SQUARES."
7030 PRINT"THE DISPLAY LØØKS LIKE THIS TØ START:"
7040 FØR I=0 TØ 3:PRINT@464+4*I,B$;
7050 PRINT@484+4*I,C$;:NEXT
7060 FØR I= 0 TØ 8:PRINT@528+4*I,"△"+CHR$(49+I)+"△△";:NEXT
7070 PRINT@576,"THE ØBJECT IS TØ REVERSE THE TWØ SETS ØF SYMBØLS"
7080 PRINT"IN THE FEWEST MØVES."
7090 PRINT"THE END ØF THE GAME ØCCURS WHEN THE SYMBØLS ARE"
7100 PRINT"IN THESE PØSITIØNS:"
7110 FØR I=0 TØ 3:PRINT@848+4*I,C$;
7120 PRINT@868+4*I,B$;:NEXT
7130 PRINT@896,"HIT ANY KEY TØ CØNTINUE--";
7140 D$=INKEY$: IF D$="" THEN 7140
7150 GØSUB 8000
7160 PRINT"A MØVE IS A SLIDE ØF ØNE SYMBØL TØ A VACANT SQUARE ØR"
7170 PRINT"A HØP ØF ØNE SYMBØL ØVER ANØTHER SYMBØL TØ A"
7180 PRINT"VACANT SQUARE.  NØ ØTHER MØVES ARE ALLØWED."
7190 PRINT"THE NUMBER ØF THE SYMBØL TØ BE MØVED AND THE"
7200 PRINT"NUMBER ØF THE SPACE TØ MØVE TØ ARE TØ BE INPUT"
7210 PRINT"TØ THE QUESTIØNS:"
7215 PRINT"      FRØM?     TØ?"
7220 PRINT"HITTING THE S-KEY AT THESE QUESTIØNS WILL STØP THE GAME."
7230 PRINT"ARE YØU READY TØ PLAY??  HIT ANY KEY--"
7240 D$=INKEY$: IF D$="" THEN 7240 ELSE 100
8000 CLS:PRINT"INSTRUCTIØNS FØR FRØGS"                      The symbol △ indicates a space.
8010 PRINT"----------------------":PRINT:RETURN
```

□

PHANTNUM

BY HOWARD A PEELLE

Good news, APL fans! We will have APL games, simulations or recreations in most every issue from now on. Use them as is or rewrite them in your favorite other language. And — just to keep up the momentum, send us an APL game to publish.
— The Dragon

INTRODUCTION

"PHANTNUM" is an elementary mathematics game in which the player is challenged to discover "*phan*tom *num*bers" — numerals which have been disguised as different symbols.

The game of PHANTNUM presents a unique and intellectually active way for children to study certain aspects of elementary mathematics. Specifically, they can learn properties of the fundamental arithmetic *functions* + − x while at the same time reinforcing their computational skills. For instance, the child playing PHANTNUM confronts the identity elements for functions, and experiences the commutative property in action. Also, children can be introduced to *modular number systems* at an early age without formal treatments. The general idea of allowing numbers to be "masked" or unknown — a prerequisite to understanding the concept of *variable* — can be accepted easily and naturally. And, children playing this game develop their general *problem-solving* abilities as they engage in systematic experimentation, observation, and analysis to decide which experiment to do next in order to gain the most information.

HOW THE GAME IS PLAYED

At the onset of the game, the computer (or some other independent agent) assigns different symbols to the ten numerals. For instance, the letters A B C D E F G H I J might be assigned as "masks" for the numerals 0 1 2 3 4 5 6 7 8 9 (although not necessarily in that order). Then, the player is invited to discover which are which by using three arithmetic functions: PLUS, MINUS, and TIMES. That is, in order to determine which numbers are hidden by which masks, the player may make inquiries to the computer. The player's inquiries must be legal arithmetic expressions. The computer's responses to these inquiries are the correct evaluations, but are expressed in terms of masks. For example:

```
        B PLUS C                (inquiry)
F               (result)
        H MINUS B               (inquiry)
A               (result)
        C TIMES E               (inquiry)
J               (result)
```

Note that only the last (rightmost) digit of a multi-digit phantom number is displayed and that the negative symbol preceding a negative phantom number is not shown.

The above evaluations would hold true if the following assignments had been made:

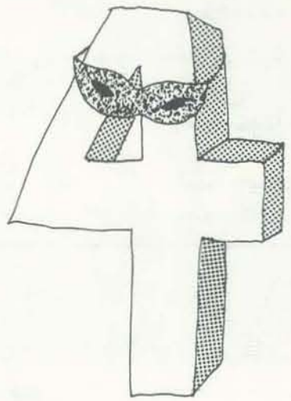
A ← 4
B ← 7
C ← 2
D ← 5
E ← 8
F ← 9
G ← 1
H ← 3
I ← 0
J ← 6

When the player thinks s/he has discovered a phantom number, the computer may be used to check conjectures. For example:

```
        A = 4                   (Does A equal 4?)
1               (Yes)
        B = 7                   (Does B equal 7?)
1               (Yes)
        C = 5                   (Does C equal 5?)
0               (No)
```

PLAYING PHANTNUM

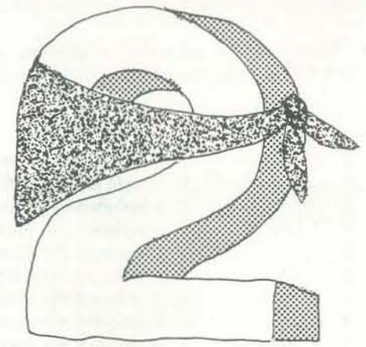
A sample game of PHANTNUM played via computer is shown below. The child enters the name of the program PHANTNUM and the computer prints the rest.



```
                PHANTNUM

BEWARE. . . . THE NUMBER PHANTOM IS HERE!
I HAVE PUT MASKS ON THE NUMERALS
      0 1 2 3 4 5 6 7 8 9
SO YOU CAN'T RECOGNIZE THEM.
THE MASKS ARE:
      A B C D E F G H I J
AND YOU CAN USE THEM LIKE NUMBERS
WITH PLUS MINUS AND TIMES LIKE THIS:
      B PLUS C
      H MINUS B
      C TIMES E
BUT I WILL ONLY SHOW YOU THE LAST LETTER
AND WILL LEAVE OFF THE NEGATIVE SYMBOL
IN THE ANSWER.

GO AHEAD. TRY SOME YOURSELF.
. . . . . HEH . . . HEH. . . . . .
YOU'LL NEVER FIGURE OUT WHICH ARE WHICH!

      B PLUS C
F
                    The child tries the examples
                    suggested, and the computer
                    prints phantom numbers as
                    results.
      H MINUS B
A

      C TIMES E     What to try now? Well, there
J                   is one numeral which can be
                    exposed right away.

      H MINUS H     Any number MINUS itself
I                   must result in 0 (the identity
                    element for subtraction).

      I = 0         Is I equal to 0?
1                   Yes.

      B PLUS I      If I is the mask for 0, then any
B                   number added to I will result in
                    the same number. Check. (0 is
                    the identity element for addi-
                    tion.)

                    Trying multiplication:
      A TIMES A
J                   A is not 1 (the identity element
                    for multiplication). Nor is it
                    5 or 6. (How do we know?)

      A TIMES B
E                   B is not 1. (Neither A nor B
                    is 1 or 0).

      A TIMES C
E                   Hmmm. A x B and A x C yield
                    the same result (modulo 10).
                    Could be 2 x 3 (6) and 2 x 8
                    (16 with the 1 left off) or 2 x
                    4 and 2 x 9 or 4 x 2 and 4 x 7
                    or. . . etc.
                    Anyway, neither B nor C can be
                    5 (or 0) and A cannot be 3.
                    (Why not?)

      A TIMES D
I                   Indeed, A must be even! (Why?)

      A TIMES E
C                   And, the results of multiplying
                    by A must be even also.

      A TIMES F
J

      A TIMES G
A                   Aha! A non-zero number times
                    G is itself. G must be 1 (or 6).

                    But G cannot be even if A, J, E,
                    I and C are even. (There are only
                    five even numerals.)

                    Therefore, G must be 1.
      G = 1
1                   It is.
                    Finding 1 is a key!

      G PLUS G
C
```



```
      C PLUS G      Adding 1 (the successor function)
H                   produces the other phantom
                    numbers in order.
      H PLUS G
A
      A PLUS G
D
      D PLUS G
J
      J PLUS G
B
      B PLUS G
E
      E PLUS G
F
      F PLUS G      Note that this simple interaction
I                   sequence is highly directed and
                    may not be representative of
                    the typical student's work.

                    Checking the answers:
      G = 1
1
      C = 2
1
      H = 3
1
      A = 4
1
      D = 5
1
      J = 6
1
      B = 7
1
      E = 8
1
      F = 9
1
      I = 10
0                   Whoops.

      I = 0         There, that's right.
1
```



COMPUTER PROGRAMS

The computer programs used for playing the game of PHANTNUM are written in A Programming Language (APL)* and may be executed on any standard APL computer system.

The main program for introducing the game is shown below:

```
     ∇PHANTNUM
[1]  'BEWARE . . . . . THE NUMBER PHANTOM IS
     HERE!'
[2]  MASK
[3]  'I HAVE PUT MASKS ON THE NUMERALS'
[4]  ' 0 1 2 3 4 5 6 7 8 9 '
[5]  'SO YOU CAN'T RECOGNIZE THEM.'
[6]  ''
[7]  'THE MASKS ARE:'
[8]  ' A B C D E F G H I J'
[9]  'AND YOU CAN USE THEM LIKE NUMBERS'
[10] 'WITH PLUS MINUS AND TIMES LIKE THIS:'
[11] ''
[12] '   B PLUS C'
[13] ''
[14] '  H MINUS B '
[15] ''
[16] '  C TIMES E '
[17] ''
[18] 'BUT I WILL ONLY SHOW YOU THE LAST
     LETTER'
[19] 'AND WILL LEAVE OFF THE NEGATIVE
     SYMBOL'
[20] 'IN THE ANSWER.'
[21] ''
[22] 'GO AHEAD.  TRY SOME YOURSELF.'
[23] '. . . . . HEH . . . HEH . . . . . '
[24] 'YOU'LL NEVER FIGURE OUT WHICH ARE
     WHICH!'
     ∇
```

*APL is a relatively new general-purpose computer programming language invented by Dr. Kenneth E. Iverson of IBM. Originally conceived as a unifying mathematical notation, APL has since been applied in the fields of business, scientific research, and education.

Most of this program displays expressions (written in between quotation marks). Line [2] calls for sub-program MASK.

Sub-program MASK makes arbitrary assignments of the masks A B C D E F G H I J to the numerals, as shown below:

```
     ∇MASK
[1]  A ← 4          A is assigned the numeral 4
[2]  B ← 7          B is assigned the numeral 7
[3]  C ← 2          C is assigned the numeral 2
[4]  D ← 5          D is assigned the numeral 5
[5]  E ← 8          E is assigned the numeral 8
[6]  F ← 9          F is assigned the numeral 9
[7]  G ← 1          G is assigned the numeral 1
[8]  H ← 3          H is assigned the numeral 3
[9]  I ← 0          I is assigned the numeral 0
[10] J ← 6          J is assigned the numeral 6
     ∇
```

Note that if the game is repeated, these same assignments will be made by MASK.

The programs which perform the arithmetic functions on the phantom numbers are defined below:

Program PLUS produces an ANSWER by first numerically adding the values of the two phantom numbers given (X + Y). Then, that result (which may be greater than 9) is divided by 10 to yield a remainder (| is the APL symbol for the remainder function) which will always be one of the numerals 0 1 2 3 4 5 6 7 8 9. This remainder is used as an index [ ] to select the appropriate one of the letters 'IGCHADJBEF' for an ANSWER:

```
     ∇ ANSWER ← X PLUS Y
[1]  ANSWER ← 'IGCHADJBEF'[10|X+Y]
     ∇
```

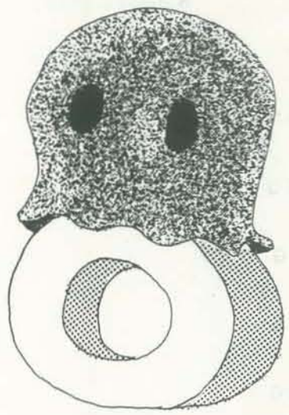
Program MINUS is similar to PLUS except that the absolute value of the difference of the values of the phantom numbers [| X-Y] is used to select a letter from 'IGCHADJBEF' for an ANSWER:

```
     ∇   ANSWER ← X MINUS Y
[1]  ANSWER ←'IGCHADJBEF' [| X-Y]
     ∇
```

Program TIMES is identical to PLUS except that X and Y are multiplied:

```
     ∇   ANSWER ← X TIMES Y
[1]  ANSWER ← 'IGCHADJBEF'[10|X x Y]
     ∇

     □ IO ← 0
```

Note that an Index Origin of 0 is used here so that the indices 0 1 2 3 4 5 6 7 8 9 will correspond directly to the letters 'IGCHADJBEF.'



FURTHER APPLICATIONS

The game of PHANTNUM can lead naturally to more formal study of elementary functions by examining patterns in *function tables* such as addition, subtraction, and multiplication tables. Other extensions of the game of PHANTNUM include: adding other functions, such as DIVIDE (quotient only) or a MODULUS function (remainder) or the POWER function (exponentiation); changing the base of the number system, say to a number less than 10; allowing multiple-digit phantom numbers and negative phantom numbers and decimal phantom numbers (in both inquiries and results); encouraging compound expressions, such as (A PLUS B) TIMES (C PLUS D), fully parenthesized to indicate the order of operation explicitly; striving to determine the phantom numbers in a minimum number of experiments; making random assignments of masks, so that the phantom numbers are different from game to game; having the player's conjectures checked under computer control, perhaps pointing out those which have been discovered correctly and those which have not; and including hints and options to stop and start again. □

RELATED REFERENCES

Elliot, PC, "Elementary Mathematics Teacher Training Via *A Programming Language*", (doctoral dissertation), University of Massachusetts, 1973.

Iverson, KE, *A Programming Language*, Wiley, 1962.

Iverson, KE, *Elementary Algebra: An Algorithmic Treatment*, Addison-Wesley, 1974.

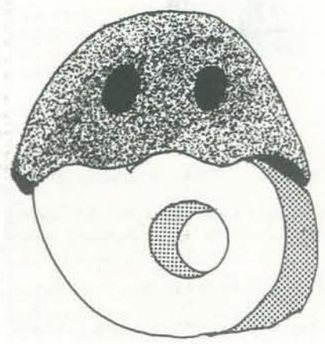
Peelle, HA, "COMPUTER GLASS BOXES: Teaching Children Concepts with A Programming Language", *Educational Technology*, Vol XIV, No 4, April, 1974.

RANDOM PATTERNS

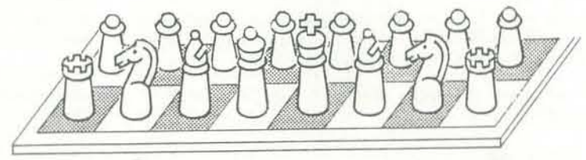
BY RAMON M ZAMORA

For those of you who like to sit up late at night and watch test patterns, here is something you may appreciate. The accompanying routine will fill your TRS-80 screen with a random pattern composed of a single character from the character set. At the question mark just input a number. This number will be translated into a character string of random length and then displayed. The numbers in the graphic code sequence (129-191) are especially interesting. For example, try 170, 179, 188 and 191. Hitting the ENTER key will halt the program; hitting the key again will cause another pattern to be generated, beginning at the top of the screen. Try this sequence and see what happens: 23 ENTER, ENTER, 179 ENTER (let the screen fill), ENTER, 28 ENTER. Explain that result to yourself.

```
100 I=0
105 INPUT M
110 N=RND(15):PRINT@ I,STRING$(N,CHR$(M));
120 PRINT,: I=I+N+RND(5): IF I>1012 THEN I=960
125 A$=INKEY$:IF A$<>"" THEN 100
130 GOTO 110
```



Some Guidelines for Microcomputer Chess



BY JIM DAY

A number of chess programs are available for microcomputers. (Please don't write to me for tapes or listings; I'm not a franchised distributor.) Some of these are pretty good, but none plays at a Grand Master level and some play a very poor game indeed. As the man said of the talking dog, 'It isn't done well, but I'm surprised to find it done at all.'

Regardless of the algorithm used to pick the next move, I believe that the following guidelines should be observed. These may appear obvious, but I've seen each of them ignored by one chess program or another.

First, the board should be displayed properly, with the black corners at the lower left and upper right and each queen on her own color.

The program should always remember whose turn it is and should prompt the human player to input a move at the appropriate time.

Also, the program should make no illegal moves, nor should it accept an illegal move by its opponent. Incredible as it may seem, some otherwise excellent chess programs do little or no checking for illegal moves by the human player. When an illegal or impossible move is input, the player should be informed of his or her mistake and prompted to input another move. Most illegal moves can be categorized as follows:

1. Moving to or from a nonexistent square, from an empty square, from a square occupied by the opponent, or to a square occupied by another piece of like color.
2. Moving so as to leave the king in check or to place the king in check.
3. Moving a piece 'through' another one (only a knight may do so).
4. Moving a piece in a direction that is illegal for it, such as moving a rook diagonally.
5. Illegal castling. Some programs don't allow castling at all, while others allow it only in the king's side. If castling is to be allowed, the usual rules of chess should be observed.

The player should be informed of a check, checkmate, or stalemate condition as it occurs, and all output to a CRT should be displayed for a reasonable length of time. Ten milliseconds isn't reasonable, even for a speed reader.

Finally, all dialogue should be trustworthy, helpful, friendly, and courteous. I've seen more than one prospective enthusiast turned off by 'cute' but insulting dialogue from a game program. Nobody wants to be put down by a machine, even in fun, and many people are easily intimidated by computers. □

A Bibliography on Computer Chess

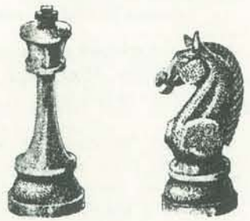
COMPILED BY ERYK VERSHEN

This is in no sense a complete bibiography. A reasonably complete one would run four hundred or more items. Instead, the items in this bibiography have been chosen both for their importance and their accessibility. My intent has been to provide a short, 'to the point' guide to the literature of computer chess. With that in mind only a few items on chess *per se* have been included.

GENERAL BOOKS

Frey, Peter W. (Ed.). *Chess Skill in Man and Machine*. New York: Springer-Verlag. 1977.
This is the best single book on computer chess that I know of. It covers more of the possible approaches to the problem than any other work. This book is good first reading for 'computer types.'

Levy, David. *Chess and Computers*. Woodland Hills, CA : Computer Science Press. 1976.



Newborn, Monroe. *Computer Chess*. New York: Academic Press. 1975.
While being general introductions to computer chess, both of these books are mainly historical. Also, despite the congruence of subject matter both are worth reading, as histories. The Levy book is probably the best first reading for 'chess types.'

OTHER BIBLIOGRAPHIES

Marsland, T.A. 'A Comprehensive List of Computer Chess Literature.' Edmonton: University of Alberta. 1977.
This is the best bibliography available. Copies may be obtained by sending $1 (for costs) and requesting 'technical report TR77-4' from: Computing Science Department, University of Alberta, Edmonton, Alberta T6G 2H1 CANADA.

Two other bibliographies are contained in the Frey and Levy books mentioned under general books.

ARTICLES AND TECHNICAL BOOKS

Adelson-Velsky, G.M., Arlazarov, V.L., & Donskoy, M.V. 'Some Methods of Controlling the Tree Search in Chess Programs.' *Artificial Intelligence*, Vol 6, 1975, pp.361-371.
This is a paper by the KAISSA Programmers. They haven't done many papers, but this one is fairly typical of their style.

Atkin, R.H. 'Multi-dimensional Structure in the Game of Chess.' *International Journal of Man-Machine Studies*, Vol 4, 1972, pp.341-362.
Atkin employs a highly mathematical approach. It involves representing positions by figures in a multi-dimensional space. The article is interesting reading if you can handle the math.

Baylor, G.W. & Simon, H.A. 'A Chess Mating Combinations Program.' Proceedings of the Spring Joint Computer Conference, April, 1966. pp. 431-447.
This article describes the MATER program. This is a special purpose program that searches for checkmates. It is later used in the COKO program.

Bell, A.G. 'How to Program a Computer to Play Legal Chess.' *Computer Journal*. Vol 13, 1970, pp. 208-219.
This is the first of the only two published chess programs. This one uses table driven move generation. The other program is by J.R. Manning.

Berliner, H.J. 'Some Necessary Conditions for a Master Chess Program.' Proceedings of the 3rd International Joint Conference on Artificial Intelligence, 1973, pp.77-85.
Berliner is a highly rated chess player and a computer scientist. This is an important article and is definitely 'must' reading.

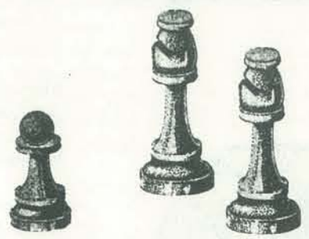
Berliner, H.J. 'Search and Knowledge.' Proceedings of the International Joint Conference on Artificial Intelligence, 1977, pp. 975-979.
This is another good paper by Berliner. This one discusses the spectrum of choices in chess programming from procedures which know nothing and search like mad to procedures which use knowledge of chess and search very little.

Bernstein, A. and Roberts, M. DeV. 'Computer vs. Chess Player.' *Scientific American*, June, 1958, pp. 96-105.
This article describes one of the first computer chess programs.

Botvinnik, M.M. *Computers, Chess and Long-Range Planning*. New York: Springer-Verlag. 1970.
Botvinnik is a former World Champion in chess. This book describes his own, rather mathematical, approach to computer chess. He is reported to be working on a program at this point.

Clarke, M. (ed.) *Advances in Computer Chess I*. Edinburgh: Edinburgh University Press. 1977.
This is the first in (hopefully) a yearly series of papers on computer chess by people working in the field. It is definitely worth reading.

Clarke, M.R.B. 'Some Ideas for a Chess Compiler.' *Artificial and Human Thinking*. (Elithorn & Jones, eds.), Amsterdam: Elsevier, 1973, pp. 189-198.
This is a proposal for an ALGOL based language devoted to writing computer chess programs. The language has been implemented as a preprocessor.



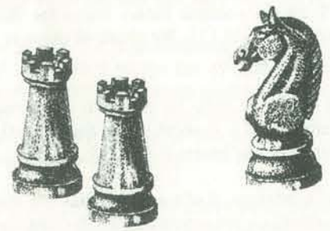
DeGroot, A.D. *Thought and Choice in Chess*. The Hague: Mouton. 1965.
This is the classic work on human thought processes while playing chess. It is usually referenced to show how far computers are behind human players.

Evans, L. & Korn, W. *Modern Chess Openings*. New York: Pitman. 1965.
Most chess programs use an opening 'book.' These are often obtained from this source.

Gillogly, J. 'The Technology Chess Programs.' *Artificial Intelligence*. Vol 3, 1972, pp.145-163.
This article describes the TECH program. TECH placed well in the first few ACM tournaments, but it has since failed to keep up and no longer competes.

Good, I.J. 'A Five Year Plan for Automatic Chess.' *Machine Intelligence 2* (Dale & Michie, eds.) Edinburgh: Oliver & Boyd, 1968, pp. 89-118.
This article contains some early speculations on how to go about developing a high quality chess program; it is still relevant.

Greenblatt, R.D., Eastlake, D.E. & Crocker, S.D. 'The Greenblatt Chess Program.' Proceedings of the Fall Joint Computer Conference. Montvale, NJ: AFIPS Press, 1967, pp. 801-810.
Greenblatt's program was the first to perform respectably against humans in tournament play. It has quite possibly played more games than any other program, but it has never competed in any computer chess tournaments.



Harkness, K. *Official Chess Handbook*. New York: David McKay, 1967.
This book is a useful compendium of information about chess. It contains a copy of the official laws of chess, which bear perusal if you want to be sure your program does everything correctly.

Harris, L.R. 'The Heuristic Search and the Game of Chess.' Proceedings of the 4th International Joint Conference on Artificial Intelligence, 1975, pp. 334-339.
Harris is of the school that says it is better to search a little in the right places than to search a lot indiscriminantly. It is easier said than done.

Huberman, B.J. 'A Program to Play Chess Endgames.' Stanford Technical Report CS 106 (also Ph.D. Thesis) Stanford University, 1968.
A classic paper on an endgame program.

Kister, J., Stein, P., Ulam, S., Walden, W. & Wells, M. 'Experiments in Chess.' *Journal of the ACM*, Vol. 4, 1957, pp. 174-177.
This was probably the first computer chess program. Most of the experiments in this article were done with a 6x6 board, however.

Knuth, D.E. & Moore, R.W. 'An Analysis of Alpha-Beta Pruning.' *Artificial Intelligence*, Vol. 6, 1975, pp. 293-326.
This is an excellent paper on the alpha-beta algorithm; in fact, it is absolutely indispensible for a thorough understanding of the algorithm.

Kotov, A. 'Think Like a Grandmaster.' *Chess Digest*, Dallas, 1971.
In this article, Kotov reflects on what he had to learn to become a grandmaster. This a very good article.

Kozdrowicki, E.W. & Cooper, D.W. 'COKO III & the Future of Inter-Snap Judgment Communication.' Proceedings of ACM 73, 1973, pp. 213-218.
COKO is another program from the early tournaments. It never did that well and eventually dropped out, but the ideas involved are of interest.

Levy, D.N.L. & Hayes, J.E. *The World Computer Chess Championship*, Edinburgh: Edinburgh University Press, 1976.
Levy is certainly the most prolific commentator on computer chess. This book's main value is the descriptions of the programs.

Manning, J.R. 'White to move and mate in N moves.' *Computer Journal*, Vol. 14, 1971, pp. 209-213.
This is the other of the two published chess programs. It uses the more common extended board representation, unlike Bell's program. As written, all it does is look for a mate in N moves by searching exhaustively.

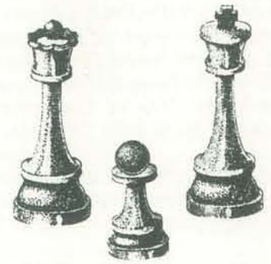
Newell, A., Shaw, J.C. & Simon, H.A. 'Chess-playing Programs and the Problem of Complexity.' *IBM Journal of Research & Development*, Vol. 2, 1958, pp. 320-335.
Here is another early program. This was the first to operate on a goal oriented basis, rather than brute force searching.

Pitrat, J. 'A Chess-combination program which uses plans.' *Artificial Intelligence*, Vol. 8, 1977, pp. 275-321.
Most people feel that a master level chess program will have to use some sort of chess knowledge. This is one approach.

Shannon, C.E. 'A Chess-playing Machine.' *Scientific American*, Feb. 1950, pp. 48-51.
This is one of the first papers on computer chess. Shannon never wrote a program, but he did lay some groundwork.

Sunnucks, A. *The Encyclopedia of Chess*. New York: St. Martins Press, 1970.
Here is another good compendium on chess. It also contains the official rules.

Zobrist, A.L. & Carlson, F.R., Jr. 'An Advice-taking Computer.' *Scientific American*, June 1973, pp. 93-103.
This is one of the several projects attempting to develop a good program by having the program be able to take advice.



MISCELLANEOUS NOTES

Two Artificial Intelligence newsletters, the SIGART Newsletter in the US, and FIRBUSH News in England, regularly carry articles on computer chess. The *Computer Chess Newsletter* started by Doug Penrod has since been taken up by *Personal Computing* magazine as a monthly column. It first appeared in the April, 1978 issue. The two issues of the newsletter that appeared prior to that are also available from *Personal Computing*.

David Levy is planning to publish another book on computer chess in early 1979. And, since the second Advances in Computer Chess Conference has been held, we can probably expect another volume in that series also.

This bibliography only covers a smattering of the literature. Indeed, several people who have made considerable contributions to the field are not even mentioned. Among them are L. Atkin, M.A. Bramer, C. Kalme, D. Michie, D. Slate, and S.T. Tan. In spite of the large amount of effort that has been expended on computer chess, many ideas still remain untried, and many ideas have not yet been implemented successfully; so do not despair of making a contribution. □

DISTANCE and ERROR CORRECTING CODES

BY TOM DWYER & MARGOT CRITCHFIELD

We are again indebted to Tom Dwyer and Margot Critchfield for this spy-fi code cracking game. Your challenge is to write a BASIC program to detect authentic code names for a notorious ring of glue thieves.

In the last issue of PC, *Tom and Margot gave us their evaluation of the TRS-80 Level II BASIC. Tom Dwyer is Professor of Computer Science at the University of Pittsburgh, and Margot Critchfield holds degrees in anthropology and education, and is currently a doctoral student at the University of Pittsburgh.*

Both are directors of SOLOWORKS, a project designed to involve kids in learning to program and use computers in creative problem solving. Tom and Margot have authored an outstanding book, BASIC and the Personal Computer *recently released in the Addison-Wesley "Joy of Computing" Series. – CB*

A good way to make an otherwise difficult (and/or dull) subject come alive is to associate it with a little fantasy. Let's examine how this might be done in the area of binary information codes.

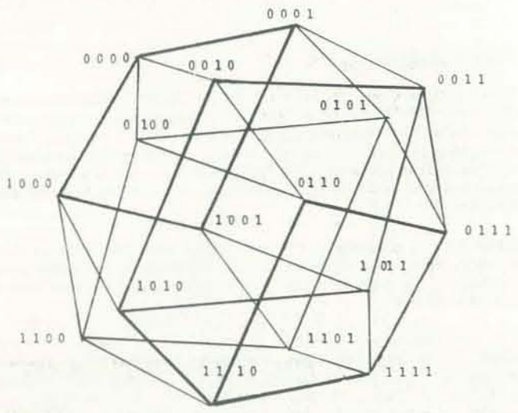


Figure 1. A 4-dimensional cube.

A binary code of length N is a string of N 0's or 1's. For example, if N = 3, all the possible binary codes are 000, 001, 010, 011, 100, 101, 110, and 111. We speak of these as 3 BIT codes (1 and 0 are called BITS).

These codes could be used to represent eight objects of any sort – the members of a musical octet, the digits 0,1,2,3,4,5,6, 7 in a computer, or the letters A,B,C,D,E,F,G,H.

Now for some intrigue. Let's assume that we wish to assign binary code names to the agents of STICK (Society to Increase Contact for Keeps), an international ring of glue thieves. Suppose we only have two agents but eight codes. Question: Can we assign codes so that:

(a) The computer will check code authenticity *without* knowing the correct codes.
(b) The computer can give the correct code even though the agent has deliberately changed one BIT (to throw off eavesdroppers).

To see how codes can be assigned to make this possible, let's place the codes at the vertices of a cube.

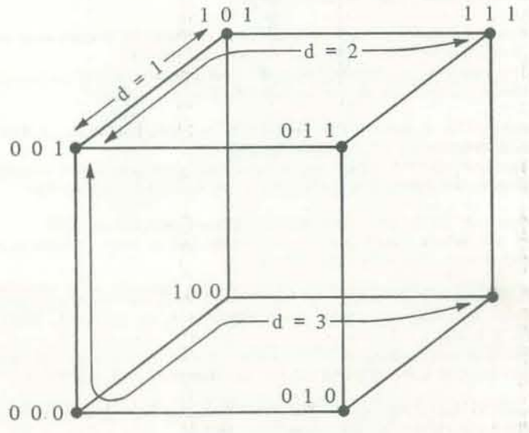


Figure 2. A 3-dimensional cube.

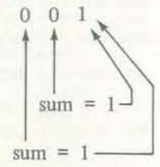
To be more precise, we should call Figure 2 a "3-dimensional cube." A picture of a "4-dimensional cube" (which has 2^4 = 16 vertices) is shown in Figure 1. Thus we can associate a unique four-bit code with each vertex of a 4-D cube. (Can you generalize this statement?)

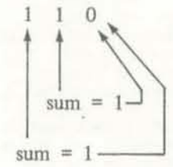
We will define the number of BITS by which two codes differ as the "DISTANCE" (=d) between these codes. Thus, for example, the distance between 001 and 101 is d = 1, the distance between 001 and 111 is d = 2, and the distance between 001 and 110 is d = 3. (Math students: Is this a legal use of the word distance?) Notice that our picture has been drawn so that "distance" between codes corresponds to the number of edges of the cube you would have to walk along to get from one vertex to the other.

Let's assign our two authentic agents the codes 001 and 110 (which are a distance of three from each other). Now suppose one agent walks up to another and says my code is 101.

(a) How can we tell if it is an authentic code? One way would be to simply compare it to the list of authentic codes! However, if there were to be very many codes, such a search of the authentic list would be time consuming. Besides, we don't want this authentic list stored in too many places! There is another way to check authenticity.

In our example the two authentic codes have the property that if we add the first and third BITS of the code we get 1, and this is *also* true if we add the second and third BITS.





For all other codes this is false.

NOTE: 0 + 1 = 1 and 1 + 0 = 1 in binary arithmetic.

ALSO NOTE: 0 + 0 = 0, but 1 + 1 = 0 (with carry of 1)

FURTHER: 0 * 1 = 0 1 * 0 = 0 0 * 0 = 0 and 1 * 1 = 1

Thus the code 1 0 1 our agent gave is not authentic.

Since: 1 + 1 = 0

(b) Suppose our agent deliberately changes one BIT in his code when giving it verbally. Applying the above rule will *detect* the error, but can we figure out what the correct code should have been?

We can see the answer from our diagram. An authentic code which has only one BIT changed is distance d = 1 from the original correct code, but distance d = 2 from the *other* correct code. Thus 101 has to be corrected back to 001, *not* to 110.

Let's now try to develop an algorithm for making corrections in codes which have 1 BIT in error: Here is how we might reason:

Let's call the 3 BITS: B1, B2, and B3.

FOR 101: B1 + B3 = 0 WRONG
B2 + B3 = 1 RIGHT
∴ Change B1, Correct code is : 001

FOR 111: B1 + B3 = 0 WRONG
B2 + B3 = 0 WRONG
∴ Change B3, Correct code is: 110 etc.

To see if you follow this, try writing programs for your wrist computer based on the following ideas or questions.

1. The Two-Agent Problem
For solution see page 54-55.
Authentic Codes: 101 and 010

INPUT: Any 3 BIT code which is either an authentic code, or which contains an error in 1 BIT.
OUTPUT: The Message: "AUTHENTIC CODE"
OR: "CODE IN ERROR
CODE SHOULD BE ___"

Can use of a 4 BIT code (see Figure 1) permit additional outputs for the above"Two-Agent" analysis program?

2. Four-Agent Problem
For solution see page 54-55.
Authentic Codes are:

SMITH	0 0 0 0 0
BOND	1 1 1 0 0
SPIRO	0 0 1 1 1
JONES	1 1 0 1 1

INPUT: Any code
OUTPUT:
The Message: "AUTHENTIC CODE"
OR: "1 BIT ERROR–CORRECT CODE IS ___"
OR: "ERROR ≥2 BITS–DOUBLE AGENT"

Here is a set of six BIT codes to play with: 000000, 000111, 111000, 110110. 011011, 101101. (NOTE: d≥3 for any two of these codes.) □

H-A-N-G-M-A-N

BY RON SANTORE

For you 8080 machine language purists, Ron Santore to the rescue! Try Ron's version of Hangman on your homebrew or other 8080 system with the tiny memory. – The Dragon

Ron is owner of the Byte Shop in San Luis Obispo, California. He, Don Inman, and Bob Albrecht have authored a beginners' guide to 8080 machine language for the Heathkit H-8 Microcomputers. The book is planned to be released by dilithium Press around the first of the year. – CB

The trouble with most machine language game programs is they're usually very long and not all that much fun to play. Here's one that won't wear your finger tips down too much when you type it in, and yet it is a game you and your friends won't get tired of. The game of Hangman is entertaining, and it's also instructive – you'll have to bribe the kids to get them away from this one.

The program is 8080 machine language and occupies 1K of memory. Basically, the computer chooses a word from a large table of words and lets you try to guess it. Every time you guess a letter wrong, the computer awards you one letter from the word H-A-N-G-M-A-N. If the computer can spell HANGMAN before you guess the hidden word . . . you lose.

The input subroutine is located at address 001,023 to 001,077. Extra room is left there in case you have to modify the routine. The output subroutine is at 001,113 to 001,135; and here too extra room is left for you.

When entering the program into memory, be sure to get the data in correctly from 000,000 to 001,200 – these op codes are important. If you make a mistake entering the codes past 001,200 it will result in a misspelled word, but the program will still run.

The words used in the game are stored from 002,200 to 003,377. You may change any of these words, but follow two rules: the words must not be longer than seven letters each, and there must be one (and only one) 377 between any two words. Happy Computing . . .

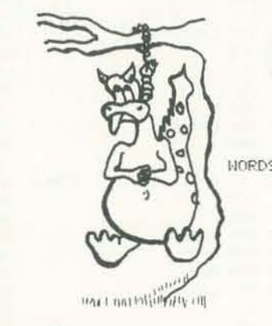
```
BEGIN:  000,000 061 LXI SP
                000
                004
        000,003 001 LXI BC
                200
                001
        000,006 315 CALL
                100
                001
INIT :  000,011 041 LXI HL
                150
                001
        000,014 076 MVI A
                055
        000,016 167 MOV M,A
        000,017 175 MOV A,L
        000,020 376 CPI
                177
        000,022 312 JZ
                021
                000
        000,025 043 INX HL
        000,026 303 JMP
                014
                000
        000,031 076 MVI A
                377
        000,033 062 STA
                157
                001
NNWRD:  000,036 052 LHLD
                136
                001
        000,041 021 LXI DE
                160
                001
        000,044 176 MOV A,M
        000,045 022 STAX DE
        000,046 376 CPI
                377
        000,050 312 JZ
                060
                000
        000,053 043 INX HL
        000,054 023 INX DE
        000,055 303 JMP
                044
                000
        000,060 353 XCHG
        000,061 315 CALL
                000
                001
        000,064 167 MOV M,A
        000,065 023 INX DE
        000,066 032 LDAX DE
        000,067 376 CPI
                377
        000,071 302 JNZ
                102
                000
        000,074 041 LXI HL
                100
                002
        000,077 303 JMP
        000,100 102 MOV B,E
        000,101 000 NOP
        000,102 353 XCHG
        000,103 042 SHLD
                136
                001
GUESS:  000,106 036 MVI E
                000
        000,110 001 LXI BC
                242
                001
        000,113 315 CALL
                100
                001
        000,116 315 CALL
                022
                001
        000,121 127 MOV D,A
        000,122 041 LXI HL
                160
                001
        000,125 176 MOV A,M
        000,126 376 CPI
                377
        000,130 312 JZ
                142
                000
        000,133 272 CMP D
```



```
        000,134 212 JZ
                154
                000
        000,137 043 INX HL
        000,140 303 JMP
                125
                000
        000,143 173 MOV A,E
        000,144 376 CPI
                000
        000,146 312 JZ
                167
                000
        000,151 303 JMP
                242
                000
        000,154 315 CALL
                000
                001
        000,157 162 MOV M,D
        000,160 315 CALL
                012
                001
        000,163 034
        000,164 303 JMP
                125
                000
NO :    000,167 001 LXI BC
                312
                001
        000,172 315 CALL
                100
                001
        000,175 041 LXI HL
                150
                001
        000,200 176 MOV A,M
        000,201 376 CPI
                055
        000,203 312 JZ
                212
                000
        000,206 043 INX HL
        000,207 303 JMP
                200
                000
        000,212 315 CALL
                011
                001
        000,215 176 MOV A,M
        000,216 315 CALL
                000
                001
        000,221 167 MOV M,A
        000,222 036 MVI E
                000
        000,224 041 LXI HL
                150
                001
        000,227 315 CALL
                254
                000
        000,232 173 MOV A,E
        000,233 376 CPI
                001
        000,235 312 JZ
                106
                000
        000,240 303 JMP
                225
                000
YES:    000,243 001 LXI BC
                266
                001
        000,246 315 CALL
                100
                001
        000,251 036 MVI E
                000
GSSPT:  000,252 041 LXI HL
                170
                001
        000,256 315 CALL
                254
                000
        000,261 173 MOV A,E
        000,262 376 CPI
                000
        000,264 302 JNZ
                227
                000
        000,267 001 LXI BC
                000
                002
        000,272 315 CALL
                100
                001
        000,275 000 NOP
        000,276 000 NOP
        000,277 000 NOP
AGN ?:  000,300 001 LXI BC
                021
                002
        000,303 315 CALL
                100
                001
        000,306 315 CALL
                022
                001
        000,311 376 CPI
                121
        000,312 212 JZ
                011
                000
        000,316 001 LXI BC
                042
                002
        000,321 315 CALL
                100
                001
        000,324 303 JMP
                000 } END OF GAME.
                251 } PUT YOUR ADDR. HERE.
        000,327 000 NOP
        000,330 000 NOP
        000,331 000 NOP
        000,332 303 JMP
                106
                000
ULOSE:  000,335 001 LXI BC
                237
                001
        000,340 315 CALL
                100
                001
        000,343 041 LXI HL
                160
                001
        000,346 315 CALL
                254
                000
        000,351 303 JMP
                200
                000
BFFPT:  000,354 176 MOV A,M
        000,355 376 CPI
                377
        000,357 310 RZ
        000,360 376 CPI
                055
        000,362 302 JNZ
                367
                000
        000,365 036 MVI E
                001
        000,367 315 CALL
                112
                001
        000,372 043 INX HL
        000,373 302 JMP
                254
                000
        000,376 000 NOP
        000,377 000 NOP
INRL8:  001,000 054
        001,001 054
        001,002 054
        001,003 054
        001,004 054
        001,005 054
        001,006 054
        001,007 054
        001,010 311 RET
DCRL8:  001,011 055 DCR L
        001,012 055 DCR L
        001,013 055 DCR L
        001,014 055 DCR L
        001,015 055 DCR L
        001,016 055 DCR L
        001,017 055 DCR L
        001,020 055 DCR L
        001,021 311 RET
INPUT:  001,022 000 NOP
        001,023 000 NOP
        001,024 000 NOP
        001,025 000 NOP
        001,026 000 NOP
        001,027 000 NOP
        001,030 000 NOP
        001,031 000 NOP
        001,032 000 NOP
        001,033 000 NOP
        001,034 000 NOP
        001,035 000 NOP
        001,036 000 NOP
        001,037 000 NOP
        001,040 000 NOP
        001,041 000 NOP
        001,042 333 IN
                000
        001,044 017 RRC
        001,045 332 JC
                042
                001
        001,050 333 IN
                001
        001,052 376 CPI
                101
        001,054 372 JM
                064
                001
        001,057 376 CPI
                122
        001,061 372 JM
                075
                001
        001,064 001 LXI BC
                070
                002
        001,067 315 CALL
                100
                001
        001,072 303 JMP
                042
                001
        001,075 323 OUT
                001
        001,077 311 RET
TXTPT:  001,100 012 LDAX BC
        001,101 376 CPI
                377
        001,103 310 RZ
        001,104 315 CALL
                113
                001
        001,107 003 INX BC
        001,110 303 JMP
                100
                001
OUTPT:  001,113 365 PUSH SW
        001,114 333 IN
                000
        001,116 007 RLC
        001,117 332 JC
                114
                001
        001,122 361 POP SW
        001,123 323 OUT
                001
        001,125 311 RET
        001,126 000 NOP
        001,127 000 NOP
        001,120 000 NOP
        001,121 000 NOP
        001,122 000 NOP
        001,123 000 NOP
        001,124 000 NOP
        001,125 000 NOP
LTABL:  001,126 100 @
HTABL:  001,127 002
        001,140 110 H
        001,141 101 A
        001,142 116 N
        001,143 107 G
        001,144 115 M
        001,145 101 A
        001,146 116 N
        001,147 377
HNGBF:  001,150 000
        001,151 000
        001,152 000
        001,152 000
        001,154 000
        001,155 000
        001,156 000
        001,157 377
WPDBF:  001,160 000
        001,161 000
        001,162 000
        001,163 000
        001,164 000
        001,165 000
        001,166 000
        001,167 377
GSSBF:  001,170 000
        001,171 000
        001,172 000
        001,172 000
        001,174 000
        001,175 000
        001,176 000
STOP
        001,200 015
        001,201 012
        001,202 052 *
        001,202 052 *
        001,204 040
        001,205 110 H
        001,206 101 A
        001,207 116 N
        001,210 107 G
        001,211 115 M
        001,212 101 A
        001,213 116 N
        001,214 040
        001,215 052 *
        001,216 052 *
        001,217 015
        001,220 012
        001,221 111 I
        001,222 040
        001,223 110 H
        001,224 101 A
        001,225 126 V
        001,226 105 E
        001,227 040
```

```
001,220 101 A
001,221 040
001,232 127 W
001,222 117 O
001,224 122 R
001,225 104 D
001,226 056 .
001,227 377
001,240 015
001,241 012
001,242 015
001,243 012
001,244 107 G
001,245 125 U
001,246 105 E
001,247 123 S
001,250 122 S
001,251 040
001,252 101 A
001,253 040
001,254 114 L
001,255 105 E
001,256 124 T
001,257 124 T
001,260 105 E
001,261 122 R
001,262 072 :
001,263 040
001,264 377
001,265 377
001,266 015
001,267 012
001,270 107 G
001,271 117 O
001,272 117 O
001,273 104 D
001,274 041 !
001,275 040
001,276 040
001,277 040
001,200 121 Y
001,201 117 O
001,202 125 U
001,203 040
001,204 110 H
001,205 101 A
001,206 126 V
001,207 105 E
001,210 040
001,211 377
001,212 377
001,213 015
001,214 012
001,215 116 N
001,216 117 O
001,217 120 P
001,220 105 E
001,221 056 .
001,222 040
001,223 040
001,224 040
001,225 377
001,226 040
001,227 040
001,230 040
001,231 040
001,232 040
001,233 040
001,234 040
001,235 040
001,236 377
001,237 015
001,240 012
001,241 015
001,242 012
001,243 121 Y
001,244 117 O
001,245 125 U
001,246 040
```

```
001,247 114 L
001,250 117 O
001,251 123 S
001,252 124 T
001,253 056 .
001,254 015
001,255 012
001,256 040
001,257 040
001,260 040
001,261 040
001,262 124 T
001,263 110 H
001,264 105 E
001,265 040
001,266 127 W
001,267 117 O
001,270 122 R
001,271 104 D
001,272 040
001,273 127 W
001,274 101 A
001,275 122 S
001,276 040
001,277 377
002,000 015
002,001 012
002,002 015
002,003 012
002,004 131 Y
002,005 117 O
002,006 125 U
002,007 040
002,010 107 G
002,011 117 O
002,012 124 T
002,013 040
002,014 111 I
002,015 124 T
002,016 041 !
002,017 377
002,020 377
002,021 056 .
002,022 040
002,022 120 P
002,024 114 L
002,025 101 A
002,026 131 Y
002,027 040
002,030 101 A
002,031 107 G
002,032 101 A
002,033 111 I
002,034 116 N
002,035 077 ?
002,036 040
002,037 377
002,040 377
002,041 377
002,042 015
002,043 012
002,044 015
002,045 012
002,046 117 O
002,047 113 K
002,050 101 A
002,051 121 Y
002,052 054 ,
002,053 040
002,054 124 T
002,055 110 H
002,056 101 A
002,057 116 N
002,060 113 K
002,061 123 S
```



```
002,062 056 .
002,063 015
002,064 012
002,065 277
002,066 377
002,067 377
002,070 077 ?
002,071 040
002,072 377
002,073 377
002,074 377
002,075 377
002,076 377
002,077 377
002,100 115 M
002,101 131 Y
002,102 123 S
002,103 124 T
002,104 105 E
002,105 122 R
002,106 121 Y
002,107 377
002,110 102 C
002,111 117 O
002,112 123 S
002,113 115 M
002,114 111 I
002,115 103 C
002,116 377
002,117 123 S
002,120 125 U
002,121 116 N
002,122 123 S
002,122 105 E
002,124 124 T
002,125 377
002,126 103 C
002,127 117 O
002,130 116 N
002,131 116 N
002,132 105 E
002,133 103 C
002,124 124 T
002,135 377
002,126 112 J
002,127 125 U
002,140 120 P
002,141 111 I
002,142 124 T
002,143 105 E
002,144 122 R
002,145 377
002,146 122 R
002,147 105 E
002,150 101 A
002,151 104 D
002,152 105 E
002,152 122 R
002,154 377
002,155 104 D
002,156 111 I
002,157 123 S
002,160 116 N
002,161 105 E
002,162 121 Y
002,163 377
002,164 104 D
002,165 122 R
002,166 101 A
002,167 107 G
002,170 117 O
002,171 116 N
002,172 377
002,173 115 M
002,174 117 O
002,175 117 O
002,176 116 N
002,177 377
```

```
002,200 102 B
002,201 105 E
002,202 102 C
002,202 101 A
002,204 125 U
002,205 122 S
002,206 105 E
002,207 377
002,210 123 S
002,211 125 U
002,212 102 B
002,213 124 T
002,214 114 L
002,215 105 E
002,216 377
002,217 112 J
002,220 105 E
002,221 101 A
002,222 116 N
002,222 122 S
002,224 377
002,225 104 D
002,226 117 O
002,227 116 N
002,230 125 U
002,221 124 T
002,222 377
002,223 123 S
002,224 124 T
002,225 101 A
002,226 120 P
002,227 114 L
002,240 105 E
002,241 122 R
002,242 377
002,243 106 F
002,244 114 L
002,245 117 O
002,246 120 P
002,247 120 P
002,250 121 Y
002,251 377
002,252 102 C
002,253 114 L
002,254 117 O
002,255 103 C
002,256 112 K
002,257 377
002,260 102 C
002,261 101 A
002,262 102 B
002,262 111 I
002,264 116 N
002,265 105 E
002,266 124 T
002,267 377
002,270 120 P
002,271 117 O
002,272 103 C
002,273 112 K
002,274 105 E
002,275 124 T
002,276 377
002,277 124 T
002,200 117 O
002,201 105 E
002,202 277
002,303 103 C
002,304 117 O
002,305 115 M
002,206 120 P
002,307 125 U
002,210 124 T
002,311 105 E
002,312 377
002,313 124 T
002,314 126 V
002,315 377
```



```
002,216 114 L
002,317 111 I
002,220 107 G
002,221 110 H
002,222 124 T
002,323 277
002,224 124 T
002,325 101 A
002,326 120 X
002,327 377
002,230 114 L
002,331 117 O
002,232 123 S
002,332 105 E
002,224 122 R
002,235 377
002,336 122 R
002,337 101 A
002,240 104 D
002,241 111 I
002,242 117 O
002,343 377
002,344 116 N
002,345 117 O
002,346 111 I
002,247 123 S
002,350 131 Y
002,251 377
002,352 102 B
002,353 117 O
002,254 117 O
002,255 113 K
002,256 114 L
002,357 105 E
002,360 124 T
002,361 377
002,262 115 M
002,263 117 O
002,264 122 R
002,365 115 M
002,366 117 O
002,367 116 N
002,370 377
002,271 102 B
002,272 111 I
002,373 116 N
002,274 104 D
002,375 105 E
002,376 122 R
002,377 377
003,000 111 I
002,001 116 N
002,002 126 V
002,002 117 O
002,004 111 I
003,005 103 C
003,006 105 E
003,007 377
003,010 115 M
003,011 101 A
003,012 122 R
003,013 113 K
003,014 105 E
003,015 122 R
003,016 377
003,017 116 N
003,020 117 O
003,021 123 S
002,022 105 E
003,023 377
003,024 120 P
003,025 110 H
003,026 117 O
003,027 116 N
003,030 105 E
003,031 377
```

```
002,032 124 T
003,033 117 O
003,034 115 M
002,035 101 A
002,036 124 T
002,037 117 O
003,040 105 E
002,041 377
003,042 102 B
002,043 117 O
003,044 120 X
002,045 277
002,046 127 W
002,047 101 A
002,050 123 S
002,051 110 H
002,052 105 E
002,053 122 R
002,054 277
002,055 106 F
002,056 111 I
002,057 122 R
003,060 105 E
003,061 377
003,062 120 P
002,063 125 U
002,064 122 R
003,065 120 P
002,066 114 L
002,067 105 E
003,070 277
003,071 126 V
003,072 111 I
002,073 104 D
002,074 105 E
002,075 117 O
002,076 377
002,077 123 S
003,100 117 O
002,101 122 R
003,102 122 R
002,103 131 Y
002,104 377
003,105 124 T
002,106 110 H
002,107 122 R
003,110 117 O
002,111 101 A
002,112 124 T
002,113 277
002,114 124 T
002,115 125 U
002,116 105 E
003,117 123 S
003,120 104 D
003,121 101 A
003,122 131 Y
003,123 377
002,124 123 S
003,125 116 N
002,126 117 O
003,127 127 W
002,130 377
003,131 103 C
003,122 110 H
003,133 105 E
002,124 103 C
002,125 113 K
003,126 105 E
003,127 122 R
003,140 277
003,141 104 D
003,142 117 O
002,143 117 O
003,144 104 D
003,145 114 L
002,146 105 E
002,147 377
```

```
002,150 117 O
003,151 124 T
002,152 110 H
003,153 105 E
003,154 114 L
003,155 114 L
003,156 117 O
003,157 377
002,160 123 S
002,161 113 K
003,162 111 I
003,163 377
003,164 123 S
003,165 131 Y
003,166 123 S
003,167 124 T
003,170 105 E
003,171 115 M
003,172 377
003,173 102 B
003,174 117 O
003,175 117 O
003,176 113 K
003,177 277
002,200 115 M
002,201 125 U
002,202 122 S
002,203 111 I
003,204 103 C
002,205 377
003,206 120 P
003,207 105 E
002,210 117 O
002,211 120 P
003,212 114 L
003,213 105 E
003,214 377
003,215 107 G
002,216 125 U
003,217 111 I
003,220 104 D
003,221 105 E
003,222 377
003,223 123 S
003,224 120 P
002,225 105 E
003,226 103 C
003,227 111 I
003,230 101 A
003,231 114 L
003,232 377
003,233 105 E
003,234 116 N
003,235 112 J
003,226 117 O
002,237 131 Y
003,240 377
003,241 123 S
003,242 125 U
002,243 122 R
002,244 107 G
003,245 105 E
003,246 117 O
003,247 116 N
002,250 277
002,251 123 S
002,252 111 I
003,252 122 Z
003,254 122 Z
002,255 114 L
003,256 105 E
003,257 277
003,260 110 H
003,261 117 O
003,262 116 N
003,263 105 E
002,264 121 Y
003,265 377
```

```
002,266 112 J
003,267 101 A
003,270 116 N
003,271 125 U
002,272 101 A
003,273 122 R
003,274 121 Y
003,275 277
002,276 127 W
003,277 101 A
002,300 122 R
002,301 116 N
002,302 111 I
002,302 116 N
003,304 107 G
002,305 377
003,306 120 P
002,307 114 L
002,310 101 A
003,311 123 S
002,212 124 T
003,213 111 I
003,214 103 C
002,215 377
003,316 110 H
003,317 101 A
002,220 122 R
002,221 122 R
002,322 111 I
003,223 105 E
003,324 104 D
002,225 277
002,326 127 W
003,327 122 R
003,330 105 E
003,331 116 N
003,332 103 C
003,333 110 H
003,324 377
003,335 103 C
003,336 101 A
003,237 122 R
002,340 124 T
002,341 117 O
002,342 117 O
003,343 116 N
002,344 377
002,345 103 C
003,346 101 A
002,247 124 T
002,350 122 S
003,351 125 U
003,352 120 P
003,353 377
003,354 377
003,355 377
002,356 377
003,257 277
003,360 277
003,361 377
003,362 377
002,363 377
002,264 377
003,365 377
002,266 277
002,367 377
003,370 377
003,371 377
003,372 377
003,373 377
002,374 377
003,375 377
003,276 377
003,377 377
```

END

FORTRAN MAN

by Lee Schneider & Todd Voros

Volume III
Episode 5

As you may recall from our last episode, Our Hero, his memory partially erased while being smuggled into microprocessorland in PROM, has joined forces with the Underground Resistance Movement in their efforts to pull down the noisy regime instituted by the despotic Glitchmaster and reload the old system, thereby restoring peace and tranquility once again to the Land of the Little People!

After proving his powers in defense of the secret rebel base—powers which, in his semi-erased state, even *he* cannot explain—the rebel army marches onward under the capable leadership of Linea, who now leads her resistance elements to attack the stronghold of the Glitchmaster himself!

On their march across the data fields, a secret disc base is discovered—and, in a swift attack, the rebels capture the discs, load their object modules aboard, and take off for their targets at highest access speed!

But then . . . as they head over the Great Core Plains, their drives are suddenly neutralized . . . and the discs begin unexplicably to crash! Most are unhurt . . . but as they raise themselves up off of ground potential and recover from the unexpected system crash, a great shadow falls over the valley, and from out of this shadow materializes the Glitchmaster's most dreaded ally . . .

Yipe!

Look out!

Yes . . . it is none other than the Lockout Monster! Stolen from its rightful home with Clan McIntel, it now guards the data fields for the Glitchmaster . . . and is fast descending upon Our Hero and his friends . . .

The great beast makes its first pass at the little network of resistance . . . and without hesitation they all change sign bits and branch away in the other direction to avoid being disassembled permanently . . . and none too soon!

SNAP!!

All right, everybody . . . time for some high-speed RUN mode! !

Hastily they branch past the erased discs and scatter out into the data fields, with the Monster right in their tracks!

RUMBLE
RUMBLE
RUMBLE
RUMBLE
RUMBLE

Puff . . . puff . . .

We've . . . gotta rest!

Can't . . . that Monster is still tracing us! Come on . . .

Over hills, around voltage curves, past data stacks and across the many field boundaries, the chase goes on . . .

They go on . . . but gradually they tire and the Monster incessantly shifts ever closer to their location . . . then suddenly, F-Man turns and halts before them . . .

It's no good, friends . . . at this data rate you'll never make it to safety!

There's only one option . . . we must split up!

Farewell . . .

And before the group can execute another MOVE instruction or issue a COMMENT, Fortran Man has entered divergent series and . . .

Wha . . . where did he go?

He . . . he . . . *left* . . . back the way we came!

And look . . . the Monster is following him!

I . . . don't understand! Shouldn't we try and help him?

Yes . . . if we could . . . but I'm afraid we are in no position to support his code!

It was *his* choice . . . and if any one of us has a chance to survive the Monster . . . that stranger does!

Come on . . . let's get out of here!

For a microsecond they watch as the Monster pursues their newest ally over the hills and out of sight . . . then they turn and branch once again back onto the current path.

Onward they travel . . . and as the falling edge of the current cycle brings darkness over the land, they arrive at the site which had been designated as common meeting node for the Resistance in the event that anything go wrong . . .

Linea! Are we glad to see *you*! We were sure you had been locked out and destroyed by the Monster . . . what happened?

We *would* have been, if not for that stranger . . .

Speaking of him, has he come in yet? Where is he?

It is only then that the news is discovered . . .

I haven't seen him . . . anybody else here know where he is?

Nope . . .

Me neither . . .

I don't think *any* of us detected his RETURN . . .

Oh, no . . .

GASP! You mean . . .

Yes . . . I'm afraid so . . .

Our new friend shall be with us no longer . . . the Lockout Monster has once again taken away one of our files!

At least . . . we shall have the knowledge to honor him . . . for he gave his code for his country!

There is a quiet PAUSE . . . and then, abruptly, it is ended by a rudely executed Interrupt from behind . . .

Excuse me, folks . . . but are you perhaps looking for *me*?

GASP! It's . . .

Eh? Wha . . . ?

They all turn at the sound . . . and lo and behold, in strolls their missing comrade . . . looking decidedly un-terminated, and carrying with him a tiny memory block . . .

You were expecting maybe the Transistor Trio?

YOU!

But . . . we thought . . . . . how . . . where . . . ????

Are you all right?

And *what* is *that*???

And that's not the *only* surprise . . .

Oh, I'm fine, thanks . . .

And as far as *this* goes . . . Linea . . . meet the Lockout Monster!

querp?

WHAT??!!!!

But . . . that *can't* be . . . . . I mean, what *happened* to it?

As the others stare in amazement, F-Man explains . . .

It was very simple, really . . . don't know why I didn't compile it sooner!

Once I got it to chase me, I changed my external FORMAT and hid in the core planes . . . then as it made its second pass, I changed its DIMENSIONS . . . and when the compile completed, I picked the little devil up and off we went!

Amazing!

It's really very friendly, you know . . . once I convinced it that we were its friends and wanted only to return it to its rightful owners at Castle McIntel!

eeep?

Linea stares for only an instant . . . then turns to face her high-precision resistance commanders . . .

You see? I *told* you that this strange piece of code was unbreakable! and now, with the Monster out of the way . . .

We do what? We are too weak as we are now . . . most of our forces were trapped on the Core Plains after the drives crashed and were captured by the Glitchmaster's troops . . . and are now in Recombination Camps!

And if General Wirewound attacks without our support, his files will be wiped out!

Linea hesitates only for an instant . . . then turns once gain back to her newest ally . . .

All right then, stranger . . . I have a plan!

And you *and* your little friend there are in it!

What could be Linea's new plan to re-align the Resistance? Shall they arrive in time to prevent the General from exceeding his tolerance and overloading? Shall Microprocessorland ever know clear, noise-free freedom again?

Tune in again for the next episode . . . same time constant, same frequency!

The Case for APL in Education

BY HOWARD A. PEELLE

Dr. Peelle is an Associate Professor and Director of Instructional Applications of Computers at the School of Education, University of Massachusetts. He received his B.S. in Engineering from Swarthmore College in 1965 and his Doctorate in Education from the University of Massachusetts in 1971. Previously, he taught secondary mathematics and worked in testing computer-related instructional systems for Science Research Associates, Inc. (a subsidiary of I.B.M.). His research interests include uses of a programming language as a conceptual framework for teaching children.

Professor Peelle's article on an APL version of the mathematical game PHANTNUM is reprinted this issue. Look for it also.–RZ

APL is *A P*rogramming *L*anguage–a rather humble-sounding acronym for what some people now believe to be *the* programming language. Originally conceived by Kenneth Iverson [1] as a unifying mathematical notation, APL is not just a *computer* programming language, per se. Indeed, its main purpose is to *communicate*–clearly and efficiently–whether between man-and-man or between man-and-machine. APL programming is characteristically direct, succinct, powerful, and far removed from machine considerations; actually, the role of the computer is to support, not constrain, the language. APL has been implemented on a variety of computing systems–including IBM 360 and 370 series, IBM 5100 (desk-top), DEC PDP 10 and PDP 11, Hewlett-Packard 3000, Honeywell Sigma 7, Burroughs, and CDC CYBER systems– and has major applications in business, scientific research and education.

In education, APL can be used (and has been used) for a wide range of instructional purposes–including conventional CAI, CMI, and problem-solving. More specifically, it has been used for developing computer-assisted instruction–i.e. drill-and-practice programs, tutorial programs, simulations and games–at educational institutions across the continent such as S.U.N.Y. at Binghamton, Fairfield University, Golden State College, Orange Coast Community College, and University of California at Irvine. It has been used for computer-managed instruction–i.e. testing, diagnosis, and prescription for learning–at University of Massachusetts, University of California at Irvine, and elsewhere. And, of course, APL has been used for laboratory experimentation, problem-solving, and research by students and faculty at many locations, notably the University of Alberta (in statistics and computer science) and Swarthmore College (in mathematics and physics).

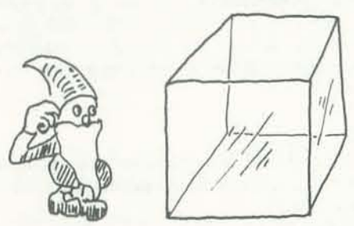
In this context, the argument for using APL in education sounds similar to that for other software systems, like BASIC. It permits individualized instruction; it speeds up learning; it fosters logical thinking and heuristic reasoning; it reduces teacher bias; it is becoming economically affordable; it relegates computational tedium appropriately to a machine, and leaves the human teacher free for the more important roles of guiding and motivating students.

But the real case for APL in education cannot be couched that conventionally. It smacks too much of revolution–a thorough rethinking of both pedagogy and curriculum. As a mathematical notation, APL seeks to reconceptualize mathematics (particularly algebra) from an algorithmic point of view; and as a vehicle for teaching, APL has great potential for clarifying students' understandings.

APL – General Characteristics. The rest of this article will be devoted to describing the specific case for APL in education but first, let me list some of the general characteristics of the language which make it suitable for teaching:

- Versatility
  APL is a multi-purpose language with applications in a diversity of areas, including business, scientific research, and education.
- Standardization
  APL is a highly standardized language. It has been an IBM program product since 1970. APL at your location is likely to correspond very closely to APL implemented anywhere else. In this sense, it is a "machine-independent" language.
- User-Orientation
  APL is designed for an interactive computing system and is usually implemented via an interpretive translator–which makes it easy for the user to enter, execute and debug a series of small programs in the process of solving a problem. Further, APL provides free-formatting of data, an immediate command execution mode, and on-line context sensitive editing.
- Mathematical Roots
  APL was originally conceived as a unifying mathematical notation, and consequently, many of its functions are familiar and all have explicit symbols. (APL has also been criticized for being "too mathematical," "too symbolic," or "cryptic.")

- Problem-Solving Power
  APL is rich—there are over 60 different primitive functions; many powerful functions are available on the keyboard, such as logarithms, the circular functions, base value, matrix inverse, and inner product; and one may create a set of user-defined functions for special purposes in a particular discipline. APL has simple rules for evaluating expressions: parentheses dominate in the normal way and every function takes the entire expression to its right as its argument (the so-called "Right-to-Left" rule)—used instead of the conventional hierarchy of functions. APL has superior array-handling capability—so many primitive functions used on scalars extend to vectors, matrices and higher-order arrays—which are treated as whole entities. And, APL is internally consistent—all primitive functions have either a monadic or dyadic syntax. APL programmers often claim that the language facilitates problem-solving efforts.
- Ease of Learning
  A subset of APL can be learned quickly (perhaps in a couple of hours) in order to get started. As one coninues to use APL, one learns more about the language—but it never seems to plateau. It takes some time before one experiences and appreciates the whole language.
- System Interface
  APL has an internal system for communicating between the langauge and the computer—using shared variables, system functions and system commands for library management.
- Teaching Effectiveness
  APL has demonstrated its effectiveness as a teaching tool—particularly in mathematics and sciences—especially for describing algorithms.



The Case for APL in Teaching. The specifics of the real case for using APL in teaching are couched both in terms of: "APL-As-A-Mathematical-Notation," and "The Glass Box Approach." Kenneth Iverson has persistently made the case for simplicity and generality in using APL to expose fundamental concepts in mathematics, engineering and the sciences [2], [3], [4]. Further, he advocates using APL as a unifying mathematical notation—where the main purpose of the language is communication—through reading, interpreting, and writing expressions formally (even on the blackboard, without a computer!).

The Glass Box Approach uses APL as a conceptual framework for learning and teaching [5]. In this approach, short and quickly comprehensible computer programs are given directly to students for their viewing. Each program embodies a concept, a procedure, or an interrelationship, and is written as *simply* and *clearly* as possible. The inner workings of such a program are designed to be visible, and hence become the basis for learning.

This approach utilizes a computer program as the initial object of study—but as a "glass box" rather than as a black box. The essential purpose of a glass box program is to *reveal* rather than to conceal; that is, by observing the structure of the program as well as its behavior, key understandings may become transparent to the student.

In contrast to most computer-assisted instruction, the Glass Box Approach allows the student significant control over his/her learning processes. Controlling the computer is intellectually (and socially) prestigious. Especially for students who have been held powerless in lock-step educational systems, use of the computer in this way opens up new worlds of learning—*active* learning, learning with *power*.

Students can learn by way of a number of different activities using a computer glass box program. Specifically, they can:

1 *Examine* the program
  —intuitively interpreting its meaning
  —logically analyzing its definition
2 *Execute* the program
  —scrutinizing the program's behavior
  —predicting outcomes for suggested inputs
  —experimenting with new inputs of their own choosing
3 *Solve* problems
  —tackling previously posed problems
  —creating new ad-hoc conjectures
4 *Modify* the program
  —changing the input or output parameters
  —revising the mathematical model
  —expanding the scope of the program
  —generalizing the program
5 *Invent* new programs
  —writing related programs
  —designing new curriculum study projects
  —generating new problems to be solved
6 *Discuss* implications
  —with their peers
  —with their teachers
  —raising issues of general concern
  —relating to other subject areas

The computer Glass Box Approach is attractive to teachers as well as students. It is pedagogically suitable for a wide range of educational levels—from elementary school children to university graduate students to on-the-job training; it accommodates different styles of teaching—including both the authoritative style and the facilitative style; it allows different modes of teaching—e.g., drill-and-practice, tutorial, simulation, gaming, laboratory experimentation, and problem solving; and it has immediate applicability—in the classroom, laboratory, or workshop. Teachers' explanations of concepts can be more lucid and effective when they are expressed in the explicit terms of A Programming Language and when they are based on a concrete, dynamic entity like a computer program. Indeed, the resulting educational activities can differ dramatically from frantic hand-waving about abstract concepts often seen in the classroom.

Examples – APL in Teaching Mathematics. The following are examples of glass box programs written in APL for teaching certain topics in mathematics:

Example 1 – Number Theory – A Program to find the DIVISORS of any integer N.

```
      ∇ DIVISORS ← DIVISOR  N
[1]   ALL ← ⍳N                    ALL integers from 1 to N
[2]   REMAINDERS ← ALL | N        The REMAINDERS of ALL the
                                  integers divided into N
[3]   WHICH ← REMAINDERS = 0
[4]   DIVISORS ← WHICH / ALL      WHICH of the REMAINDERS
                                  equals 0?
      ∇
                                  Finally, the DIVISORS
                                  are those WHICH selects
                                  from them ALL

      DIVISOR  18                 For example
1 2 3 6 9 18                      These are the DIVISORS
                                  of 18
```

We can study the inner workings of this program by examining the results of each line, thusly:

```
      N ← 18                      For N equal to 18.

      □ ← ALL ← ⍳N
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
                                  A display of ALL the
                                  integers from 1 to 18

      □ ← REMAINDERS ← ALL | N
0 0 0 2 3 0 4 2 0 8 7 6 5 4 3 2 1 0
                                  A display of the REMAINDERS
                                  of ALL divided into 18

      □ ← WHICH ← REMAINDERS = 0
1 1 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1
                                  0s and 1s indicate WHICH
                                  of the REMAINDERS equals 0

      □ ← DIVISORS ← WHICH / ALL
1 2 3 6 9 18                      Finally, the DIVISORS are
                                  those which WHICH selects
                                  (with 1s) from them ALL
```

Example 2 – Set Theory – A Program for finding the INTERSECTion of two sets A and B.

```
      ∇ COMMON ← A  INTERSECT  B
[1]   WHICH ← A ∈ B
[2]   COMMON ← WHICH / A
      ∇

      1 2 3 4 5  INTERSECT  2 3 5 7
2 3 5
```

First, find out WHICH elements of (∈) A are members of B.

Then select (/) those elements from A for the result, called COMMON.

For example,

The INTERSECTion of 1 2 3 4 5 and 2 3 5 7 is 2 3 5.

Similarly, we can examine the inner-workings of INTERSECT:

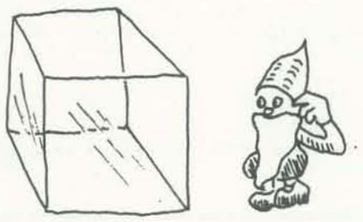
```
      A ← 1 2 3 4 5
      B ← 2 3 5 7
      □ ← WHICH ← A ∈ B
0 1 1 0 1
      □ ← COMMON ← WHICH / A
2 3 5
```

For A equal to the first five positive integers.

For B equal to the first four primes.

The elements of A which are members of B are indicated by 1s: those which are not by 0s.

Then the result is those elements WHICH are selected from A.



The specific concepts of DIVISOR and INTERSECT can be combined, giving rise to another concept in mathematics, namely GREATEST COMMON DIVISOR:

Example 3 – Number Theory – Program for finding GCD of two numbers N1 and N2.

```
      ∇ GREATEST ← N1  GCD  N2
[1]   DIVISORS1 ← DIVISOR  N1
[2]   DIVISORS2 ← DIVISOR  N2
[3]   COMMON ← DIVISORS1  INTERSECT  DIVISORS2
[4]   GREATEST ← ⌈/COMMON
      ∇

      6  GCD  15
3
```

First, find the DIVISORS of each number.

Then, find those which are COMMON (by set INTERSECTion).

Finally, take the largest of those numbers.

That number will be the GREATEST COMMON DIVISOR.

For example,

The GCD of 6 and 15 is 3.

Again, following this program step-by-step:

```
      N1 ← 6
      N2 ← 15
      □ ← DIVISORS1 ← DIVISOR  N1
1 2 3 6
      □ ← DIVISORS2 ← DIVISOR  N2
1 3 5 15
      □ ← COMMON ← DIVISORS1  INTERSECT  DIVISORS2
1 3
      □ ← GREATEST ← ⌈/COMMON
3
```

For N1 equal to 6 and N2 equal to 15.

Using the DIVISOR program to find the DIVISORS of 6.

Similarly finding the DIVISORS of 15.

Using INTERSECT to find the COMMON elements 1 and 3.

The largest of these is 3, the GREATEST COMMON DIVISOR.

Now, it is natural to develop a program for the related concept of LEAST COMMON MULTIPLE:

Example 4 – Number Theory – Program for finding LCM for two numbers N1 and N2.

```
      ∇ LEAST ← N1  LCM  N2                    First, find the MULTIPLES
[1]   MULTIPLES1 ← N2 × ⍳N1                    of each number -- (Only
[2]   MULTIPLES2 ← N1 × ⍳N2                    the multiples of one
                                               number up to the product
                                               with the other number).
                                               Then, find the numbers
[3]   COMMON ← MULTIPLES1 INTERSECT MULTIPLES2 in COMMON (using INTERSECT).
[4]   LEAST ← ⌊/COMMON                         Finally, take the smallest.
      ∇                                        This is the LEAST COMMON
                                               MULTIPLE.

      6  LCM  15                               For example,
30                                             The LCM of 6 and 15 is 30.
```

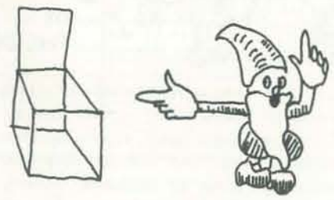
(Notice the parallelism between GREATEST COMMON DIVISOR and LEAST COMMON MULTIPLE program.)

And, examining program LCM step-by-step:

```
      N1 ← 6

      N2 ← 15                                  For N1 equal to 6 and
                                               N2 equal to 15.
      ⍳N1
1 2 3 4 5 6                                    The integers from 1 to 6.

      ⍳N2                                      The integers from 1 to 15.
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

      □ ← MULTIPLES1 ← N2 × ⍳N1                The MULTIPLES of 15 times
15 30 45 60 75 90                              the integers from 1 to 6.

      □ ← MULTIPLES2 ← N1 × ⍳N2                The MULTIPLES of 6 times
6 12 18 24 30 36 42 48 54 60 66 72 78 84 90    the integers from 1 to 15.
                                               Using INTERSECT to find
                                               the COMMON of the MULTIPLES.
      □ ← COMMON ← MULTIPLES1 INTERSECT MULTIPLES2
30 60 90

      □ ← LEAST ← ⌊/COMMON                     The smallest of these is
30                                             the LEAST COMMON MULTIPLE.
```


CONCLUSION

These examples are but a few of many which could have been chosen to illustrate the case for APL in education. (See also [6], [7], [8], [9].) The challenge to educators, now, is to identify topics suitable for embodiment in computer programs, to search out the key concepts to be taught, and to lead students to better understandings of those concepts using A Programming Language. □



References

[1] Iverson, Kenneth E. *A Programming Language.* Wiley, 1962.

[2] Iverson, Kenneth E. *APL in Exposition.* Pleasantville, NY: APL Press, 1972.

[3] Iverson, Kenneth E. *APL for Scientists and Engineers.* Pleasantville, NY: APL Press.

[4] Iverson, Kenneth E. *Introducing APL to Teachers.* Pleasantville, NY: APL Press, 1972.

[5] Peelle, Howard A. "COMPUTER GLASS BOXES—Teaching Children Concepts With A Programming Language." *Educational Technology*, vol. 14, no. 4, April 1974.

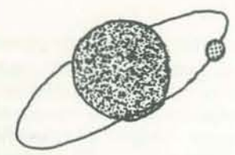
[6] Peelle, Howard A. "Euclid, Fibonacci, and Pascal—Recursed!" *International Journal of Mathematical Education in Science and Technology*, May 1975.

[7] Peelle, Howard A. "Learning Mathematics Using Recursive Computer Programs." *Journal of Computer-Based Instruction*, vol. 3, no. 3, Feb. 1977.

[8] Peelle, Howard A. "Teaching Mathematics Via A Programming Language." *The Mathematics Teacher* (to appear).

[9] Peelle, Howard A. *INSTRUCTIONAL APPLICATIONS OF COMPUTERS Using A Programming Language—A Glass Box Approach.* (To be published) Menlo Park, CA: Addison-Wesley.

[10] Peele, Howard A. *APL—An Introduction* (a self-teaching text on introductory APL). Rochelle Park, NJ: Hayden Publishing Co., 1978.



L-5 CHAPTERS

Continued from page 16.

Ashland L-5
Box 1420
Randolph-Macon College
Ashland, VA 23005

Austin L-5
P.O. Box 8213
Austin, TX 78712

Bay Area L-5
c/o David Brandt-Erichsen
(415) 526-9346
or (415) 645-5990
(Has newsletter)
814 Miramar Ave.
Berkeley, CA 94707

Boston L-5
c/o Eric Drexler
WI-518A MIT
Cambridge, MA 02139

Fresno L-5
c/o Gale Smith
36874 Cressman Road
Auberry, CA 93602

Houston L-5
c/o Larry Friesen
502 S. Austin #17
Webster, TX 77598

Jacksonville L-5
c/o Penny Stombock
Star Route 1 Box 1208
Yulee, FL 32097

Maryland Alliance For
Space Colonization
c/o Gary Barnhard
(Has newsletter)
4323 East-West Highway
Bethesda, MD 20742

Melbourne L-5
c/o Mike Dalton
650 E. Dixon Blvd. #3C
Cocoa, FL 32922

Michigan L-5
(Has newsletter)
Box 126
Michigan Union
Ann Arbor, MI 48109

Mississippi L-5
c/o Robert McWilliams
Box 5563
University, MS 38677

John Muir High School L-5
1905 Lincoln Avenue
Pasadena, CA 91103

North Carolina State L-5
c/o Robert Baldwin
Rt. 4 Box 121A
Waxhaw, NC 28173

OASIS
c/o Terry Savage
(213) 374-1381 or
(213) 536-3209
(Has newsletter)
P.O. Box 704
Santa Monica, CA 90406

Put It In Orbit Society
c/o George Timothy Bigham
Rt. 2 Box 82
Huntsville, TX 77340

Reno L-5
c/o Ray Bryan
1071 Glen Meadow Dr.
Sparks, NV 89431

Tulsa L-5
c/o Tom Huffman
3424 E. 41st
Tulsa, OK 74135

University of Houston L-5
c/o Physics Dept.
University of Houston
Houston, TX 77004

Urbana L-5
c/o Steve Vetter
168 Townsend, U.R.H.
Urbana, IL 61801

Virginia Tech L-5
c/o Kimber Smith
4016 W. Pritchard, VPI & SU
Blacksburg, VA 24061

West European L-5
c/o Roger Sansom
45 Wedgwood Dr.
Lilliput, Poole, Dorset
BH14 8ES England

Williamsburg L-5
Box 1795
Williamsburg, VA 23185

Space settlement oriented groups—cooperating with but not affiliated with L-5

Mankato Space Society
Box 58 Activities Office
Mankato State University
Mankato, MN 56001
or Dan Lundquist
(507) 345-3624

The Chicago Society for
Space Settlement
4 N. 186 Walter Drive
Addison, IL 60101

Spaceworks
c/o Robert Topple
1930 Quince Street
Denver, CO 80020

Space Futures Society
c/o Richard Bowers
(215) 739-7780
3059 Cedar Street
Philadelphia, PA 19134

Put some real world magic into your life, or into your next computer game. How? First read *Authentic Thaumaturgy* by P E I Bonewits. [And . . of course . . . read every issue of *Recreational Computing* for *magic* ideas!]

Call it magic, parapsychology or psionics, *Authentic Thaumaturgy* tells you how to use it in fantasy adventure games. As the author says, "It is *not* necessary to be able to physically see a specific target in order to use magic on it, as long as the mage has a clear idea of the nature of the specific target (usually through visualizing it or imagining it) as well as a clear idea of its general location in time and space." Watch our pages for ideas on how to use magic in most *any* computer game. Or, ask any kid.

"In recent years the war games industry has seen the phenomenal rise of a whole new family of games, the "fantasy simulation games." The first of these appear to have been outgrowths of medieval combat simulation games such as Dark Ages *or* Chainmail, *which had rules for individual as well as group combat. These were modified by science fiction and fantasy fans to allow for combat with dragons and other mythical beasts, and after a few years of experimentation entirely new games were invented in which the fantastical aspects took front-stage and in which the action could range from the clash of gigantic armies of elves, orcs, dragons and dwarves, to the adventures of small bands of brave and hardy beings.*

In the earliest of these latter games, known as "fantasy role playing games", players did not run entire battalions of soldiers against each other, but instead developed and ran individual characters *who could be Warriors, Magicians, Clerics, Thieves or members of other professions common to fantasy stories. They ran them against referees (known as "Dungeon Masters" or "DMs") who created entire fantasy worlds of deadly dungeons, harrowing wildernesses and perilous cities, populating these worlds with all manner of monsters and supernatural beings with both natural and unnatural powers. Thus, in the course of playing each game, the players and the referee would join in the continual creation of a living fantasy novel, unique on each occasion.*

To someone with professional training in the various arts, sciences and technologies that comprise the vast worlds of "the Occult", or even to someone with training in that rigid and apologetic science known as "parapsychology", it quickly becomes apparent that the rules for "magic" used in most of the games are simply not very successful at accurately simulating the behavior of magical phenomena in the real world. This is not too surprising, since with the exception of one or two of the game designers for Chivalry & Sorcery *and the author of* The Arduin Grimoire, *none of the rule makers seem to have much of a serious background in any known system of real world magic (Western, Eastern or Tribal), nor to know very much about the results of recent parapsychological research.*

Instead, the sources for most of the magic systems currently being used in the games seem to be fictional works *of Science Fiction, Fantasy, Sword and Sorcery, and Horror (novels, short stories, television and motion pictures). Unfortunately for the sake of simulations authenticity, the authors and scriptwriters of these tales, no matter how knowledgeable and talented they may be in other areas of endeavor, usually know very little and care even less about the ways in which magical phenomena and magicians really behave. This is, of course, because they have all accepted the official Scientific dogma that magic is "impossible" and that psychic phenomena are caused by some mysterious and all powerful cosmic force known as "coincidence."*

. . . probably the most popular gimmick for scientistic writers is to use the magical Law of Words of Power (see Chapter 3) and to change the horribly unscientific sounding word "magic" to something else, such as "psychic," or "parapsychological," or even better, "psionic." That last one sounds really scientific! This allows them to use the same old magical materials in respectable new arrangements, without ever having to admit what they have done. Many of the games thus set up systems of "psionics" as if the phenomena covered under that term were somehow different from those of "magic." This can be a workable game system (especially in such games as Metamorphosis Alpha, *where they are called "mental mutations" and which have no magicians as such) but I tend to feel that it is not as accurate a simulation as it could be. Another example of a playable psionic system can be found in the* Arduin Grimoire *and its author and I have had many enjoyable hours arguing over the merits of our respective approaches."*

The use of magic in games requires the use of magical devices (such as dice or computers) and tables of possible outcomes. The next page shows a table from the book.

TABLE 2: PSYCHIC ABILITIES TABLE (II) THE "ANTI-PSI" POWERS		"RAW TALENT" USE: Probability of success for single talent use.					"FULL SPELL" USE: Contribution to Probability of Spell Success (PoSS).						
	%ile	R-1	R-2	R-3	R-4	R-5	R-1	R-2	R-3	R-4	R-5	Basic Cost in MPs	See Note of this letter
Catapsi (static)	91-00	.30	.40	.50	.60	.70	.10	.13	.16	.19	.22	03	L/P
Splodging (yelling)	86-90	.25	.35	.45	.55	.65	.08	.11	.14	.17	.20	03	L/P
Apopsi (avoiding)	81-85	.15	.25	.35	.45	.55	.04	.07	.10	.13	.16	05	L/P
Deflection (bouncing)	76-80	.30	.40	.50	.60	.70	.10	.13	.16	.19	.22	05	M
Reddopsi (returning)	71-75	.25	.35	.45	.55	.65	.08	.11	.14	.17	.20	07	M
Negapsi (inverting)	66-70	.30	.40	.50	.60	.70	.10	.13	.16	.19	.22	04	M/P
Filtering	61-65	.25	.35	.45	.55	.65	.08	.11	.14	.17	.20	05	N/P
Retuning	56-60	.20	.30	.40	.50	.60	.06	.09	.12	.15	.18	05	N/P
Damping	46-55	.35	.45	.55	.65	.75	.12	.15	.18	.21	.24	05	O/P
Amplifying	31-35	.35	.45	.55	.65	.75	.12	.15	.18	.21	.24	05	O
FLAIR!	01-30	+10	+10	+10	+10	+10	+10	+10	+10	+10	+10	00	J

The inside front cover of *Authentic Thaumaturgy* has a bibliography of games and game supplements. A library of fantasy game adventuring!

Yes, we know it needs updating. Please send update stuff to The Dragon!

BIBLIOGRAPHY OF GAMES & GAME SUPPLEMENTS

Dragon Pass (White Bear & Red Moon)
Greg Stafford
Nomad Gods, Greg Stafford
Elric, Greg Stafford
Perilous Encounters, S.L. Lortz
Runequest, Steve Perrin & Friends
All the World's Monsters (Vols 1&2)
Perrin/Pimper
All from The Chaosium
PO Box 6302
Albany, CA 94706

The Emerald Tablet
Miceli/Loback/Facciolo
From Creative Wargames Workshop
330 East 6th St, No 1-E
New York, NY 10003

Chivalry & Sorcery, Simbalist/Backhaus
Lords & Wizards, Adam L Gruen
War of the Ring, Thomas Drake
All from Fantasy Games Unlimited
PO Box 182
Roslyn, NY 11576

Tunnels & Trolls, Ken St Andre
T & T Supplement, Ken St Andre
From Flying Buffalo
PO Box 1467
Scottsdale, AZ 85252

The Book of Sorcery, Konstant/Bress
The Book of Demons, Phil Edgren
Ringbearer, Bress/Konstant
All from Little Soldier Games
1776 East Jefferson St
Rockville, MD 20852

Wizard, Steve Jackson
Monsters! Monsters!, Ken St Andre
From Metagaming Concepts
PO Box 15346
Austin, TX 78761

Arduin Grimoire (Vols 1 & 2), Dave Hargrave
From R Powell
5820 John Ave
Long Beach, CA 90806

Sorcerer, Simonsen/Walczyk
Gondor, Mosca/Berg/Beavers
Sauron, Mosca/Beavers
War of the Ring, Simonsen/Barasch/Berg
(Last 3 published jointly as *Middle Earth*)
From Simulations Publications
44 East 23rd St
New York, NY 10010

Dungeons & Dragons, Arneson/Gygax
Grehawk, Gygax/Kuntz
Blackmoor, Dave Arneson
Eldritch Wizardry, Gygax/Blume
Gods, Demigods & Heroes, Kuntz/Ward
Metamorphosis Alpha, James Ward
Empire of the Petal Throne, M A Barker
War of Wizards, M A Barker
All From Tactical Studies Rules
542 Sage Street
Lake Geneva, WS 53147

One more thing—a word about the author. Philip Emmons Isaac Bonewits is the world's only academically accredited occultist, with a B.A. from the University of California in Magic. He is the author of the classic "Real Magic" and of numerous articles on ceremonial magic, witchcraft, divination and related topics.

As a Pagan "theologian," he is considered one of the top living experts on Polytheistic religions, ancient and modern, with a special focus on Druidism and Witchcraft. He lectures widely, teaches private students, and counsels clients from a variety of backgrounds.

P. E. I. Bonewits

Authentic Thaumaturgy. $7 from The Chaosium, P.O. Box 6302, Albany, CA 94706. □

CHIVALRY & SORCERY

We are beginning to look at *Chivalry & Sorcery* by Ed Simbalist and Wilf Backhaus. This appears to be a very complete encyclopedia/rulebook for creating fantasy adventure games. More about this in a future *Dragonsmoke*. In the meantime, if you want to try it: $10 from Fantasy Games Unlimited, P.O. Box 182, Roslyn, NY 11576, or The Chaosium.

REVERSE

BY RAMON M ZAMORA

In the March 1977 issue (Vol. 5, No. 5, p. 34) of this magazine, the Dragon (Bob Albrecht) put out a call for an animated version of the classic game REVERSE. REVERSE begins with a random scrambled list of from two to nine numbers. The game player attempts to unscramble or order the list by reversing two or more numbers in the list. The reversing operation is applied to the numbers beginning on the left end of the list.

The program shown here is one of two this issue (Look for the game FROGS elsewhere!) that utilizes the TRS-80's graphic capabilities to 'animate' the game. When the player specifies how many numbers to reverse, the rotation of that subset of the list is displayed on the screen. The numbers pivot about themselves; they always remain visible to the user. (See the photographs of the operation.)

REVERSE and FROGS are building block versions of a series of recreational routines that will be part of a forthcoming book of TRS-80 games and programs. Watch for more of these TRS-80 programs in future issues of *RC*. Happy ЯEVEЯSEing!!

1

3

2

4

REVERSE

```
5 REM **REVERSE 16K LEVEL II BASIC TRS-80 R. ZAMORA 9/13/78**
10 DIM D(8,6),A(9)
20 DATA 0,64,66,2,-62,-64,0
30 DATA 0,64,68,4,-60,-64,0
40 DATA 0,128,134,6,-122,-128,0
50 DATA 0,128,136,8,-120,-128,0
60 DATA 0,192,202,10,-182,-192,0
70 DATA 0,192,204,12,-180,-192,0
80 DATA 0,256,270,14,-242,-256,0
90 DATA 0,256,272,16,-240,-256,0
105 CLEAR 100
110 FØR J=1 TØ 8:FØR I= 0 TØ 6: READ D(J,I):NEXT I:NEXT J
120 CLS:PRINT@10,"THE GAME ØF REVERSE";
122 PRINT@64,"DØ YØU NEED INSTRUCTIØNS?";
124 A$=INKEY$:IF A$="" THEN 124
126 IF A$="Y" THEN 3000
128 CLS:PRINT@10,"------REVERSE------";
130 PRINT@74,"HØW MANY NUMBERS DØ YØU WANT IN THIS GAME?";
131 A$=INKEY$:IF A$="" THEN 131 ELSE N=VAL(A$)
132 PRINT@118,N;:IF N=0 ØR N=1 THEN 232
133 C=-1:X7=468
134 FØR I=1 TØ N
135 S=RND(N)
136 IF A(S)=0 THEN A(S)=I ELSE 135
137 NEXT I
140 FØR I=0 TØ 2*(N-1) STEP 2:PRINT@X7+I,A(I/2+1);:NEXT
142 PRINT@33,:GØTØ210
150 PRINT@74,"NUMBER TØ REVERSE?";
152 A$=INKEY$:IF A$="" THEN 152 ELSE L1=VAL(A$)
153 PRINT@94,L1;
154 IF L1=0 ØR L1=1 GØTØ 232
155 IF L1>N GØTØ 150
160 FØR I=1 TØ 3:X=X7:K1=1:K2=L1:L=L1
170 FØR J=1 TØ INT(L/2): X1=D(L-1,I)+X
180 X2=D(L-1,I-1)+X:X3=D(L-1,I+3)+X:X4=D(L-1,I+2)+X
190 PRINT@X2,"△△";:PRINT@X1,A(K1);:PRINT@X4,"△△";
192 PRINT@X3,A(K2);
200 L=L-2:K1=K1+1:K2=K2-1:X=X+2:NEXT J:GØSUB 2000:NEXT I
205 K1=1:K2=L1:FØR I=1 TØ INT(L1/2)
206 S+A(K1):A(K1)=A(K2):A(K2)=S:K1=K1+1:K2=K2-1
207 NEXT I
210 L1=0
220 FØR I=1 TØ N:IF A(I)<>I THEN L1=1 ELSE NEXT I
222 C=C+1
225 IF L1=1 GØTØ 150
230 PRINT@72,"THEY ARE IN ØRDER--IT TØØK THIS MANY REVERSES "+STR$(C)
232 PRINT@138,"NEW GAME?";
233 A$=INKEY$:IF A$="" THEN 233
235 FØR I=1 TØ 9: A(I)=0:NEXT
236 PRINT@150,A$;:PRINT@119,"△";
240 IF A$="Y" THEN 128
242 PRINT@33,:PRINT@74,
245 PRINT@192,"THANKS FØR PLAYING--LET'S PLAY AGAIN SØMETIME.";
246 END
2000 FØR M=1 TØ 500:NEXT:RETURN
3000 GØSUB 3500
3020 PRINT"THIS IS THE GAME OF REVERSE. I WILL GIVE YØU A"
3030 PRINT"SCRAMBLED LIST ØF NUMBERS. YØU UNSCRAMBLE THE LIST."
3040 PRINT"YØU PUT IT IN NUMERICAL ØRDER. FØR EXAMPLE...."
3050 PRINT:PRINT"SCRAMBLED LIST:      2 5 1 4 7 6 3"
3060 PRINT:PRINT"PUT IT IN THIS ØRDER: 1 2 3 4 5 6 7"
3070 PRINT:PRINT"WHEN YØU ARE READY, HIT ANY KEY, AND I WILL"
3080 PRINT"TELL YØU MØRE ABØUT THE GAME ØF REVERSE.";
3090 A$=INKEY$:IF A$="" GØTØ 3090
3100 GØSUB 3500
3110 PRINT"YØU MAY REVERSE TWØ ØR MØRE NUMBERS IN THE LIST,"
3120 PRINT"STARTING WITH THE LEFT END ØF THE LIST."
3130 PRINT"FØR EXAMPLE:":PRINT
3140 PRINT"ØRIGINAL LIST    2 5 1 4 7 6 3"
3150 PRINT"WE REVERSE 3     1 5 2 4 7 6 3":PRINT
3160 PRINT"YØU MAY CHØØSE THE SIZE ØF THE LIST, FRØM 2 TØ 9"
3170 PRINT"NUMBERS. HITTING ANY ØTHER KEYS WHEN YØU ARE BEING"
3180 PRINT"ASKED FØR THE LIST SIZE ØR THE NUMBER ØF ELEMENTS"
3190 PRINT"TØ REVERSE WILL GIVE YØU THE CHANCE TØ START A"
3200 PRINT"NEW GAME. IF YØU ARE READY TØ BEGIN, HIT ANY KEY...";
3220 A$=INKEY$:IF A$="" THEN 3220 ELSE 128
3500 CLS:PRINT"INSTRUCTIØNS FØR THE GAME ØF REVERSE"
3510 PRINT"-----------------------------------":PRINT
3520 RETURN
```

Data elements for controlling where printing occurs during reverse operations.

Check to see if user needs instructions.

Generate and print scrambled list.

Perform reversing operations.

Check for completed game and replays.

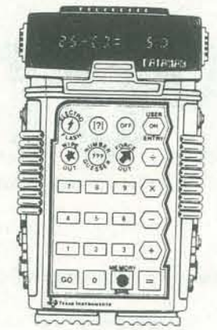
Delay routine.
Instructions for the game.

The symbol △ indicates a space.

□

DATAMAN

BY BERNIE DE KOVEN



DATAMAN (Texas Instruments, Lubbock, TX 79408. Prices discounted as low as $17.95.)

Dataman is an electronic device allowing the user to play any one of six games using elementary mathematics.

The game that has held everybody's attention at my house for the longest time is called "Number Guesser." You begin the game by pressing a button that has this symbol "???" on it. The large LED display shows you the following: "9 ( ) 100". Then, you guess any number between those two numbers. Suppose you enter 65.

After you enter your number, the display goes through a minor pyrotechnic phase of dashes and a circle, and then shows you something like this: "9 ( ) 65". This means that the number you're looking for is between 9 and 65. Then you enter your next guess. Say you guess 25. Then the display would show you something like "25 ( ) 65". This means that your number is between 25 and 65.

You continue in this manner until you hit the number the calculator has generated for you, and the display then shows you how many guesses it took you to guess the number.

The other games on the device are as follows: "Electro-Flash," which allows the user to select any function (multiplication, division, addition, or subtraction), and then provides the user with a sequence of problems. When the user makes an error, the display goes through some more pyrotechnics and then offers the problem again. When the user misses the second time, the display shows the correct solution and moves on to the next problem in the sequence.

When the sequence is over, the display shows the user how many problems were in the sequence, how many of them were answered correctly on the first trial, and how long it took the user to solve the entire sequence. We've been able to solve the "? X 0=" sequence in 2 seconds.

"Wipe Out" offers a random series of addition problems. In this game, as soon as one user has correctly answered the problem posed by the program, he passes the device to the next player. This game is really a version of hot potato, because the object is to make sure that somebody else has the device when it stops giving problems and goes into a long and rather sinister display sequence. This game has been the least popular, though we all enjoy watching the display do its thing.

Then there's a game which is called "Force Out." This is a game of nim in which the object is to subtract any number from 1 to 9 so that the last player using the device has to make the solution equal to zero. Until we figured out the pattern, this was our second most favorite.

Then there's a game that's merely indicated by a "(?)". Actually, it's not a game, but a different kind of problem you can pose yourself in which the solution lies in either the first, second or third element of the equation [3 + ( ) = 3, or ( ) + 0=3, or 3+0=( )]. The program gives the user a random series of problems, and when the series is complete, it again tells the user how many problems were offered, how many were correctly answered and how long it took to complete the series.

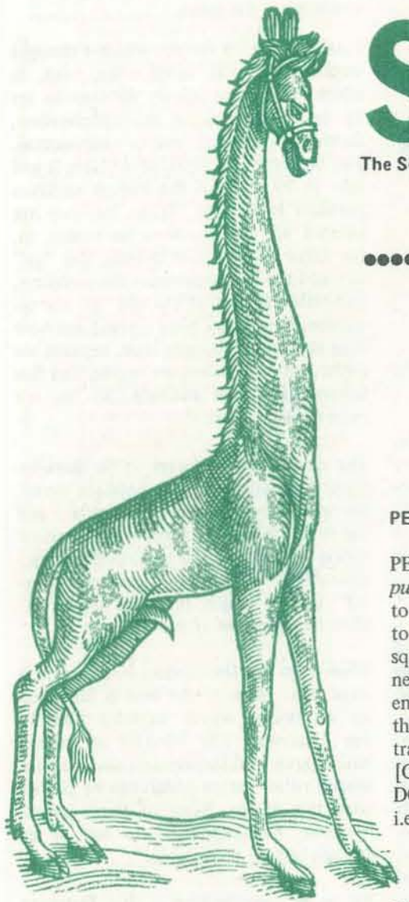
Finally there's a feature which I thought would prove the most often used, in which the device allows the user to set up any ten problems, in multiplication, division, addition, and/or subtraction. The highest multiplication problem it will take is 99 x 9, and the highest addition problem is 99 + 99. When the user has entered all the problems he wishes to, the other player then presses the "go" key and the device presents the problems, and tells the user, at the end of the sequence, how many were correct and how long it took. I thought that, because we could test each other, we would find this feature the most exciting. So far, it's been the least used.

The device itself appears to be thoughtfully designed. The game keys are round, the number keys are rectangular, and the function keys hexagonal, so discrimination is easy and the keyboard is attractive. Also, there is an "automatic turn-off" feature which turns the device off after five munutes of non-use.

When you get the device, it comes in a large box. Most of the box is taken up by a booklet which explains how to use Dataman. The booklet is written with humor and fantasy and also includes several other games which can be played with the device. None of these games, however, have appealed to any of us enough to play them again.

All in all, our finding is that Dataman is exciting as a prototype. Better than Little Professor, it offers children a genuinely entertaining opportunity to explore recreational arithmetic. It is most effective, or at least most often used, for recreation rather than for an alternative to educational drill and practice. Given these hints, I look forward to a new generation of such devices which allow people, through play, to develop mathematical skills. ☐

Part of the new generation is already here with the Speak and Spell by Texas Instruments reviewed on page 56 of this issue. – CB



SPOT

The Society of PET Owners and Trainers

PET POINTERS

● ● ●

PET LISTING CONVENTIONS

PET Program listings in *People's Computers* employ the following conventions to represent characters that are difficult to print on a standard printer: Whenever square brackets appear in the listing, neither the brackets nor the text they enclose should be typed literally. Instead, the text between the brackets should be translated to keystrokes. For example, [CLR] means type the CLR key, [3 DOWN] means [DOWN, DOWN, DOWN] i.e., press the first CRSR key three times.

● ● ●

TAPE TIP

I recently bought tapes and was able to load only one side of each tape. I couldn't load the other side and was ready to mail them back until I tried this: I placed a small piece of cloth (about 3/8" by 2") under the cassette and forced the lid to close. Then they both loaded O.K. Maybe I aligned the tape and tape head by doing this? R.L. McArthy, San Clemente, CA.

Who knows... but it certainly sounds worth passing on the information. Thanks for the tip.—Ed.

RANDOM REMARKS

Concerning RND(X): Memory locations 218-222 store previous random number, in usual PET notation:

R = ((((PEEK(222)/256 + PEEK(221))/256 + PEEK(220))/256 + PEEK(219))/256 + .5)*2↑(PEEK(218) – 128)

To "randomize," try a statement like "X=RND(–TI)." Don't use the resulting X, but call RND(1) thereafter. (RND(negative #) fills memory locations 218-222 with a scrambled-bytes version of the argument.)

Mark Zimmermann
Pasadena, CA

VANCOUVER PETS

The Vancouver PET Users Group recently held their second meeting. The success of the PET has caught us all by surprise. The attendance at our second meeting was over double the first, with some 40 owners and 15 PETs. The Commodore dealers indicate that there are many more PETters who are not aware of our group. Interested persons should phone Rick Leon at: (home: (604) 734-2060); (work: (604) 324-0505). They can also write to:

Vancouver PET Users Group
Box 35353 Station E
Vancouver, BC
Canada

The club format includes a short presentation by a PET owner on programming on PET hardware, news from Commodore and other sources. Followed by PET Patter and program swapping.

PROGRAM TAPE SEARCH

I use an external cassette recorder with my PET, and I have 10 seconds between programs on a tape (recorded at zero volume). This makes it possible to locate any program using a SHARP RP 1155 Deck with Automatic Program Search System (APSS). On fast forward or rewind it will stop at the first blank in the tape. Great for verifying and locating programs.

Jack Clark
Oxon Hill, MD

Commodore's PET is a factory assembled personal computer based on a 6502 microprocessor. The $795 system includes a keyboard, cassette tape unit, built-in TV screen, some graphics, upper and lower case, and extended 8K BASIC, and 8K of user memory. Each bimonthly issue of People's Computers *since the September-October 1977 issue has included an article on the PET.*

PET PAWS

KEYBOARD HOOKUP

In a recent SPOT column we learned that our PETs can be trained to speak (via hard copy printout) using a Diablo Printer and a PET ADA interface (see the article on "PETting a Diablo," Jul-Aug of *PC*, p.47). Well, now it is also possible to produce, if you will, a new set of "paws" for our PETs. ("Paws" in the sense of a new kind of keyboard for the PET . . . or, another "set of fingers," a new "hand," so to speak). I am alluding, of course, to the possibility of wiring up another, more standard kind of keyboard to the PET-one which will work in parallel with the existing PET keyboard.

As Ludwig Braun pointed out in a recent review of the PET which appeared in *Creative Computing* (see page 24, July/August issue), the PET does lose some points if you consider the more negative features of its keyboard design. Obviously this keyboard is best suited for young users with small hands. It is apparent that many adult users would prefer a larger, more standard size keyboard to work with, especially when they want to enter a lot of text, graphics, or data. It is particularly true that when the PET is used for business applications, often requiring large amounts of data processing (number crunching), the present PET keyboard is not so well suited for rapid data entry in volume.

So, we must conclude that for some purposes the PET needs a new set of "paws." And for those SPOT readers who may be interested in checking out the following reprint of an article on this topic, here is a solution to this problem:

> "So you don't like the KYBD that comes with your PET?!!! Well it has proven to be a snap to add on your own KYBD . . . and I do mean add on, not replace!
>
> Since encoding the KYBD information for the PET is done by the BASIC under the 60-cycle interrupt, there is only one of ten lines strobed out to the existing KYBD (10 by 8) matrix at a time. The BASIC then looks to see if there is a return of this strobe on any of the eight returning lines from the matrix, caused by a depressed key. If there is no return of the strobe, the BASIC chooses another of the ten lines to strobe until it has gone through the full ten, when, if there have been no keys pressed, it will put a 255 (or FF hex, or the Pi character) in location 515 (or 0203 hex). If a key had been depressed, a simple contact was made from one of the ten strobe lines to one of the 8 return lines. The BASIC would then have inserted a value in location 515 which was equivalent to the sum of the decimal numbers in the corresponding row and column (as indicated in the first diagram to follow).
>
> The KYBD plug/receptable on the mother board (inside the PET) has numbered strobe lines and lettered return lines. If you arrange your add-on keyboard strobe line connections according to the pattern in the second diagram to follow, so that they make contact with the return lines (indicated by the lettered groupings over the diagram in figure 1), then you can connect your eighteen lines to the existing KYBD plug (in parallel) and still retain the graphics keypad on the PET!!!!
>
> As an example: a #4 strobe line connected to, and returned by an "E" line will print the character "P"."

This excerpt was from an article originally printed in the "PET User's Group Newsletter" (volume 0, no. 1), published by Pete Rowe, for the PET Computer Project, Lawrence Hall of Science, Berkeley, and was written by Neil Bussey and Richard Tobey.

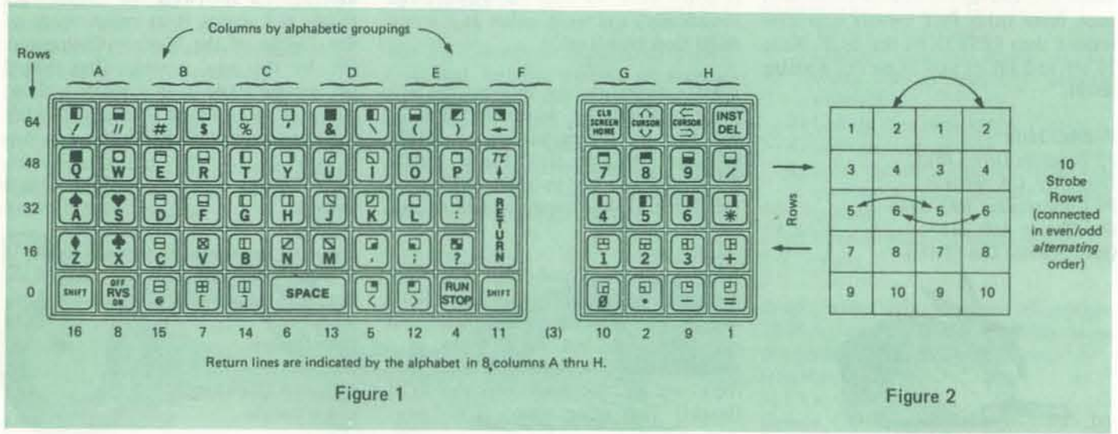


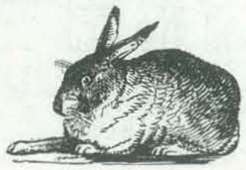
Figure 1 Figure 2

In my experience, it was a little bit more than "a snap" to add on my own keyboard to the PET! However, I can now say that all my effort was really worth it, because now that I have another keyboard to use with my PET it is really easy to sit down in a comfortable chair and enter in long passages of text or a great many data statements! With a larger size, typewriter-like keyboard, it would be quite easy to type in text at 60+ wpm speed (with the aid of a mini-word processing software system). If you are able to wire in a 3-foot long ribbon cable from the keyboard to the PET, then you could conveniently place the PET on a table top and still sit nearby in a comfortable chair with the new keyboard resting on your lap. Now isn't that a neat way to make home computing fun and comfy?!!

For those of you who might be interested in tackling this project of adding on another keyboard to your PET, you can write to me at the address given below for more details. It will not be a very expensive project, as the keyboard alone only costs about $30.00. However, you must be willing to spend several hours wiring up the keyboard matrix connections, which I found to be not an especially easy task. If you are interested in contacting me, then I will be able to give you further hints that may save you time and trouble.

I'll conclude this letter now by leaving you with this thought to ponder: wouldn't it be neat if you could shift-LOCK on the PET, especially when you want to draw on the screen using the graphic keyboard characters? Well, with an add-on keyboard you can do just that. I know of at least three other PET owners who have trained their PETs to do this trick. Write to me and I'll let you in on this training secret.

Randall Julin
15 Poncetta Drive #322
Daly City, CA 94015
PET housecalls: 992-6946
Day Phone: 469-1157
(Info Science Lab SFSU)



PET PEEVES

CONLEY VS. COMMODORE

Thanks for the copy of *People's Computers*. (I guess my subscription has expired—just can't figure out the Greek on your mailing labels.) Thought I'd let you know what happened on that Commodore/PET problem we were having.

Sometime about the middle of June, I got a letter from a Ms. Lennie A. Cooper, Manager, Customer Services at Commodore. It basically (BASICally?) said that the PET has a defective ROM and the tape transport heads may be out of alignment and that I should send it off to Palo Alto to get it fixed. I wrote back and said that I didn't want to send our PET off for an unknown length of time to who knows what kind of treatement at the gentle hands of UPS/Commodore. I said to forget the tape deck problem (I don't think she read my second letter too closely) and we'd rather *buy* a new ROM than send it off.

I got a letter in July showing the new ROM part number which cost $10.65, and that the defect was in the BASIC software. By this time I was totally PO'ed and had my assistant call her on the phone. (I didn't want to talk to anyone by this time).

After much back-and-forth with her, she finally agreed to send us a new ROM out of the kindness of her heart and not only that, but she'd throw in a manual/instruction book. After a few days, we got the book and the ROM (book?! That thing wasn't much more than the 8-page throwaway that came with the computer. It *was* on shiny paper, though).

The ROM wouldn't fit!!! $@#%¢# %$*&!! It seems that the older machines (I don't think she read my first letter either!) use a 24 DIP ROM and she sent me a 28 DIP ROM that the newer ones use.

Another phone call. "Send it back and we'll send the right one."

Another week . . . I got the other ROM with the attached note. This one went in OK. Did that fix it right up, you ask? All I can say is that it works *somewhat* better. I only lose the cursor (or curser, as I call it) about once a day now, instead of 5 times a day.

At this point, I'm giving up on Commodore—I'll live with the disappearing curser and accept it as symptomatic of the decline of the Western Civilization. Oh, by the way, a week after that, I got another ROM in an envelope without any note at all (good Fairies?). It didn't work any better than ROM #3.

How those guys run their service dept is strange!

I wonder how their printer will work? If it won't work correctly right out of the box, here we go again. The hell with 'em.

Dave Conley
Minister
Universal Life Church
of the Pacific

PET PRODUCTS

S-100 BUS CONVERTER BY EXCEL

Excel Co. has designed two products to extend the usefulness of your PET. One is the S-100 Bus Converter (S100 BC), an inexpensive device to allow your PET to interface with the many S-100 products on the market. Some of the boards that can be used are Godbout's 4K, 8K static memory; Solid State Music's 4K, 8K static memory; and IMSAI's Multiple I/O boards. The S100 BC is $85, not including cables and connectors.

The other product is an ASCII Keyboard Interface. It allows any ASCII encoded keyboard to talk to the PET through the same connector as the original PET keyboard. No modifications are necessary; and since the PET keyboard is not disabled, it can be used for all its original functions including graphics. The Interface is $80. For more information contact:

EXCEL Company
P.O. Box 1147
El Cerrito, CA 94530

CGRS MICROTECH S100 ADAPTOR

CGRS Microtech, Inc. announces the PET /S100 adaptor. CGRS, the 6502/S100 experts, have developed an adaptor card that will convert the "memory expansion" connector from the PET computer to the S100 bus. With this card the owners of the PET can expand memory, add I/O devices (printers, floppy disc, etc.) and enjoy the advantages of the numerous S100 products.

The PET/S100 is a single board that plugs into a card slot of any S100 motherboard and connects to the PET memory expansion connector via a flat ribbon cable. the PET/S100 adaptor can be used to adapt the KIM, the MOTOROLA EVII, and other 6502 or 6800 computers to the S100 bus using the appropriate connector cable. This board is unique in that it generates all of the required S100 control signals such as Psync, I/O address, wait states and – much more. This is the same design that has been proven by CGRS over the past several years with their S100 6502 MPU card.

The PET/S100 comes with instructions and a complete S100 compatibility list. It is available fully assembled and tested for $195.95. Send check or money order to: CGRS Microtech, Inc., PO Box 368, Southampton, PA 18966.

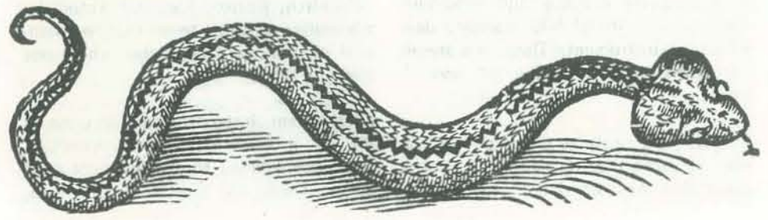
PET CRT SPACING CHART

PET CRT spacing and layout charts are available in 8½ by 11 inch pads. With green printing on white background, one side shows 25 numbered rows by 40 columns for planning your PET screen output. The other side is printed with spaces to record BASIC variable descriptions, subroutine descriptions and also shows the PET keyboard with ASCII codes. approximately 50 sheets per pad at $2.00 each, or 5 pads for $7.50 post paid.

Alcorn Data Services
P.O. Box 1535
Corinth, Miss. 38834

PET GAZETTE

The *Pet Gazette* is an informal publication that includes reveiws of software and equipment, tips on operating and using your PET, and a software exchange. The *Pet Cassette Exchange* has over 150 user supplied programs available for only $1.00 each. The *Gazette* also features subscriber donated articles.



The pocket sized issue ran 40 pages in July '78, and operates on a small income from advertisers. Copies are free to individuals, but they would love any donations you can send. It's worth it for the helpful bits of information that isn't readily available elsewhere. Contact:

Len Lindsay, Editor
Pet Gazette
1929 Northport Dr., Room 6
Madison, WI 53704

READ, WRITE PET MEMORY©

The PET computer's BASIC language instruction set provides four commands/statements which allow the programmer to store and execute machine-language programs. There are: POKE, PEEK, SYS, and USR. The POKE statement, which stores data directly in memory, is a little clumsy to use because it takes a separate statement for each memory position; and the address of the memory position, and the data to be stored have to be in decimal form. Addresses and memory contents are usually expressed in hexadecimal form.

The *Read, Write PET Memory* program by Don Ketchum, greatly simplifies the entire procedure. With this program you store data in memory by simply keying in the starting address (in hexadecimal) and then just key in the hex digits in the sequence in which you want them stored. Nothing could be simpler! Here is a list of the functions which can be very easily carried out with this program. (All data and addresses will be in hexadecimal.)

1) Write into memory.
2) Display memory (both as hex digits and as assembly language mnemonics).
3) Change data in memory.
4) Save memory on tape.
5) Load memory directly from tape starting at any selected memory address.
6) Execute machine-language programs.

Read, Write PET Memory (Copyright 1978 by Don Ketchum) is available on tape cassette, $7.95 postpaid from:
Don Ketchum
313 Van Ness Ave.
Upland, CA 91786 □

REVIEWS

TELEPHONE ACCESSORIES YOU CAN BUILD
By Jules H Gilder
Hayden Book Company, Inc, 1976
84 pp. $4.95

The telephone is to electronics as military thinking is to thinking. It is intended for only one use—to carry the human voice (or a modest reproduction of it) from one place to another and work when everything else fails. The secrecy which surrounds this device is legend. If you doubt the legend, try this experiment. Call your local telephone business office, and order a schematic for your telephone.

On second thought, to actually do the experiment may be a bad idea. If you are the sort of person who might read, enjoy and act upon *Telephone Accessories You Can Build*, then you may not wish to suggest to the phone company that you are about to peek and poke into their instrument. Ma Bell's reason for discouraging you is simple and mechanistically logical. If you and millions of other people are allowed to unscrew telephones, and place Foreign Attachments on the line, the reliability of the entire system will be reduced. And this would be a violation of the prime directive.

Anything that touches the telephone installation in a clinging way is a Foreign Attachment. I have heard stories dating many years back, of the persecutions of small town businessmen who eked out a living by printing ads on plastic telephone *directory* covers. Is it any surprise that Michael Scheff opened his introduction to *The Pushbutton Telephone Songbook* with these words, "About the only bond people the world over seem to have these days is (1) a desire for peace and (2) a dislike of The Telephone Company." The *Songbook* was Scheff's effort to help improve the telco image.

The Bell System has long had the very best technical reputation and a consistently poor record of human relations. Three factors are known to have improved customer satisfaction with Bell services: (a) federal court litigations and FCC rulings; (b) actions before the state public utility commissions, and (c) disclosures of financial and technical information to the general public.

Technical disclosure is necessary because Bell made a practice of telling technical half-truths biased toward gaining favorable regulation. The most outrageous example is that of the infamous Voice Connecting Arrangements. Gilder briefly discusses them in chapter two. The "need" for VCAs was hastily discovered following the 1968 Carterfone Decision, but this need was actually financial, not technical. The Carterfone, by the way, is a two-way radio phone patch.

Telephone Accessories is a somewhat useful interim book on do-it-yourself telephone technology. I hope that later and better illlustrated books on this subject will appear. The book has absolutely no discussion of Blue or Black Boxes, or phone phreaking of any kind. Unfortunately for computerists, it also has no discussion of modems, cheap or otherwise.

The only proprietary information revealed is in chapter one, which is a little less than four pages long. There is a cutaway drawing and schematic diagram of a Model 500, standard desk telephone instrument. There is a simple but adequate explanation of how it works.

Chapter two provides a concise description of current tariffs and your options concerning them which are: obey and use loopholes; obey and pay Bell; or ignore them. Telco propaganda generally encourages you to believe that tariffs are laws. In general, they are not.

Tariffs are rules of system use including fees for service, made up by regional telephone companies (plus AT&T for long distance service), and submitted to utility commissions and the FCC for approval. In spite of all the discussions of "illegal" extensions you have probably heard over the years, violation of a simple tariff is not a crime. The worst that could happen is loss of phone serivce until you agree to abide by the rules.

Having said that most tariffs are toothless, please note that there is also a short list of laws which make some irregular acts, illegal as well. It is a crime to: make obscene calls; fail to yield a party line in an emergency; commit wire fraud (i.e., use a Blue Box to avoid long distance charges, or use a computer to steal money/secrets/data via the phone lines); commit a few other acts such as theft of telco equipment.

Chapter three contains some simple and clever methods for quickly making printed circuit boards using among other tools, a leather punch.

A major defect in all the projects is a lack of mechanical drawings or pictures. I find it very irritating to read a long paragraph of construction steps with no drawings to reference. It makes an otherwise easy project unnecessarily difficult.

The best feature is the use of building block circuits, to which the builder is introduced one at a time in the early projects. The later, more complicated accessories are often collections of the earlier units. This approach does get out of hand. From a casual inspection of the Teleswitch, Remote Ear, and Autodialer schematics, it seems to me that two identical circuits are constructed where one could do the job.

I was disappointed to find that some of the 15 projects were not worthwhile, so I have divided them into three categories: Useful—8; Useful, but . . . —4, and Silly—3. Lend me your Remote Ear while I make specific comments starting with the Silly group. The number of parts is approximate.

Note to Software Vendors: We are now reviewing recreational programs. If you have software you would like reviewed please send us cassettes of your games and simulations. Please enclose information on the language used, the name of the machine the software runs on, etc.—RZ

The Silly Group

- *Dialer*—10 parts. Claims to be the equivalent of a Bell-supplied card dialer, but I doubt it. It is almost entirely a mechanical device, to be constructed without the aid of mechanical drawings. If you did manage to get it built, and get all the microswitches properly aligned, I question whether you would bother using it. Personally, I would prefer one of those little dialing balls that fits on the end of a pencil.
- *Telephone Burglar Alarm*—60 parts. Just an ordinary burglar alarm which generates a tone in a speaker. You pick up your business phone and call your home. You lay the phone down next to the speaker and leave. At your home a sound switch listens for the tone indicating a burglar has entered. Gilder admits this scheme would tie up your home phone all the time it was in operation.
- *Phone Lock*—5 parts. This electrical combination phone lock is just not very secure. Whenever the phone is in use the combination is exposed for all to see. Anyone with a screwdriver and a little time can easily unwire it, use the phone, and replace it without a trace. Gilder objects to the mechanical phone dial lock, because it and its keys are easily lost. A valid objection, but the electrical version is no improvement. Why do you need a phone lock anyway? I think that the desire for one is often neurotic, and that class struggle and empire building may result from its use in an organizational setting. There are usually better ways to solve phone use problems.

The Useful Group

- *Remote Ring Indicator*—40 parts. Rings a distant bell.
- *Speakerphone*—36 parts. Not quite as good as Bell's. Identical to the Radio Shack design.
- *Pulse Programmer*—38 parts. A subproject of the Autodialer, it turns dial pulses into a string of beeps.
- *Autodialer*—46 parts plus a cassette player. A clever and cost effective burglar alarm dialer; perhaps the best large accessory.
- *Answering Machine*—75 parts plus two cassette recorders. This device is supposed to work better than Bell's because it can take longer messages. The circuit depends on the detection of silence. Since phone systems get noisier in bad weather, unreliable operation might result. However, if you have more time than money . . .
- *Silencer*—2 parts. A switch in series with the yellow ringer lead.
- *Hold Button*—6 parts. The best and most cost effective small accessory, and Bell can't supply one for a home phone! Has an LED.
- *Remote Ear*—57 parts. A speakerphone with provisions for single ring automatic answering, so that only you can listen in on your house or business while you are elsewhere.

The Useful, but . . . Group

- *Scrambler*—40 parts. A 1970 Popular Electronics project. If you are into a lifestyle which really requires secure conversations, take an old-fashioned tip from the mob and call phone booth to phone booth.
- *Digit Monitor*—44 parts. A lot of parts are used here to display only the *last* digit dialed, not all of the 11 possible.
- *Telecorder*—26 parts. Automatically records telephone conversations. A useful device, but I don't like this circuit. It adds some capacitance (and therefore ring load) to the line, and may be detectable by Bell. I have a friend who was hasseled for using a commercial version of this circuit . . . Also, depending on the specific cassette recorder and AC adaptor you might use, you could unbalance the line and get a bad hum. Or worse, you might place a ground fault on the telco central battery.
- *Teleswitch*—63 parts. Remote control of 120 vac appliances while you are away from home, by detecting single rings of your phone. The use of this accessory is clumsy, requires that you have a good memory, and could be unsafe. Gilder suggests turning on a coffee pot. In principle, I am opposed to the distant remote control of heating appliances other than a central furnace. Also, the casual description of mounting the 120 vac outlets on the chassis is an invitation to kluge wiring where there could be a heavy current draw. Good grief—there are no fuses either!

The book as a whole gives me the impression of having been written pretty fast, by someone who is fairly good at writing fast books. I don't object because the work does fill a gap in the literature. There is an entire spectrum of flaws, trivial through serious, but these telephone accessories more or less get their jobs accomplished at truly minimal cost.

Reviewed by Jim Berkey

INTRODUCTORY SPECIAL
Kingdom, One Queen, Matador, Poker
Personal Software
P.O. Box 136-E3
Cambridge, MA 02138
$12.95 (Cassette)

In the last issue of *RC* (or is it *PC*?; or *RC/PC*?; or *PC/RC*? —anyway, in the last issue) there is an article on the simulation game, KINGDOM. The Personal Software people (listed above) offer an introductory game tape that contains a version of KINGDOM. The tape has three other classic games as well: ONE QUEEN, MATADOR and POKER. I will discuss only KINGDOM in this review and comment on the rest of the games in a later issue.

KINGDOM, as presented in this version, is highly engaging. A result that is a direct function of the use of complete sentences and the pronoun 'you' in the program's communciations with the user. You, as the user, quickly feel that you are the ruler of an ancient kingdom. The reports from the program take on the aura of personal communications to you about your subjects, the harvests and the disasters that occur within the kingdom. One disaster in this simulation, which is not found in all variations, is that you may fall victim to assassination attempts during your reign. Also a variety of endgame messages are produced that comment on how the end of your rule affected your subjects and your treasury. KINGDOM is enjoyable and stimulating.

The tape I reviewed of these games was for a Level I BASIC 4K TRS-80 (PET programs are on the other side of the tape). I had the pleasure (?) of using the Level I-to-Level II TRS-80 program convertor to make the game operational on my system. Actually, the conversion went smoothly once I remembered to correctly set the volume controls and could locate the start of the programs on

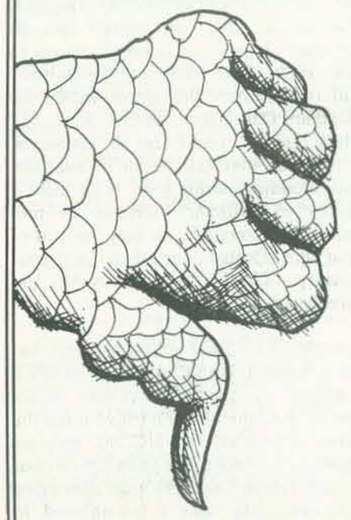
the tape. You, however, can request a version in Level II BASIC when you place an order.

I have several general comments about the game KINGDOM that apply to most versions of this simulation. Some of these items can be handled by the user making simple modifications to the program. Some require extensive reprogramming. Here is the list of comments without amplification: the rats always seem to spoil too much of the harvest; food production is just not profitable; the game reduces to a 'real estate' venture with the buying and selling of land a generally good strategy; there is no dependent logic built into the random events; the addition of another commodity or manufactured product might add to the richness of the game (more tradeoff strategies); a multi-player version of the game might prove interesting. I guess this list reduces to a single question: Where is KINGDOM II? If there are enhanced versions of this simulation in existence, send them to us for review and publication.

Reviewed by Ramon M. Zamora.

REVIEW

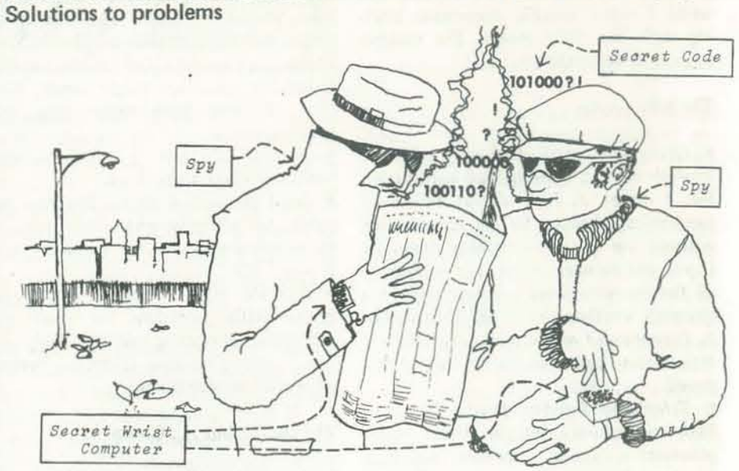
BATTLESTAR GALACTICA



To be continued . . .

DISTANCE AND ERROR CORRECTING CODES (see article on page 26)

Solutions to problems



Sample Solution — Problem 1

SOLUTIONS

```
>LISTNH

 10 S(1) = 0
 20 S(2) = 1
 85 PR."TWO-AGENT PROBLEM"
 90 PR. "TO END PROGRAM, TYPE THE CHARACTERS
    END WHEN ASKED FOR A CODE"
"
 95 PR." "
100 PR. "ENTER A 3-BIT CODE."
105 INPUT B$
106 IF B$ = "END" GOTO 185
110 GOSUB 200
115 S(3) = S1
120 S(4) = S2
125 IF (S(1) = S(3) AND S(2)=S(4)) GOTO 180
126 IF (S(1) # S(3) AND S(2)#S(4)) GOTO 140
129 X = 1-S(3)
130 IF (S(1) = X) GOTO 150
134 X= 1-S(4)
135 IF (S(2) = X) GOTO 160
140 B3$ = RIGHT(  STR(1-VAL(B3$)) ,1)
145 GOTO 165
150 B1$ = RIGHT(  STR(1-VAL(B1$)), 1)
155 GOTO 165
160 B2$ = RIGHT(  STR(1 - VAL(B2$)), 1)
165 PRINT" CODE IN ERROR"
170 PRINT" CODE SHOULD BE ":B1$+B2$+B3$
172 GOTO 95
175 STOP
180 PRINT" AUTHENTIC CODE"
182 GOTO 95
185 STOP
190 END
200 B1$ = LEFT(B$,1)
205 X$ = RIGHT(B$,2)
210 B2$ = LEFT(X$,1)
215 B3$ = RIGHT(B$,1)
220 S1 = VAL(B1$) + VAL(B3$)
225 S2 = VAL(B2$) + VAL(B3$)
230 IF S1 = 2 LET S1 = 0
235 IF S2 = 2 LET S2 = 0
240 RETURN
```


Problem 1 (cont.)

```
>RUN
TWO-AGENT PROBLEM
TO END PROGRAM, TYPE THE
CHARACTERS END WHEN ASKED FOR A CODE.

ENTER A 3-BIT CODE.
?000
 CODE IN ERROR
 CODE SHOULD BE 010

ENTER A 3-BIT CODE.
?001
 CODE IN ERROR
 CODE SHOULD BE 101

ENTER A 3-BIT CODE.
?010
 AUTHENTIC CODE

ENTER A 3-BIT CODE.
?011
 CODE IN ERROR
 CODE SHOULD BE 010

ENTER A 3-BIT CODE.
?100
 CODE IN ERROR
 CODE SHOULD BE 101

ENTER A 3-BIT CODE.
?101
 AUTHENTIC CODE

ENTER A 3-BIT CODE.
?110
 CODE IN ERROR
 CODE SHOULD BE 010

ENTER A 3-BIT CODE.
?111
 CODE IN ERROR
 CODE SHOULD BE 101

ENTER A 3-BIT CODE.
?END
```

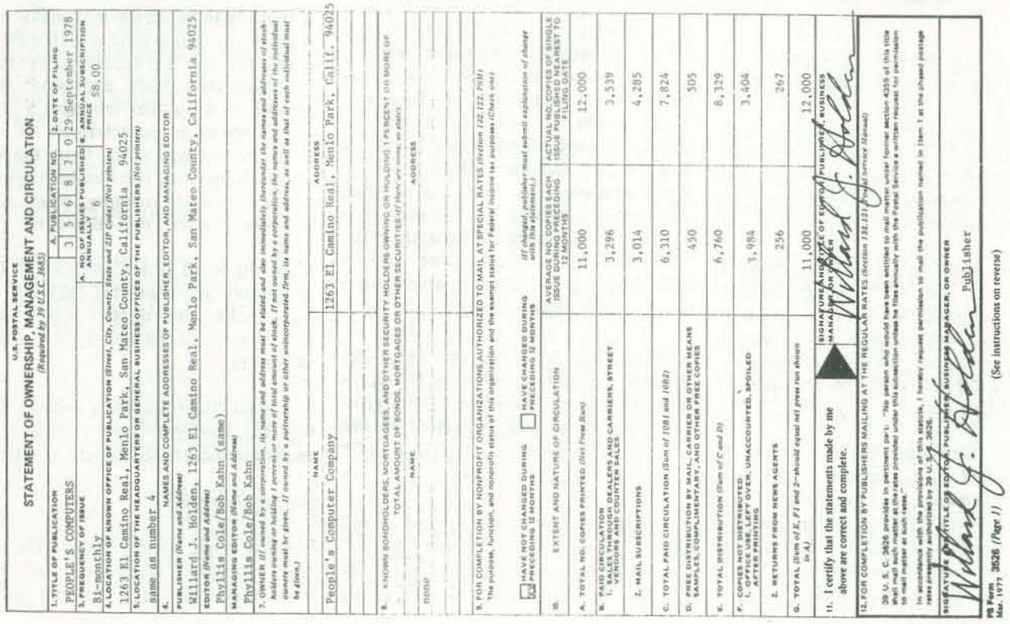
Sample Solution — Problem 2

```
>LISTNH

  5 PR. "FOUR-AGENT PROBLEM"
 10 PR."TO END PROGRAM, TYPE THE CHARACTERS
    END WHEN ASKED FOR A CODE"
 12 PR." "
 13 VAR = ZERO
 15 PR."ENTER A 5-BIT BINARY CODE"
 20 INPUT B$
 25 IF B$ = "END" GOTO 250
 30 B1$ = LEFT(B$,1)
 35 X$ = RIGHT(B$,4)
 40 B2$ = LEFT(X$,1)
 45 X$ = RIGHT(B$,3)
 50 B3$ = LEFT(X$,1)
 55 X$ = RIGHT(B$,2)
 60 B4$ = LEFT(X$,1)
 65 B5$ = RIGHT(X$,1)
 70 S(1) = VAL(B1$) + VAL(B2$)
 75 S(2) = VAL(B4$) + VAL(B5$)
 80 S(3) = VAL(B1$) + VAL(B4$)
 85 S(4) = VAL(B2$) + VAL(B5$)
 90 FOR I = 1 TO 4
 95 IF S(I) = 2 LET S(I) = 0
100 NEXT I
105 IF S(1) # 0 LET F1 = 1
110 IF S(2) = 0 GOTO 125
115 IF F1 = 1 GOTO 240
120 F2 = 1
125 IF VAL(B3$) # S(3) LET F3 = 1
130 IF VAL(B3$) # S(4) LET F4 = 1
135 K = F1 + F2 + F3 + F4
140 IF K = 0 GOTO 230
145 IF K # 2 GOTO 240
150 IF (F1 + F3) # 2 GOTO 165
155 B1$ = RIGHT( STR(1 -VAL(B1$)),1)
160 GOTO 215
165 IF (F1 + F4) # 2 GOTO 180
170 B2$ = RIGHT( STR(1 - VAL(B2$)),1)
175 GOTO 215
180 IF (F3 + F4) # 2 GOTO 195
185 B3$ = RIGHT( STR(1 - VAL(B3$)),1)
190 GOTO 215
195 IF (F2 + F3) #2 GOTO 210
200 B4$ = RIGHT( STR(1 - VAL(B4$)),1)
205 GOTO 215
210 B5$ =RIGHT( STR(1 - VAL(B5$)),1)
215 PR." "
220 PR." 1 BIT ERROR---CORRECT CODE IS
    "+B1$+B2$+B3$+B4$+B5$
225 GOTO 12
230 PR." AUTHENTIC CODE"
235 GOTO 12
240 PR." ERROR = 2 BITS --- DOUBLE AGENT!"
245 GOTO 12
250 STOP
260 END
```

Problem 2(cont.)

```
>RUN
FOUR-AGENT PROBLEM
TO END PROGRAM, TYPE THE CHARACTERS
END WHEN ASKED FOR A CODE

ENTER A 5-BIT BINARY CODE
?00000
 AUTHENTIC CODE

ENTER A 5-BIT BINARY CODE
?10000

 1 BIT ERROR---CORRECT CODE IS 00000

ENTER A 5-BIT BINARY CODE
?00100

 1 BIT ERROR---CORRECT CODE IS 00000

ENTER A 5-BIT BINARY CODE
?00010

 1 BIT ERROR---CORRECT CODE IS 00000

ENTER A 5-BIT BINARY CODE
?11011
 AUTHENTIC CODE

ENTER A 5-BIT BINARY CODE
?11010

 1 BIT ERROR---CORRECT CODE IS 11011

ENTER A 5-BIT BINARY CODE
?10011

 1 BIT ERROR---CORRECT CODE IS 11011

ENTER A 5-BIT BINARY CODE
?10101
 ERROR = 2 BITS --- DOUBLE AGENT!

ENTER A 5-BIT BINARY CODE
?10110
 ERROR = 2 BITS --- DOUBLE AGENT!

ENTER A 5-BIT BINARY CODE
?END
```

□



U.S. POSTAL SERVICE
STATEMENT OF OWNERSHIP, MANAGEMENT AND CIRCULATION
(Required by 39 U.S.C. 3685)

1. TITLE OF PUBLICATION: PEOPLE'S COMPUTERS
A. PUBLICATION NO.: 3 5 6 8 7 0
2. DATE OF FILING: 29 September 1978
3. FREQUENCY OF ISSUE: Bi-monthly
A. NO. OF ISSUES PUBLISHED ANNUALLY: 6
B. ANNUAL SUBSCRIPTION PRICE: $8.00
4. LOCATION OF KNOWN OFFICE OF PUBLICATION (Street, City, County, State and ZIP Code) (Not printers): 1263 El Camino Real, Menlo Park, San Mateo County, California 94025
5. LOCATION OF THE HEADQUARTERS OR GENERAL BUSINESS OFFICES OF THE PUBLISHERS (Not printers): same as number 4
6. NAMES AND COMPLETE ADDRESSES OF PUBLISHER, EDITOR, AND MANAGING EDITOR
PUBLISHER (Name and Address): Willard J. Holden, 1263 El Camino Real, Menlo Park, San Mateo County, California 94025
EDITOR (Name and Address): Phyllis Cole/Bob Kahn (same)
MANAGING EDITOR (Name and Address): Phyllis Cole/Bob Kahn
7. OWNER (If owned by a corporation, its name and address must be stated and also immediately thereunder the names and addresses of stockholders owning or holding 1 percent or more of total amount of stock. If not owned by a corporation, the names and addresses of the individual owners must be given. If owned by a partnership or other unincorporated firm, its name and address, as well as that of each individual must be given.)
NAME: People's Computer Company
ADDRESS: 1263 El Camino Real, Menlo Park, Calif. 94025
8. KNOWN BONDHOLDERS, MORTGAGEES, AND OTHER SECURITY HOLDERS OWNING OR HOLDING 1 PERCENT OR MORE OF TOTAL AMOUNT OF BONDS, MORTGAGES OR OTHER SECURITIES (If there are none, so state): NAME / ADDRESS — None
9. FOR COMPLETION BY NONPROFIT ORGANIZATIONS AUTHORIZED TO MAIL AT SPECIAL RATES (Section 132.122, PMM) The purpose, function, and nonprofit status of this organization and the exempt status for federal income tax purposes (Check one)
☐ HAVE NOT CHANGED DURING PRECEDING 12 MONTHS ☐ HAVE CHANGED DURING PRECEDING 12 MONTHS (If changed, publisher must submit explanation of change with this statement)

10. EXTENT AND NATURE OF CIRCULATION	AVERAGE NO. COPIES EACH ISSUE DURING PRECEDING 12 MONTHS	ACTUAL NO. COPIES OF SINGLE ISSUE PUBLISHED NEAREST TO FILING DATE
A. TOTAL NO. COPIES PRINTED (Net Press Run)	11,000	12,000
B. PAID CIRCULATION 1. SALES THROUGH DEALERS AND CARRIERS, STREET VENDORS AND COUNTER SALES	3,296	3,539
2. MAIL SUBSCRIPTIONS	3,014	4,285
C. TOTAL PAID CIRCULATION (Sum of 10B1 and 10B2)	6,310	7,824
D. FREE DISTRIBUTION BY MAIL, CARRIER OR OTHER MEANS SAMPLES, COMPLIMENTARY, AND OTHER FREE COPIES	430	505
E. TOTAL DISTRIBUTION (Sum of C and D)	6,740	8,329
F. COPIES NOT DISTRIBUTED 1. OFFICE USE, LEFT OVER, UNACCOUNTED, SPOILED AFTER PRINTING	3,984	3,404
2. RETURNS FROM NEWS AGENTS	256	267
G. TOTAL (Sum of E, F1 and 2—should equal net press run shown in A)	11,000	12,000

11. I certify that the statements made by me above are correct and complete.
SIGNATURE AND TITLE OF EDITOR, PUBLISHER, BUSINESS MANAGER, OR OWNER: Willard J. Holden, Publisher

12. FOR COMPLETION BY PUBLISHERS MAILING AT THE REGULAR RATES (Section 132.121, Postal Service Manual)
39 U. S. C. 3626 provides in pertinent part: "No person who would have been entitled to mail matter under former section 4359 of this title shall mail such matter at the rates provided under this subsection unless he files annually with the Postal Service a written request for permission to mail matter at such rates."
In accordance with the provisions of this statute, I hereby request permission to mail the publication named in item 1 at the phased postage rates presently authorized by 39 U.S.C. 3626.
(See instructions on reverse)

PS Form 3526 (Page 1)
Mar. 1977

ANNOUNCEMENTS

HARDWARE

TALKING LEARNING AID INTRODUCED BY TEXAS INSTRUMENTS

An innovative talking learning aid called Speak & Spell™ that helps children learn how to spell and pronounce over 200 basic vocabulary words was introduced by Texas Instruments.

Speak & Spell, utilizing TI's advanced Solid State Speech™, helps children sharpen word skills scientifically and is intended to provide a supplement to regular classroom language instruction. Speak & Spell was created with the guidance of leading educators, according to TI.

"The first 200 words for Speak & Spell were selected on the basis of studies that indicated they are most needed by children in the seven to 12 age group—mainly because these words do not follow 'the rules,' but have to be memorized," the spokesman said. "They include words most frequently misspelled from kindergarten days into adulthood—words like anxious, ocean, language, obey and learn."

Speak & Spell offers an entirely new concept in speech reproduction. "Unlike tape recorders and pull-string phonograph records used in recent years in many 'speaking' toys," the TI spokesman observed.

TI's Solid State Speech circuitry has no moving parts. A single integrated circuit, smaller than an aspirin, has all the digitally-coded data necessary to produce synthetic speech electronically. Word models are stored in a read-only memory (ROM). When the unit is told to say something, it pulls a word model from memory, processes it through the integrated circuit and speaks electronically through a standard loudspeaker.

How It Works

In its main mode of operation, Speak & Spell randomly pronounces a word in standard American English (including proper inflections). When the child keys-in his spelling on the alphabetic keyboard, the word is shown visually on the unit's eight-character display screen. Right answers earn verbal and visual praise; wrong answers are followed by a patient encouragement to try again. After every set of 10 words, the voice of Speak & Spell announces the score, which also is shown on the display screen.

A Speak & Spell student progresses through four levels of over 50 words each, setting his own pace as he goes. Level one starts with beginners' words; as he gains confidence and skill, he progresses to tougher challenges.

A number of games are offered to intrigue pre-teenagers. "Mystery Word," for example, is a word guessing game similar to the generations-old game of "Hangman." Other games call for word completion or solution of secret codes.

A Speak & Spell feature game aimed at improving pronunciation is "Say It," which displays a word, pauses to give a child a chance to say it out loud, then gives the correct pronunciation.

Speech Technology

The state-of-the-art Solid State Speech TI developed for Speak & Spell is based on a solid-state read-only memory similar to that used in the company's programmable handheld calculators. The 128K dynamic ROM is believed to be the first production device that incorporates over 100,000 electronic functions on a single integrated circuit chip. TI expects to introduce additional word lists in coming months that will increase a unit's capabilities manyfold. These word lists will be interchangeable modules which will plug into the back of Speak & Spell.

A spokesman explained that analog audio signals representing spoken works are processed through a complex mathematical procedure that divides each word into a number of slices that represent a word's most distinctive characteristics as digits rather than sound waves. Once the word has been digitized, it is stored in Speak & Spell's solid-state memory, much as the numbers are stored in a calculator.

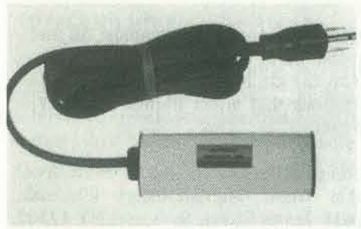
When the word is needed, it's called out of memory and converted back into analog impulses that drive a regular audio speaker to form the sounds that make up each word.

TI's Solid State Speech is believed to be the first low-cost, mass-produced unit which produces speech electronically.

Price & Delivery

Speak & Spell, which comes with a 28-page illustrated instruction and game book, uses four C-cell batteries (not included), weighs approximately one pound and measures 10 inches by 6.5 inches by 1.5 inches. It is warranted against defective materials and workmanship for one year, and has a suggested retail price of $50.

Texas Instruments Incorporated, Consumer Relations, P.O. Box 53 (Attn: Learning Aids), Lubbock, Texas 79408.



GLITCH KILLER

Glitches, false print-out and memory loss are often caused by lightning and machinery surges or AC power line hash. The AC Line Cord Suppressor/Filter being introduced by Electronic Specialists will bring glitch causing hash and surges under control.

Capable of handling up to a 1000 watt load, the line cord suppressor/filter provides both line-to-line and line-to-ground protection. Model S/F-KW-3 priced at $22.50. Electronic Specialists, Inc., Box 122, Natick, MA 01760 Tel: (617) 655-1532.

RCA ADDS COLOR TO VIP

RCA today demonstrated its new expansion board which will allow its VIP personal computer to provide video displays in eight colors.

RCA's basic VIP unit has heretofore been limited to black and white display on a video monitor or, through an external modulator, onto a standard television set.

The VP-590 Color Board, which will retail for $69.00, allows the user to select one of three background colors for his display; he then specifies one of eight foreground colors for each of sixty-four screen areas. Any bit which is turned on in an area will be displayed in the foreground color, while 'off' bits in the area will display the background color. Both foreground and background colors can be changed at any time.

Software support for the VP-590 will be provided through CHIP-8C, a color graphics language which is upward compatible with the CHIP-8 language used on the present VIP. The VP-590 plugs directly into a socket provided on both existing and future VIP units.

According to Richard Simpson, Product Manager for the VIP, "The low cost of the VIP Color Board is a result of using the new RCA CDP1862 color display IC circuit. The CDP1862 was specifically designed to interface with the CDP1861 circuit, which is used to generate the VIP's video graphics."

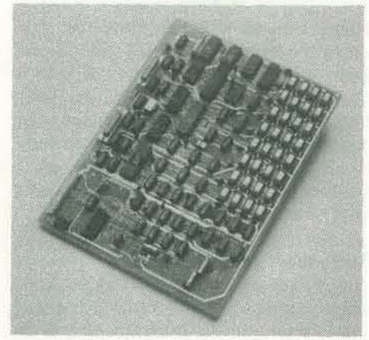
For further information, call Rick Simpson at (717) 291-5848, or write RCA COSMAC VIP Marketing, New Holland Avenue, Lancaster, PA 17604.

MICROSTAR™/5 SINGLE BOARD COMPUTER

The MicroStar/5 Single Board Computer is a complete microprocessor based computing system available on a single 9½ x 13½" board. It supports the CP/M™ disc operating system including BASIC, FORTRAN IV and COBOL.

The MicroStar/5 integrates a microprocessor (INTEL 8085), Floppy Disc Controller (INTEL 8271), Random Access Memory (16KB to 64KB), Real Time Clock, Direct Memory Access, Programmable Interrupt Controller and 2 RS232 peripheral/terminal interfaces. A 32K RAM version of the MicroStar/5 is priced at $1,270.00 in OEM quantities.

The system will interface a variety of IBM compatible floppy disc drives including Shugart SA 800 and SA 850, the CALCOMP 142M and 143M, the PerSci 277 and the Shugart mini floppy



SA 400 and SA 450. Printers available with the system—include Teletype, Centronics, and Diablo.

System software includes a macro assembler, interactive debugger, interactive editor and text processor. A bootstrap loader is an integral part of the system stored in ROM. It runs diagnostics and provides automatic loading from disc.

A CP/M application software development system including 32K RAM and dual floppy disc drive (512KB) is $3,995.00 with RS232 interfaces for video display terminal and line printer.

Micro V Corporation recently announced the MicroStar/55 Small Business System with a data base management system called UPDATE™. This permits an English dialog with the system.

From: MICRO V, 17777 S.E. Main St., Irvine, CA 92714 (714) 957-1517.

RCA ANNOUNCES MUSIC SYNTHESIZER FOR VIP

RCA demonstrated its new expansion board which will allow its VIP personal computer to generate music.

The VP-550 Super Sound Board will retail for $49.00, and is expected to be available about October 15, 1978. The board is completely assembled and will be sold through RCA industrial distributors and many computer stores.

RCA will also provide a computer program called PIN (Play It Now) which will allow VIP users to easily transcribe sheet music to play on the VIP, or compose their own music.

According to Richard Simpson, Product Manager for the VIP, "This low-cost add-on to the VIP will allow the user to generate a wide variety of sounds over a four-octave range. Because the user has control over the tone envelope, he can simulate the sound of various conventional musical instruments, such as a clarinet, or create 'new instruments' with unusual sounds."

Since the VIP Music Board can generate any frequency within its range, users may program it for the conventional music scale, non-traditional scales, or arbitrary sounds which can be used to generate sound effects rather than music.

The Music Board has two independent outputs so that the user can program both melody and harmony or two-part harmony.

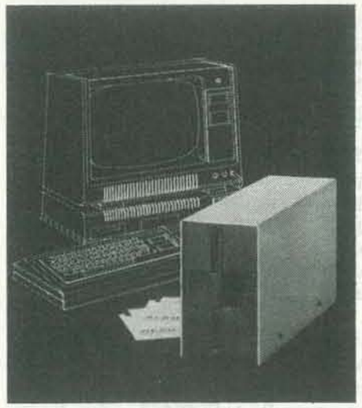
Provisions are made to create up to six-part music using an external four-channel tape recorder. In addition, the PIN program can control a readily available drum synthesizer for accompanying percussion effects.

The VIP is a microcomputer based on the RCA COSMAC (CDP1802) microprocessor, and is designed to interface directly with a video monitor or through an external modulator onto a TV set. It is provided with an interpretive language which makes it easy for the user to write graphic games and other applications without having to learn machine language. The VIP contains a 16-key keypad for entering programs and has a built-in audio cassette interface to permit storing programs on a cassette recorder. Documentation provided with the VIP contains listings for twenty games for use on the system.

For further information, call Rick Simpson at (717) 291-5848, or write RCA COSMAC VIP Marketing, New Holland Avenue, Lancaster, PA 17604.

PERCOM MANUFACTURING ADD-ON DISK DRIVES FOR TANDY RADIO SHACK TRS-80 COMPUTER

PerCom Data Company announced here today that the company is now manufacturing "add-on" minifloppy* disk drives for the Tandy Radio Shack TRS-80 computer.



The PerCom unit, which includes the drive, drive power supply, and enclosure, is identical in all important respects to the TRS-80 Mini-Desk System. The PerCom unit sells for $399.00. The Radio Shack Mini-Disk System lists at $499.00.

The drive itself is the proven Shugart SA-400—the same drive used in the Radio Shack unit. The data transfer rate is 125 kilobits per second. Access time is a fraction of a second. The drive power supply features overload current limiting and thermal protection.

Interfacing of disk drives to the TRS-80 computer is accomplished with the Radio Shack TRS-80 Expansion Interface, which accommodates up to four drives (and other peripherals), and includes controller electronics and a four-drive cable. Operating software for all drives is obtained by the user with the purchase of the first drive from Radio Shack.

For more information, please contact the PerCom Data Company, 318 Barnes, Garland TX 75042, (214) 272-3421.

* Trademark

CONVERT YOUR PRESENT IBM SELECTRIC® TO A COMPUTER PRINTER

ESCON conversion kits let you convert a standard IBM Selectric typewriter into a quality printer in just a few hours. No holes to drill. No mechanical genius needed. And no changes in the appearances or normal operation of your IBM Selectric once conversion is complete.

Interface: Compatible with most computer systems utilizing the S-100 buss—ALTAIR, IMSAI, SOL Poly 88, Equinox, and many more. Consists of a single card utilizing parallel output, ASCII coded. The multiple pin connector is compatible with all S-100 mother boards.

Power supply: Optional. Provides 30 VDC @ 2A. Customized power output supply is matched to the requirements of the kit. Includes computer-grade capacitor and driver power transistors to operate solenoids.

Mechanical parts and solenoids: All mechanical parts and solenoids are provided with the kit. No drilling of holes or modifications that affect normal use of typewriter.

Complete kits are priced at $450. Available from: MiniMicroMart Wholesale, 1618 James Street, Syracuse, NY 13202. Phone (315) 422-4467.

COMPUTER JUGGLES MANY TASKS

S-100 computers can handle simultaneous problems when outfitted with the MULTITASKER, an interrupt handling board from Objective Design, Inc. Having interrupts in the system allows one coputer to do the work of many. For example, with an interrupt driven system you can: handle program development on several terminals at once; run household appliances and play games at the same time; and continue using the computer while a slow printer is churning out a listing.

The MULTITASKER can be located anywhere in memory—and will generate 'CALL' vectors to any location. The assembly language software required for handling interrupts is available on PROM (which goes into space provided on the board), making the interface into a high level, relatively easy job.

Another MULTITASKER board option is a crystal derived Real Time Clock. Timed interrupt intervals are hardware selectable from 100 microseconds to 100 milliseconds. Software counters can extend this time into days or years.

MULTITASKER without PROM in kit form is $205.95 from Objective Design, P.O. Box 20325, Tallahassee, FL 32304, phone (904) 224-5545. Shipping costs are $5.00 Canada; $20.00 overseas. Call factory direct for further info.

ADAM AND MIKE

ADAM is the logical machine that has revolutionized the small business computer market. It is the first business system that can actually be instructed by the business owner in plain everyday English.

MIKE is the latest addition to this product line. It is a voice recognition and response unit that allows you to control ADAM by speaking.

ADAM and MIKE are designed to complement each other. The ease of operating ADAM coupled with MIKE's voice data entry, voice inquiry and voice response capability provide a system confined only by the dimensions of the user's imagination and ingenuity. Your voice and chosen vocabulary will allow you to have access to information stored in ADAM without the use of the keyboard. This means the business owner can speak to ADAM via a microphone and ask for 'year-to-date sales' or 'net profit' and receive either a printed report, screen display, or voice response.

You may give MIKE a request to have ADAM display information such as management financial figures, inventory status, order records, credit checks, or any other business data being processed by ADAM. You may also use MIKE to enter data into ADAM's files. New orders, daily shipments, payroll hours, and product inspection information are a few examples. MIKE can also confirm your inputs verbally to ensure accurate data entry.

MIKE is designed to learn and recognize a user-specified vocabulary consisting of up to sixteen (16) words or phrases. ADAM can store multiple vocabulary sets of sixteen words that can be recalled upon voice or keyboard command. MIKE is also capable of recording voice response vocabularies and producing natural sounding verbal responses to requests for information or for confirmation of voice data entry items. Contact: Logical Machine Corporation, 1294 Hammerwood Avenue, Sunnyvale, CA 94086, (408) 744-1290.

SOFTWARE

PERSONAL LEDGER FOR COMMODORE PET™ *

Channel Data Systems introduces Personal Ledger, a complete double entry bookkeeping system with provisions for budgeting and keeping records of income, deductible and non-deductible expenses, assets and liabilities. Its interactive features enable entering transactions, adding or editing accounts, and printing of a detailed Income Statement and Balance Sheet.

Up to 50 accounts are allowed with names and budgets specified by the user. An audit trail of all entered transactions is printed on the printer of your choice or on the screen if you do not have a printer. All data is stored on cassette, loaded prior to entering transactions and stored after entering transactions. There is no waiting for printing to the tape during operation of the system. Extensive error recovery features are included to allow reentry of an erroneous instruction or value.

Personal Ledger is supplied on cassette, along with a complete manual, program listing and sample data for only $20.00 by Channel Data Systems, 5960 Mandarin Avenue, Goleta, CA 93017, or telephone (805) 964-6695.

* 8k Bytes RAM Required

A NEW GUIDE TO LEARNING THE APL LANGUAGE

A combination workbook/textbook offering a problem-solving approach to learning computer programming in APL has just been published by Hayden Book Company, Inc. It's entitled: *APL: AN INTRODUCTION*, by Howard A. Peelle.

The guide is self-instructional; that is, you can teach yourself the APL language by using this book—with or without a computer. Each chapter opens with an explanation of APL problem-solving tools, followed by numerous examples of APL expressions. The reader is then asked to solve selected exercises, which can be done with or without a computer. Answers are provided in the appendix.

The book is written in an informal style, with handwritten annotations that serve as supplementary explanation alongside examples of APL expressions. Each chapter ends with a review, and summary tables of all expressions appear at the end of the book for quick reference. 256 pages, 8½ x 11, paper, $8.50.

Look for Howard's articles in this issue.

NORTHSTAR USERS GROUP RELEASES PROGRAM LIBRARY

Some 300+ programs on 20 diskettes are now available for a small copying charge plus the price of a new diskette from the Northstar Users Group. The disks are designed to run in microcomputer systems that utilize the Northstar DOS and Northstar BASIC, a high level interpreter.

The library includes business and finance programs, mailing list programs, math, programming utilities, a PILOT interpreter, a Palo Alto Tiny Basic interpreter and an enormous number of games. Computer games are still a pet fancy among cómputer hobbyists and amateur programmers.

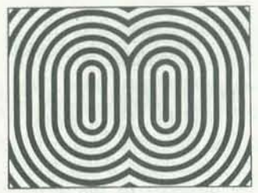
Further information can be obtained by sending a self-addressed stamped envelope to J. Dvorak, Authorized NSUG Program Library Distributor, 704 Solano Ave., Albany, CA 94706.

SOFTWARE PACKAGED IN NORTH STAR FORMAT

MicroAge has announced software packaged on mini-diskette in North Star format. Each disk is ready to run on any S-100, 8080/Z80 computer system. Current offerings include:

- Financial programs from 'Some Common BASIC Programs', by Osborne & Associates.
- Mathematical analysis programs from 'Some Common BASIC Programs.'
- Games. Vol. 1. Trap, Batnum, Hurkle, Stars, Reverse, Mathdrill I, Cannon, Chomp, Weekday, Calendar, Pony.
- Games. Vol. 2. Includes Button, Frog, String, Change, Civilwar, Golf, Golfhand, Chase!, Shooting Star, Lunar Lander, Mathdrill II.
- Backorder program using disc data files.
- Mailing list using disc data files.
- Northstar DOS for the Centronics printer.
- Retail sales reporting using disc data files.

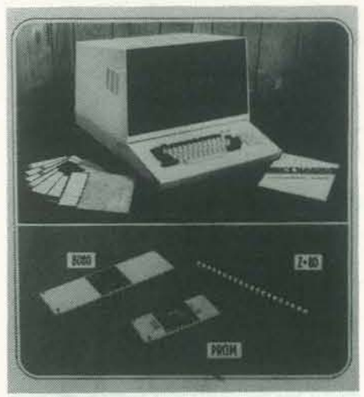
The price for each package is $35. This software is available from local computer stores or from MicroAge Mail Order, 803 N. Scottsdale Road, Tempe, AZ 85281.



XYBASIC FOR 8080-BASED SYSTEMS

The Mark Williams Company announces XYBASIC, a high level language for process control and data acquisition.

Control features include a software-interrupt feature (ENABLE) which effectively multiplies the power of your computer. It allows you to monitor external devices and execute a program at the same time. ENABLE automatically checks to see if the specified condition is met before executing each program statement. In addition, a DELAY command builds real-time delays into the program without the addition of a real-time clock to the system.



Other control features include: PEEK and POKE commands to examine and modify any location in the computer's memory; IN and OUT commands to input or output at the machine level; and SENSE for looking at any individual bit on any port. In addition, XYBASIC has a number of bit manipulation commands usually only found in an assembler such as ROTATE, SHIFT, and TEST.

Another significant feature is XYBASIC'S ROMability. With the ROM version, XYBASIC appears instantly when the system is turned on. (Of course, XYBASIC is available on paper tape, cassette tape, or floppy disk if desired.) And what's more, a special ROM Squared version lets you create stand-alone systems. This means that both XYBASIC and your XYBASIC program can be run in ROM. The system begins running simply by hitting the RESET button.

All XYBASIC versions are available from stock for as little as $295.00. The XYBASIC Programming Manual may be purchased separately for $20.00. Mark Williams Company, 1430 West Wrightwood Avenue, Chicago, Illinois 60614 Telephone: Area Code 312 472-6659.

TELECOMMUNICATION SYSTEM ANNOUNCED FOR MITS 8800b MICRO COMPUTER

The Microware-PBS has announced a telecommunication system for the MITS Altair 8800b microcomputer. This system enables the MITS 8800b to be used as an intelligent terminal in two way communication with a remote time sharing device over the telephone lines using an acoustic coupler.

The telecommunication system consists of an assembly language routine and another program written in MITS disk extended BASIC. Control commands are provided to enable the operator to switch command console communication from the remote device to MITS and vice versa. This system may be used to communicate and transfer any kind of data files to and from a remote time sharing device at speeds up to 300 baud. The hardware requirements are MITS 8800b with 32K core memory and one or more floppy disks. It is supplied on an 8" floppy disk with an 18 page User's Manual. The system is priced at $195.00 which includes the source listing. The User's Manual may be purchased separately for $20.00 for system evaluation and its cost may be applied towards the purchase of the telecommunication system at a later date. For more information contact: Mr. Mark Shelton, Microware-PBS, P.O. Box 47, Blacksburg, VA 24060.

THE RECREATIONAL PROGRAMMER™

The Recreational Programmer is a novel small journal for sophisticated users of computers and programmable calculators. Total emphasis is devoted to leisure uses for computers and programmable calculators. The owner who enjoys games, graphics, astronomy, sailing, amateur radio, or recreational mathematics and programming puzzles will find *The Recreational Programmer* a rich source of usable, well-documented programs and articles.

Each program features a flow diagram written in English, and patterned as a BASIC program. This allows the user of any computer programmable in BASIC, such as the PET, the TRS-80, the APPLE, or any of the numerous S-100 bus systems to easily use any program on his or her machine.

Programmable calculator owners will find that each useful flow diagram is accompanied by a thoroughly commented listing for any of a variety of personal calculators! Hewlett-Packard and Texas Instruments programmables are always represented in each issue.

The Recreational Programmer is printed on sturdy text in an eye-pleasing format. Each issue's cover displays striking artwork to insure that the reader will enjoy receiving and using this sensibly priced bimonthly magazine.

Readers interested in subscribing to *The Recreational Programmer* may send $12 ($15 outside the USA and APO's) to *The Recreational Programmer*, P.O. Box 2571, Kalamazoo, Michigan, 49003.

The Recreational Programmer is published bimonthly by Programmers Publishing Co., 3013 Cameron Street, Kalamazoo, Michigan, 49003, (616) 343-3546.

OTHER

GUIDE TO TRS-80™ INFORMATION

The *Guide* covers: addresses, articles, authors, bugs, comments, hardware, ideas, phone numbers, predictions, reduced prices, rumors, software, sources, tips, used unit and user groups, and wanted. One *Guide*: $3 ea. 2 to 9: $2.50 ea. 10 or more: $1.90 ea. Full refund on return, if not satisfied. Send check or M.O. with Name, Address & Zip to: F.E. Huebner, P.O. Box 37206, Oak Park, MI 48237.

RADIO SHACK TO OPEN 50 COMPUTER CENTERS

Plans to open 50 computer sales and service stores in 1978–9 were revealed by Lewis Kornfeld, president of Tandy Corporation's Radio Shack division.

"While some will be located within new or existing Radio Shack stores, most will be separate entities and all are expected to be in major markets," he said.

"They will be called *Radio Shack Computer Centers*, and their purpose," according to Kornfeld, "will be to assist area Radio Shack stores in answering computer questions and closing sales, and to develop quantity sales, principally of Radio Shack TRS-80 Microcomputer Systems and peripheral equipment, to businesses and institutions.

"The stores," Kornfeld continued, "will provide market area service on Radio Shack computer products, thus extending to nearly 100 the number of service facilities operated by Radio Shack in this country, and will include classroom areas where the Company can teach computer use and programming to its customers and prospects."

In addition, the new Radio Shack Computer Centers will display and sell "a variety of pieces and parts, as well as packaged software and, possibly, hardware items of makes other than Radio Shack," Kornfeld stated.

"In respect to the significant number of TRS-80 personal computers we have sold," he added, "we believe the time is right to quickly expand our ability to keep them in top working order and provide solid local know-how for our customers and employees alike."

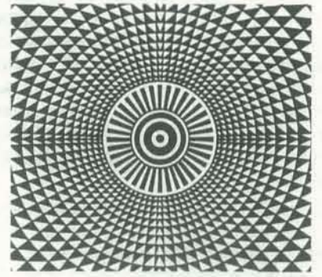
"We have converted the former Tandy Computer store, located in Fort Worth, into a Radio Shack Computer Center, and are in the process of identifying the other 49 or so cities and exact locations."

"To date we have selected our warehouse store in Garden Grove, California, and our late-fall-opening shop in Chicago's Water Tower Place as two sites for Computer Centers within Radio Shack stores. We are also considering converting some existing Radio Shack stores to Computer Centers in order to hasten the day when we are truly nationwide with this project." Capital expenditures for the 50 centers are expected to total about $4 million.

Ted F. Cromer, formerly General Manager of Tandy Computers and a veteran of 13 years with IBM, has been selected as Director of Radio Shack Computer Centers. He will be assisted by the Radio Shack district manager in the area of each center.

According to Kornfeld, "Within each Computer Center we expect to see at least one sales manager with extensive computer experience, a technician, and clerical help."

"Several hundred new job openings have been created by the launching of this program, and in addition, Radio Shack's Fort Worth computer factory continues to expand in people and space." The factory is one of 20 owned and operated by the consumer electronics chain.



TRS-80 MODS IMPROVE COMPUTER

Radio Shack TRS-80 computer users can have both Level I and II Basics in the same machine, can have lowercase, can reduce the maze of cords, and can get some of the information about how to do these things free.

The second edition of the free Computer Information Exchange Newsletter tells how to shoehorn both Level I and II onto the same circuit board. It is available by sending SASE (with 15¢ postage) to CIE, Box 158, San Luis Rey, CA 92068

The lowercase articles, including what to do about the Radio Shack Level II blooper, are in CIE's magazine, *TRS-80 Computing* 1:1, subscriptions available for $10 (12 issues).

Also in the 32-page first edition of the magazine are articles by TRS-80 designer Steven W. Leininger, by a Radio Shack repairman, and a couple of programmers. Also, how to install your own 16K memory chips, and save. A complete set of schematics is reproduced, on 10 pages. The magazine is standard 8½ by 11 inches.

Upcoming in issue 1:2 is a complete listing of an 8K-byte journal/ledger program, which runs in either Level I or II.

1802 USERS GROUP

A national 1802 user's group is being formed for ALL 1802-based computers, including Basic-ELF, ELF-II, Super-ELF, VIP, Infinite, etc. Although only a few months old, they already have over 400 members. Actual membership is free, and each member gets a free listing in our official newsletter, "THE PERIPHERAL", which is published monthly and furnished at cost to subscribing members [$5.00/year – 12 issues – 8-pages minimum]. There is one basic requirement for all members: When corresponding, please include a loose 15¢ stamp inside your letter, to help defray postage costs. Anyone interested in joining, please write directly for further details. Patrick Kelly, P.O. Box 7162, Los Angeles, CA 90022.

MSU STUDENT DEMONSTRATES 'TALKING COMPUTER'

Use of Michigan State University's "talking computer" is being demonstrated at a U.S. Government interagency conference in Washington, D.C., Wednesday (Sept. 6) by an MSU student who is handicapped.

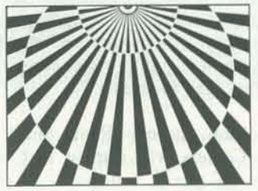
James Renuk, who has cerebral palsy and cannot command his vocal tract to produce speech, will deliver his remarks from his wheelchair with use of the MSU computerized portable communications system. Renuk is a fourth year food science major from Westland, Mich. He hopes to become a medical researcher.

The portable system being used by Renuk was developed at the MSU laboratory and incorporates a modified Heathkit Microcomputer with a modified Phonic Mirror HandiVoice developed by Federal Screw Company of Detroit. The Detroit product is an electronic simulation of the human voice tract.

Renuk is able to use the microcomputer which he carries on his wheelchair.

Many similar devices, Renuk contends, will help people to work with the young handicapped in order to evaluate the intellectual and physical potentials of these children.

Support for the MSU program and services to students like Renuk, as well as to the handicapped in public schools, has come from the Wayne County Intermediate School District, Northville Public Schools, the United Cerebral Palsy Association of Michigan and the federal government's funding of research and development of such communication devices. Contact: Fran Murray, (517) 355-2293, MSU Department of Information Services, East Lansing, Michigan 48824.



NATIONAL COMPUTER TOURNAMENT

Looking for a challenging project for your strategic instincts and your programming skills? This computer tournament may be for you. There is no charge to enter.

The tournament is based on a nifty little game called the Prisoner's Dilemma. In the Prisoner's Dilemma there are two players. Unlike most games, such as chess, the two players are not in total conflict. In fact, both can do well or both can do poorly.

Here is how the tournament works. The game will be played for an average of 200 moves, and in each move, each player can choose either to cooperate or to defect. If both cooperate, both do well. But if one defects while the other cooperates, the defecting player gets his highest payoff, and the cooperating player gets taken for a sucker and gets his lowest payoff. The catch is that if both defect, both do poorly.

The precise payoffs in the tournament for a given move are 3 points each if both cooperate; 5 points to a player who defects while the other cooperates, with 0 points to the sucker; and 1 point each if both defect. The score of a player in a single game is his or her total over all the moves.

To win the tournament you have to get the highest total score summed over all the games you play. Therefore your object is to get a good score in each separate game, but *not necessarily* to get a better score than the player with whom you are currently playing.

To join the computer tournament you submit a program written in BASIC or FORTRAN IV which will be a decision rule for the selection of the cooperative or the defecting choice at each move. The decision rule may be based on the history of the game so far. For example, a simple and pretty effective decision rule is TIT FOR TAT: cooperate on the first move, and then do exactly what the other player did on the previous move.

Quite sophisticated decision rules can be written in as little as 25 lines.

This tournament is part of a research project to understand the nature of skillful performance in a two-sided environment which is partially cooperative and partially competitive.

Each person who completes an entry will receive a report describing the results of the tournament. The winner will receive a handsome engraved trophy.

To get further details on the tournament, write to Professor Robert Axelrod, Institute of Public Policy Studies, The University of Michigan, 506 E. Liberty St., Ann Arbor, Michigan 48104.

THE CHANNEL DATA BOOK FOR PET USERS

Channel Data Systems will publish a hardware/software reference service for users of the Commodore PET™ personal computer. The Channel Data Book is a user-oriented directory of PET-related products including:

- Software
- Hardware and Peripherals
- Literature and Periodicals of special interest to PET users
- Listings of user groups and distributors
- Cross references by product type and supplier

Designed as a personalized working tool, the Channel Data Book provides a reference service for PET-related products, plus convenient dividers and color coding to organize programs, articles, and newsletters of specific interest to each user. Special sections for filing correspondence you have received from Commodore and flyers from other product suppliers are also provided. Flyers from suppliers of PET-related products who elect to advertise in the Data Book will be included in product sections. Venders that market PET-related products or services should be sure to contact CDS, as a summary of the information will be included at no cost.

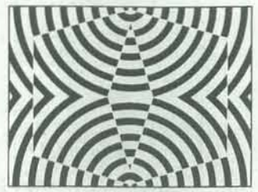
The Data Book includes a 3-ring binder and updated supplements with instructions for filing new and revised material. The price of $19.95 includes the Channel Data Book and update service through calendar year 1979. Order from Channel Data Systems, 5960 Mandarin Ave., Goleta, CA 93017 (805) 964-6695

CUSTOMER NEWSLETTER FROM IMSAI

The IMSAIder, a customer newsletter from IMSAI Manufacturing Corporation, is now a bimonthly publication in a new glossy magazine format. Its purpose is: in the words of General Manager Wesley Dehn, "To establish communication with all the people who purchased (IMSAI) equipment, (and) as the capabilities, the usefulness and the performance of our product are improved or expanded, to make that information available." To customers, it is available by subscription at $4.00 per year. Dealers are encouraged to have copies for sale at $1.00 a copy.

Featured in the June issue are articles about the new FORTRAN IV software, a description of what's in the MPU-B Board, and an explanation of a few of the enhancements available in IMDOS, IMSAI's Multi Floppy Disk Operating System. In future issues, look for articles on new products, such as the IMSAI VDP-40 (Video Data Processing System), software applications programs, and many other articles of interest to users of IMSAI computer systems.

The IMSAIder welcomes articles and letters from IMSAI users concerning applications, software enhancements, experience with hardware, user groups, and seminars and events of general interest. The use of articles from the IMSAIder is encouraged, and these may be reprinted upon receipt of written permission from the editor. Contact: Barbara Otto, IMSAI Manufacturing Corporation, 14860 Wicks Blvd., San Leandro, CA 94577 (415) 483-2093.



USC TO LOOK AT PERSONAL COMPUTING

Personal computers may have a more far-reaching effect on our current way of life than television had on life over 25 years ago, believes Jack Nilles of the University of Southern California.

Nilles and a team of four USC professors are studying the effects of the personal computer revolution, supported by a grant from the National Science Foundation.

During the next 18 months, the group will try to answer such questions as: will personal computers raise or lower educational standards? increase or decrease employment? expand the information society or create a new class of underprivileged – the information poor? influence energy consumption? cause economic chaos? protect or invade personal privacy?

The team will focus on two aspects of the uses of personal computers: probable impact areas and public policy alternatives to those impacts. They will first try to outline the probable impact areas, concentrating on the kinds of uses and users of personal computers. Through interviews and questionnaires, the researchers will try to find out who is doing what and who is affected. Secondly, they will perform a detailed analysis of the different impacts and of public policy alternatives to those impacts.

"We do not claim to be able to predict the future, rather, we will provide information so policy makers can anticipate many of the problems that happen to arise," said Nilles.

Contact: Prof. Jack Nilles, University of Southern California, University Park, Los Angeles, CA 90007.

A GOOD CONFERENCE TO *GAMBOL* AT

Recently, Professor William Eadington, Professor of Economics at the University of Nevada, Reno, has put out a call for papers and participants for the Fourth Conference on Gambling.

It will be held at the new MGM-Grand Hotel in Reno, Nevada, December 17 through December 19 of this year. This is a special announcement that is being put out in order to reach members of the various computer clubs, users groups, and readers of the many computer magazines, and the research they have made. With the recent surge in the amount of computers available to the general public, there is a large amount of people available to do computer simulation.

This year, as in previous years, the conference promises to be an exciting event, and any serious student of gambling should plan to attend.

Anyone wishing to present a paper should submit two copies as soon as possible to:

Professor William Eadington
Program Coordinator
Bureau of Business & Economics Research
University of Nevada, Reno
Reno, Nevada 89557

For further information, Bill can be contacted at (702) 784-6850.