People's Computer Company

vol 5 no 4
jan-feb 77

CONTENTS

IN THIS ISSUE

Come and get it! A jam-packed over-sized issue! Robots, Satellites, space games, space colonies, space ships, space out! Futuristic technology and computers are almost inseparable, and we're just on the frontiers of exploration. How about home microcomputers linked via public interest satellite to play a massive game to design and build a starship? Or what about . . . well, read the articles and let us know your reactions!

We've lots of educational articles for all teachers–those in the classroom as well as the home educators: see how to teach kids with conversational programming and Don Inman's front-of-a-series of how-to articles on the 6502-based Data Handler. Several of the educational articles will be helpful both to beginners and also those who teach beginners.

And there's more, more, more: something for everyone: games listings, calculator articles, Tiny languages, etc., etc., etc. There's a gigantic announcement section with lots of cheap software and other good stuff.

Last (well, sort of; in fact it's the first article) but by no means least are the results of the readership survey. The survey was highly successful–a fine response, with lots of good ideas. We hear you! Keep giving us feedback as we keep responding to your input!

THANK YOU!

Dragon Emeritus: **BOB ALBRECHT**
Co-Editors: **PHYLLIS COLE & MARC LE BRUN**
Artists: **ANN MIYA & JOAN LARIMORE**
Circulation: **BEVERLY ROBINSON**
Bookstore: **DAN ROSSET**

Last issue, Larry Press was a non-credited contributing editor . . . our thanks, Lar. And as ever, wild applause for the many without whom we wouldn't have this issue.

RETAINING SUBSCRIBERS:

John R. Lees, Jr.
The Computer Corner, Harriet Shair
John Rible
Bill Godbout Electronics
Mark S. Elgin
James Muysenberg
Daniel Dick
Dexter Fletcher
George Bowie III
Lt. Stan Jurgielwicz
Joe Weisbecker
Bernice Pantell
Larry Press
Frank Otsuka

SUSTAINING SUBSCRIBERS:

Paul, Lori and Tom Calhoun

THANKS!

People's Computer Company is an independent non-profit California corporation, so donations to PCC beyond the cost of subscriptions are tax-deductible.

□ Retaining subscription @ $25
  ($19 tax deductible)
□ Sustaining subscription @ $100+
  ($94+ tax deductible)

PCC SUBSCRIPTIONS
U.S.
- □ $1 single copy
- □ $6/yr. (six issues)
- □ $11/2 yrs. (12 issues)

Foreign Surface Mail
- □ add $4/yr. for Canada
- □ add $5/yr. elsewhere

Foreign AIRMAIL
- □ add $8/yr. for Canada
- □ add $11/yr. for Europe
- □ add $14/yr. elsewhere

Back issues, $1 each; circle those you wish to order; indicate how many copies if more than one of each is requested.

Vol. 1, Nos. 1, 2, 3
Vol. 2, No. 5
Vol. 3, Nos. 1, 4
Vol. 4, Nos. 2, 3, 4, 5, 6
Vol. 5, Nos. 1, 2

Dr. Dobb's Journal, our sister publication, is the reference journal for home computer users. It offers free or low-cost software and "all-meat" contents without display ads. DDJ carries complete systems and applications software, lots of soft and hardware projects, independent consumer information, reprints, directories, hot news, and all sorts of other good stuff. DDJ is published 10 times a year.

U.S. Subscriptions
- □ $12 for 1 year
- □ $21 for 1 year first class/ airmail to anywhere in the U.S.

Foreign Surface Mail
- □ add $4/yr. anywhere outside the U.S.

Foreign AIRMAIL
- □ add $9/yr. to Canada
- □ add $16/yr. to Europe and Pan American
- □ add $20/yr. elsewhere

Back issues, $1.50 each; circle those you wish to order; indicate how many copies if more than one of each is requested.

Vol. 1 Nos. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
□ All of Volume 1 for $13.

In Britain:
PACS
c/o John Remizo
142 Grove Lane
Hale, Altrincham
Cheshire, UK

PCC: £7.50/year
DDJ: £14.00/year

In West Germany:
Pan Atlantic Computer Systems gmbh (PACS)
Frankfurter Strasse 78
D 61 Darmstadt, West Germany

PCC: 30 DM/year
DDJ: 55 DM/year

People's Computer Company is published six times a year by People's Computer Company, 1010 Doyle St., Menlo Park, CA 94025, a tax-exempt, independent non-profit corporation. Donations to People's Computer Company are tax deductible. *Postmaster:* Please send Form 3579 to: Box E, Menlo Park, CA 94025. Return postage guaranteed. Application to mail at second-class postage rates is pending at Menlo Park, CA.

WHO WE ARE

Lots of information resulted from the PCC readership survey sent out with the last issue. Here're the results, from the 191 surveys tabulated so far, except for write-in comments, which will be presented next issue.

Based partly on survey results, we're initiating a variety of changes in PCC. For one, we now have an explicit policy to avoid stuff that's as far out as bicycle repair cartoons and discussions of stapling faults. Have you noticed that San Andreas and the crosswork puzzle are no longer in evedence? The letters from San Andreas fans are a last gasp.

More changes are in the works: we'll discuss them in the next issue. Meanwhile, keep those cards and letters coming – they DO make a difference!

Phyllis Cole

1.

Name, Address

PCC readers come from all over the U. S. and Canada, plus other countries (though slow mails prevented questionaires from other areas from reaching us in time for this profile).

By looking at names, we're almost 97% MALE! OK, so females tend to lack proportional representation in many fields. But most computer-related areas aren't so biased. Hmmmmm . . . . .

2.& 3.

Age, Schooling

	Junior High	High School	Junior College	BA	MA	Phd
under 14	2					
14 – 18	6	5	2			
19 – 30		12	7	33	16	1
31 – 50		8	5	35	29	17
over 50		1	1	3	2	4

4.

Employment

We found 124 of the 191 people who've returned surveys so far *work* with computers. Here's how it breaks down:

work with maxis:	30
minis:	24
micros:	22
more than 1 type:	26
type not specified:	22

Here's what the people working with computers do; many wear several hats:

education:	25
education plus one or more of programmer, engineer, management, technician:	30
programmer	24
programmer/engineer:	15
programmer plus one or more of management, marketing, technician:	4
engineer:	9
engineer plus either or both marketing, technician:	4
management:	5 (plus 18 already listed work in management)
technician:	3
marketing:	1
marketing/technician	1

Lots of people who read PCC don't work with computers. They include:

- teachers at all levels, some work with the gifted, others with the handicapped, some are retired, one does curriculum development
- students, one of whom is not yet 14 but advises his school, a local bank, etc., on micros
- a TV producer and a TV engineer
- musicians/composers
- doctors
- lawyers
- (no Indian Chiefs that I'm aware of, though)
- someone in military service
- a hippy leathersmith
- a telephone repairman, an electronic repairman and journeyman machinist, and an electrician
- magazine editors and a technical writer
- engineers of various sorts
- bankers (yes, more than one–two!)
- an architect
- some people in sales

What a superbly motley crew! Except, of course, that 185 of the 191 are men. . .

5.

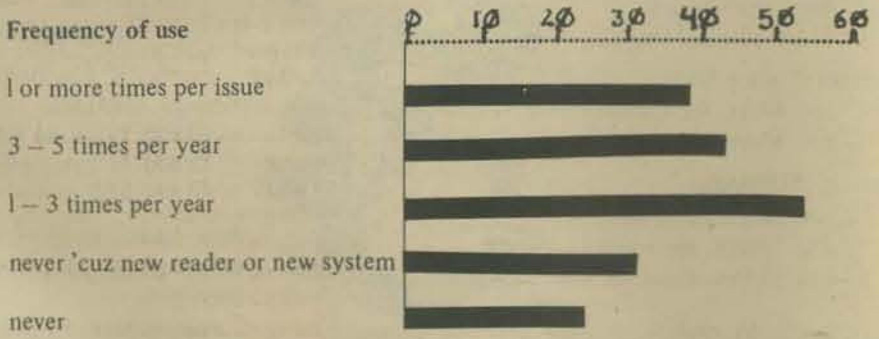
Who pays for your PCC Subscription?

Most people pay for their own, but schools or universities pay for 15, businesses for 5, public agencies (state, county, etc.) for 3 and friends or relatives for 4. A couple of people don't have a subscription and don't want one, though they read PCC regularly – they like an excuse to go to their local computer store regularly.

6.

How often do you use PCC listings?

Several people mentioned they used the listings to get ideas and/or learn about programming.

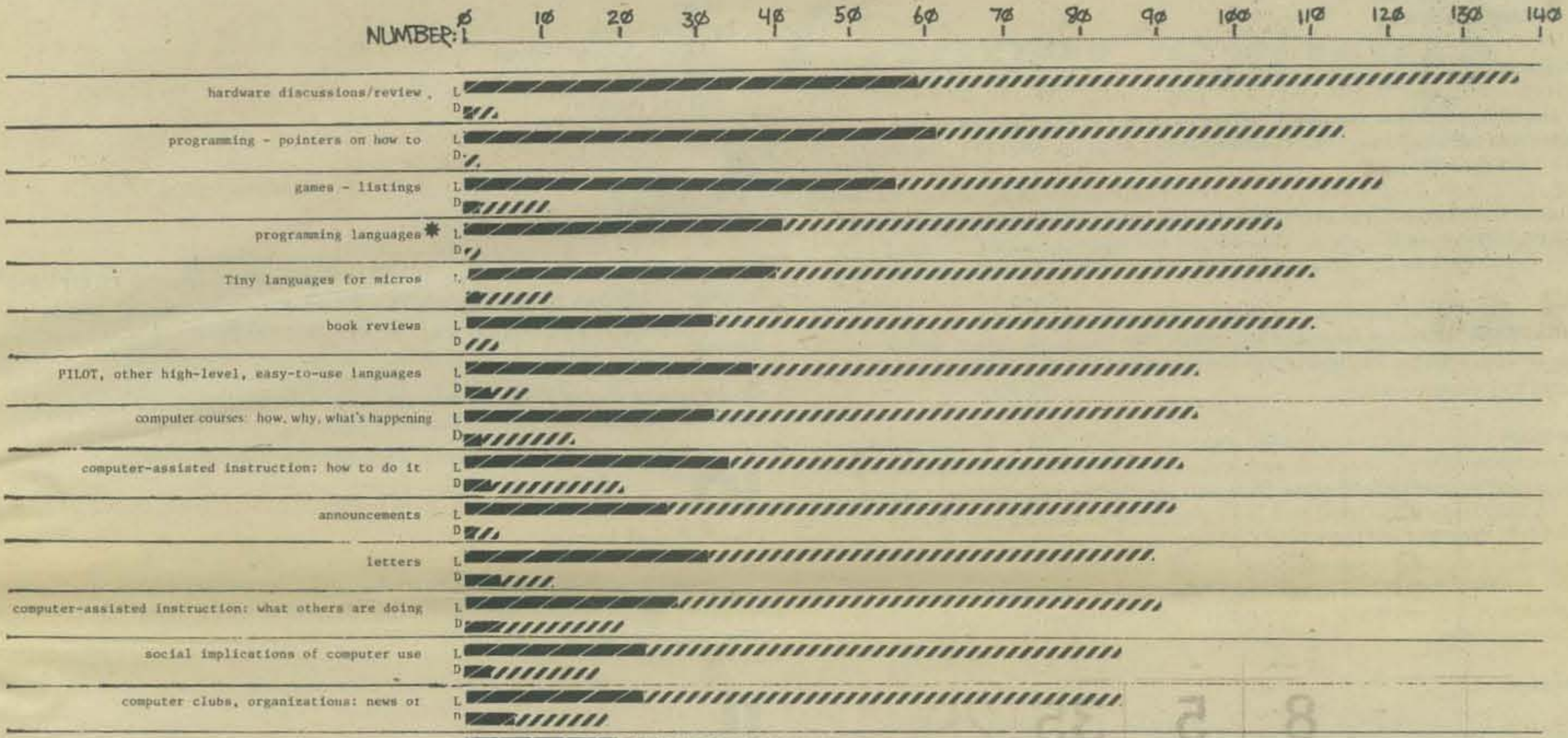


7.

Reactions to PCC content

Whew! What a tabulation job! Some topics, such as Don Quixote Starship, may have gotten short shrift since they were unfamiliar to readers. Others, such as Sprocket Man got a strong vote even if they *were* unfamiliar.

OK, we hear you: no more crossword puzzle, San Andreas, or Sprocket Man. And we'll keep any philosophy that appears relevant to areas of expressed interest.



* Most often mentioned were
BASIC: 47
FORTRAN: 12
APL: 10
assembly: 8
PL/1: 4
TRAC: 4

MORE→

WHO WE ARE

8.

Interest in Computers

Three main areas of interest in computers were listed: programming, hardware and education. Most readers are interested in more than one area.

programming/hardware:	56
programming	47
education/programming/hardware:	34
education:	26
education/programming:	14
education/hardware	1

Programming

Of the 191 responders to the survey, 145 are interested in programming micros. Most often mentioned were 8080 based systems (45) , the Z80 (13) , the 6800 (11) and the 6502 (6).

Some of the 63 people who expressed interest in programming minis also expressed interest in micros and/or maxis. Most interest was expressed in DEC systems (16), with 4 votes each for Interdata and HP, and 2 votes each for Data General and Wang.

About 30 people expressed interest in programming maxis. In addition, topics such as BASIC and other languages(particularly high level), calculators, digital logic devices, artificial intelligence, operating systems, and timesharing were mentioned by 1 or more readers.

Hardware

Micro hardware interests 88 of the 191 readers who replied. The 8080 was mentioned by 24, the Z80 by 9, the 6800 by 7 and the 6502 by 3. Mini hardware was of interest to 10 readers and large system hardware to 1.

Other hardware interests included other specific systems than those named, displays, line printers, interface, terminals for time-shared systems and design of micro systems.

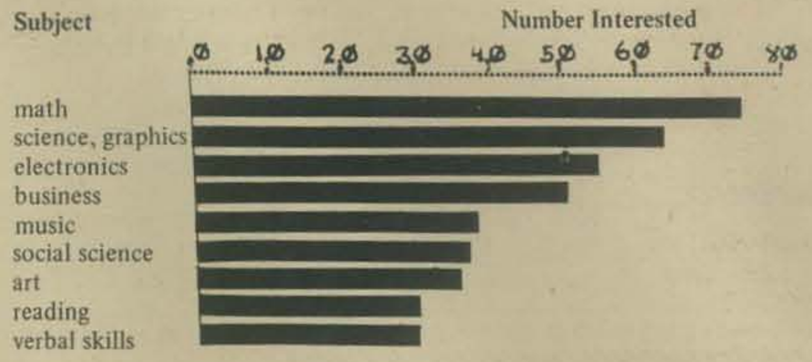
Education

Readers include 59 teachers, who teach at the following levels (often more than one):

K – 6:	8
7 – 8:	12
9 – 12:	25
college/ university:	23

Write-in categories included everything from preschool to postdoc and "my kids."

There are 59 teachers but 123 people who want educational topics available on their systems. Many of these people have micros at home and want all sorts of software, including educational. Here're the subjects of most interest:



Write-in votes included games (11), artificial intelligence (6), home applications such as record-keeping and remote control functions (5). Space science graphics and languages each received 2 votes.

Here're the results of the forms-of-computer-in-education question:

form	votes
simulations	77
games	73
problem solving	70
tutorial computer-assisted instruction	57
drill and practice	47

9.

PILOT—like languages

We find 85 readers in the survey are interested in getting an interactive PILOT —like language; in addition, 6 people already have PILOT or a similar language, and 11 people wanted more information. A total of 58 were not interested in such a language.

On the type of display for such a language, there were 56 votes for CRTs of some kind or other and 24 for others (Yes, some people voted for both).

10.

Ownership of Micros

Lots of people own micros: 103 of 191! No wonder there's such interest in hardware. People mostly owned 8080 – based systems (69), 6800s (18) Z80s (12) and 6502s (10). The figures make it look as if quite a few people own more than one!

11.

Access to Computers

PCC readers have access to numerous systems from micros to maxis of every description.

12.

What do you want in PCC?

Next issue we'll present the answers to this question in some less-than-form (the state it's in at this moment, as we rush to press).

Your response has been great – – Thanks! ■

My Computer Likes Me Even More When We Have Conversations

or

ANOTHER WAY OF INTRODUCING CHILDREN TO PROGRAMMING

by Joanne Koltnow Verplank

This is a short disclaimer for the language I'm writing about: Our computer runs Basic Plus, a version of BASIC with string variables of (seemingly) unlimited length. We don't have PILOT. People who know PILOT tell me that what I'm doing would be much easier for the children if we used that language. Maybe someday we will.

It's important for people to have control over the things they use. Children are using computers at the Center and, as much as possible, I'd like to give them some control over the machine. I introduce children to computer programming with this in mind.

The children who are in classes where they're learning programming are all ones who have already played games on the computer for several sessions. Programming is different from game-playing, and initially, takes more work and may not be quite so exciting. While I try to minimize the problems and maximize the excitement, it helps if they're not being tantalized by a lot of exciting looking games just at the outset.

I use a language-oriented rather than number-oriented approach, and I use string input extensively. For young children especially, their expertise with language is far greater than with numbers. A computer can do marvelous machinations with numbers, but if the children can't relate to them, we're left with rather boring addition and subtraction. On the other hand, the children have been thinking and talking most of their lives, and their language capabilities are well developed. A computer language with string fuctions of unlimited length is just what they need.

I have been using this language-oriented approach with children whose ages range from seven to fourteen, in groups of about ten. My specific techniques vary with the ages and sizes of the groups, and so does the speed at which I present the material. I wouldn't recommend starting to teach programming to children as young as seven. Several seven-year-olds were in a class of sevens to nines. Those who were successful were fortunate in that I wouldn't have taught them at all if the whole class were composed of such young children. On the other hand, it was so difficult to keep their interest that I don't think it's really worth the effort. In the case of these beginners, there was always a teacher from the school who took an active part, so children who couldn't read or spell had help.

```
HOMER EATS JELLY FISH
              ************
KATHI R. LIKES WHEN
LEONARD B. PLAYS CLARINET
              ************
LEONARD B. LIKES WHEN
KATHI R. PLAYS DRUMS?
              ************
WIERD ISN'T IT
```

I start by gathering the group around our large display terminal and telling them that I'm going to show them how *they* can tell the computer what to do. As I type PRINT I tell them that we can get the computer to print whatever we tell it. I type quotation marks, and write a sentence. It usually looks like this:

```
PRINT "HELLO THERE"
```

Then I press return and it says:

```
HELLO THERE
READY
```

We practice this pattern with different sentences. Each time I type, I remind them (or they say) that we have to type PRINT and enclose the line in quotation marks. Before I press return, they tell me what the computer will print. At this point, what they have to type and what the computer will do is well fixed in their minds. If they are young children, they stop at this point and practice on their own. Sitting at the terminals in pairs or groups of threes, they have conversations with each other, using the computer in immediate mode as an intermediary.

```
PRINT "DO YOU LIKE SNOW?"
DO YOU LIKE SNOW?
READY
PRINT "YES OF CORSE I LIKE SNOW."
YES OF CORSE I LIKE SNOW.
READY
```

The next step is writing a program. I bring them back to the display terminal and we discuss the fact that, using this method, the computer will only print one line at a time before it says READY. Then, the process has to start again, and this interrupts conversations. At this point, I tell them that the computer can do something really great, that it can store what they type in, and not print anything at all until they tell it to. We arrange this by writing line numbers in front of the print statements and the line numbers are a signal for the computer. When we put in the line numbers, the computer doesn't do any of the program until we type RUN. Line numbers also help the computer keep the program in order and let us insert lines throughout the program, even if we think of them later on. I mention that people usually count by tens when they write line numbers, that they don't have to, but it's handy.

Now I type in some variation of the following program:

```
10 PRINT "HELLO THERE"
20 PRINT "HOW ARE YOU?"
30 PRINT "TODAY IS SUNNY, ISN'T IT?"
100 END
```

As I type each line, I ask them what the computer will do with it, just as a review. They're usually surprised that the computer in fact doesn't print the line the first time I press return. As I type in the last line, I tell them that the computer has to know when to stop, so that the largest number in the program has to have END after it. Then I type RUN, and we watch the computer run the program. This time the children take an active part, giving me sentences and line numbers, reminding me to write PRINT and to put in the quotation marks.

```
THIS IS A NONSENSE STORY.
*************************

KATHI R. IS SHORTER THAN LEONARD.
JODI IS TALLER THAN KATHI.
ARVIND DOESN'T TALK.
```

```
MISS HAGERTHY PLAYS BASKET BALL.
KATHY D PLAYS TUBA.
MARIANNE RIDES AN ELEPHANT.
LLOYD BROWN WEARS SHORTS.
MR. BJONERUD DOESN'T LIKE CHESS.
******************************
```

After we have written and run a few more short programs, I show them that they can type LIST instead of RUN to get the computer to print out just what they've been typing in. We compare the listing and the run of whatever program we're doing, and ask them to tell me the difference between the two results. It's not a problem, and they're on their way to being in control.

The last thing, and this is something I often save for later if it seems they've had enough at one sitting, is showing them that they can insert lines into a program. We take the current program and I say that I want to write the line "My name is Joanne" between the first and second lines. They know that I have to use PRINT and enclose the sentence in quotation marks. I ask what line number I should use. They come up with a number between 10 and 20 – usually 15. I say 15 will be fine, and ask them what other number I might have given it (while I'm typing in the line.) They tell me other possible numbers. I quiz them about numbers that would put the line some place else. They usually all answer appropriately. Then I say what we've been thinking all along – the computer's quite wonderful and that it will know just what to do with line 15 even though we put it in last. We type LIST and sure enough, there's the program with the new addition. Then we type RUN, and they see their prediction come true. Everything we've done makes sense, and they're getting a lot of reinforcement from the computer for their accurate prediction of its logical behavior. That *has* to increase their confidence and make them anxious to learn more.

At this point, I send them to the terminals to write some short stories. This is the only time that I consider allowing them to compose at the terminals – it's not a good habit to get into. To avoid this, even with a group of 10 children, compose several stories as a group, and have them work together to type and run them at the terminals. Another possibility is stopping the lesson and sending them off to write a three line story (with line numbers and PRINT statements) to type in the next time.

```
HELLO MARK
HELLO MR. SMITH
HELLO MATSU
END

******************************

BYE TRAVIS
BYE RICH
BYE DAVID
END
```

The next step, if they haven't picked it up already, is to show them that typing a new line, using a line number already in the program, replaces the old line with the new one. Related, and also easy to understand, are (1) "erasing a line" by typing its line number and then pressing return, and (2) printing a blank line by typing the line number, PRINT, and then pressing return.

None of this is too hard for most children to get if it's presented at the right speed and with sufficient examples. They need practice, however, and this is where their poor typing skills slow them down. A visiting teacher's suggestion for typing practice led to my duplicating a drawing of the keyboard for each student. They had a copy to practice with at home and to use for "typing in" their stories. The practice keyboard helps familiarize them with the real keyboard. Unskilled as many of us may be at typing, we're still years beyond most youngsters in our experience with the layout of a typewriter keyboard.

```
KATHI LIKES LEONARD
*****       *******
LEONARD LIKES KATHI
*******       *****
THEY BOTH GOT MARRIED AND LIVED HAPPILY EVER AFTER
              *******     ***** ******* **** *****
```

Some of the class will get involved with writing stories and others will want to go on with the next step. Try to keep this in hand and organize to the degree that their needs and your sanity are being served.

GOTO is an easy and dramatic thing to teach next. Write a short program with the class:

```
10 PRINT "HELLO THERE"
20 PRINT "MY NAME IS MARC"
100 END
```

Ask them what it will do when you RUN it. Then add:

```
30 PRINT
40 PRINT "HI"
```

Ask them again, making sure that they mention the inserted blank line. Now add:

```
50 GOTO 10
```

and talk them through it, line by line. Ask them if the computer will ever get to the end of the program. Talk through it until they're pretty sure of what they imagine will happen, then run it. Even though they're prepared, it's neat to see the computer printing line after line of the same thing. Quickly teach then about control C or the break key. Now, assign the writing of a program using GOTO. Also, encourage them to make their previously written programs fancier, using spaces and lines of *s. This will give them a little more typing practice and will also give them more control over what their output looks like. Sometimes I do several examples of an assignment with the group before they go off on their own. Remember, this isn't a test for them. They should be able to do everything you assign.

```
LAST SUMMER I WENT ON A SAFARI AND BAGGED THREE
BEAR

I HAVE MORE MONEY THAN ANYONE ELSE
JOES LIAR CLUB U.S.A.
```

Depending on the circumstances (young children, need for more action) I sometimes skip the whole GOTO part and go right to INPUT.

I tell them that I'm going to teach them a new word that will let us ask questions and then *save the answers* so we can use them later. I remind them about STORY, where your favorite color turned out to be the color of an animal, and what you were afraid of almost ate you until the person you loved came and scared it away.

One way of teaching INPUT is by writing a group story. Another way is to write a program so that the computer will have a conversation with whomever happens along. Choose the one your group will get the most out of – or try both.

STORY APPROACH

The children are all familiar with STORY and some of its variations. I tell them that we're going to write our own story game, and that we need some questions to start with. (You could also start with the text of the story and work back to the questions.) I do all of the following on a blackboard rather than at the terminal. This shows them that we plan the program before we type it in, and it also allows us time to disucss and change what we write before we make final decisions. Here are some questions that we came up with for the program that finally became TREES:

What's your name?
How old are you?
What's your favorite food?
What's your pet's name?
What's your favorite color?

I discourage them from writing questions that have yes or no answers by pointing out that they're going to use the answers in the story and that it will be hard to deal with a bunch of yesses and nos. They usually understand. (That's another reason to have them thoroughly familiar with the story games before starting to program.) Next we start writing a story, using the answers to the questions as a guide. Whenever we come to one of the to-be-string-variables, I make a point of saying "Here's where the computer will put in whatever the person said for a name . . . . . .Here's where it will put the age the person said . . . . . ." So the kids are aware, right from the start, of the variables, although I never have to say that word. There hasn't been much trouble with their understanding this idea, either. Again, it helps that they've played the story programs so much, and that they're well aware that the computer will just insert, in the story part, whatever you give it as an answer to the question.

```
TREES

WHAT'S YOUR NAME?
? ERIKA
HOW OLD ARE YOU?
? 9
WHAT'S YOUR FAVORITE FOOD?
? STEAK
WHAT'S A PET'S NAME?
? WOLFIE
WHAT'S YOUR FAVORITE COLOR?
? ORANGE
ONCE UPON A TIME ERIKA TOOK WOLFIE
FOR A WALK.  THEY MET A WIERD ORANGE
TREE THAT HAD 9 LEAVES.  THEN THEY
TOOK THE TREE HOME WITH THEM TO EAT
STEAK.
```

Sometimes we have to change or discard questions as we write the story. I talk as I write down what they say, and a conversation with the group goes something like this:

Me: How shall we start? Do you want to start "Once upon a time?"
Them: Once there was a little dog named . . .
Me: OK. Whatever the person types for the name will go here. Let's call it N$ to remind us that it's the name.

That's usually all I have to do to use the string variable notation. I suggest and they accept. Remember, we're writing a group story, so they don't all have to remember all of this – just follow it. We go on making up the story, while I point out the variable notation we insert to remind us that we're going to use whatever the person types in. At some point in this progression I may ask them for suggestions of letters to use as the variable symbols.

As we create the story, we make final decisions on the questions we wrote first. Some of them have answers that are in the wrong form, or would be complicated to use, and we try to fix them. (Some of the fix-up doesn't occur until we finish the program and start running it.)

Now it's time to program the new story. I just start writing PRINT statements and INPUT lines, and talk as I do it.

```
10 PRINT "WHAT IS YOUR NAME?"
20 INPUT N$
```

Now we're going to make a place for the computer to store the name. We say INPUT and call it N$ to remind us that we're putting the name in there. We could use any letter we want, as long as we remember that that's the place we're storing the name. Now, a line number and PRINT for the next question:

```
30 PRINT "WHAT DO YOU LIKE TO DO AT THE BEACH?"
40 INPUT B$
```

We finish the questions in this manner, and write the story with print statements, inserting the appropriate variable labels from the questions, and following the text we just created. There are always little finishing touches to do after the first run. Usually, the answer to the question will run together with the words just before and after it. The children have to learn to put in spaces before the quotation marks, but it's easier if they see the need than if it's just one more thing I have to tell them.

```
WHAT IS YOUR NAME?
? JOANNE
WHAT DO YOU LIKE TO DO AT THE BEACH?
? SUNBATHE
WHAT ARE YOU AFRAID OF AT THE BEACH IN THE WATER?
? CRABS BITING MY FEET
WHAT DO YOU LIKE TO DO AT THE BEACH?
PLEASE USE ING WORD
? SWIMMING
HOW DO YOU RUN?
? FAST

ONCE UPON A TIME JOANNE WENT TO THE BEACH.
SHE WAS SWIMMING UNTIL THIS BIG CRABS BITING MY
FEET CAM UP   SHE SCREAMED SHE WAS SWIMMING
AS FAST AS SHE COULD BACK TO SHORE
```

It takes a while to do a story like this, and sometimes a group as large as ten gets restless. Avoid this by making a short story so everything moves along quickly, being careful to involve everyone, or begin it with the whole group and finish it with a few diehards as the others drift off to start their own.

The other approach, good with older children, is to show them briefly how the string input works in individual examples, then send them off to write their own stories or conversations.

CONVERSATION APPROACH

```
10 PRINT "WHAT'S YOUR NAME?"
20 INPUT N$
30 PRINT "HELLO"; N$
```

As I type this, I tell them to imagine that the computer has a lot of spaces, like boxes, where it can store things. We store something in one of the boxes, we name the box, so we know where we put the information. When we want to use the stored information, we refer to the name of the box. We can put more than one word in the box at the same time, but if we want to put something else in the same box, later, the computer dumps out all the information it had previously stored there.

```
40 PRINT "WHO'S YOUR BEST FRIEND?"
50 INPUT F$
60 PRINT "SO";N$;" AND ";F$;" ARE GOOD FRIENDS
100 END
```

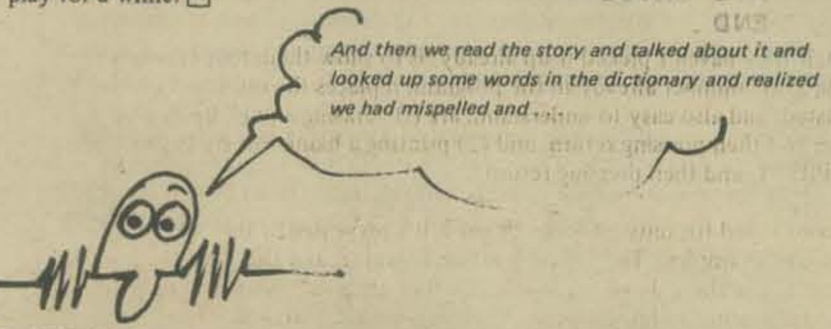
We can use lots of boxes for different bits of information. Whatever we type in comes back to us when we refer to it later.

This is a good time to do an exercise using GOTO.

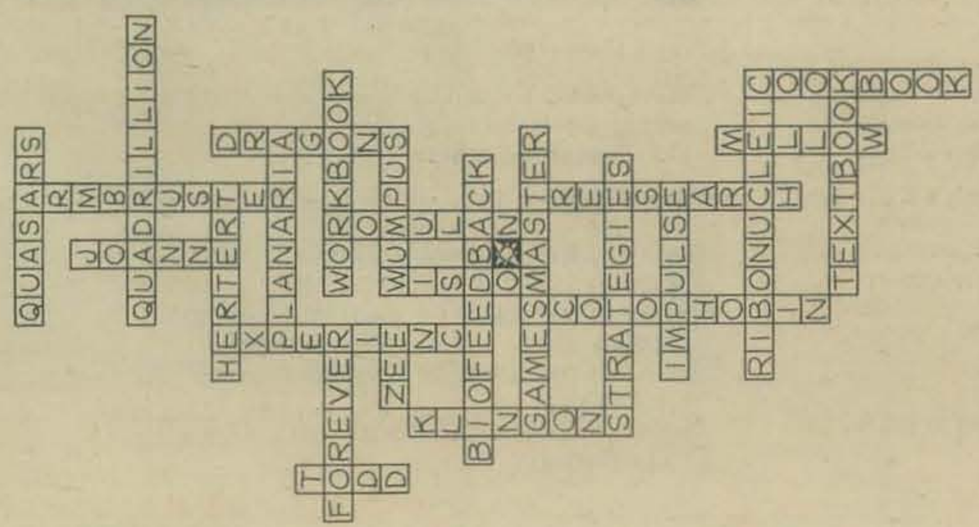
```
10 PRINT "HELLO.  WHAT'S YOUR NAME?"
20 INPUT N$
30 PRINT "HI ";N$
40 GOTO 10
100 END
```

Talk them through it, and then run it. The point of this is that one variable can be anything they type in for it, but only one thing at a time. So the computer "forgets" the first when it gets the second.

This should get you through a few weeks. From here, some children will want to write games, others will be entranced with the possibilities of PRINT and INPUT and will go off into writing conversations, stories, and making designs. Let them play for a while. □


And then we read the story and talked about it and looked up some words in the dictionary and realized we had mispelled and . . .

Answer To Crossword (from last issue)

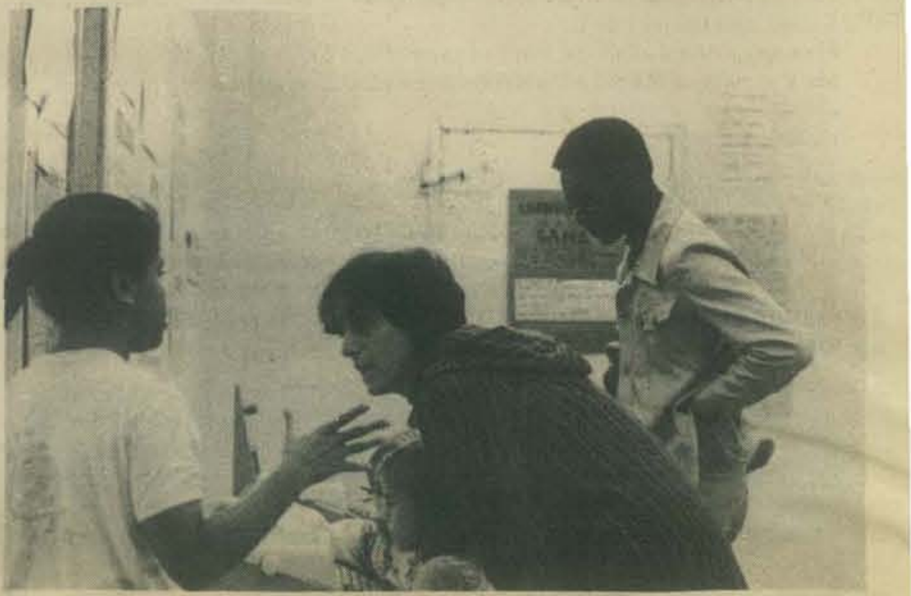


Answer	Page
1. Klingons ● Romulans	2
2. impulse	2
3. mellow	2
4. Joanne	10
5. strategies	10
6. Wumpus	11
7. gamesmaster	20
8. dragons	20
9. forever	20
10. wisdom	21
11. experience	21
12. lee ● todd	23
13. quasars	24
14. quadrillion	24
15. ribonucleic	25
16. scotophobin	26
17. planaria	26
18. Armbruster	30
19. treesearch	30
20. Hertert	44
21. cookbook	44
22. workbook	44
23. bugbook	44
24. textbook	44
25. biofeedback	46

A Day In The Life of Community Computer Center

by Dean Daily

Caldor and friends.



"But what if I . . ."

"We did it!"

Wednesday, December 8, 1976
10 A.M.

Joanne Verplank, the director of the Community Computer Center, usually arrives at about 9 or 10 a.m. on weekday mornings to get ready for one or a series of field trips to the Center.

Field trips by outside groups are the most important activity of the Community Computer Center. Groups of San Francisco Bay area grade school students take time out from their classes to visit CCC where they learn how they can use the computer for their enjoyment and enlightenment.

A typical field trip involves from 15 to 30 students of varying ages, and usually lasts 1 to 2 hours. What Joanne teaches a particular group depends on both the grade level of the students and their experience with the computer. First graders who can barely read can play a simple number guessing game. At the other extreme, high school students can learn programming or work with computer simulations.

Today Joanne has arrived early to tend to the Center's business. Right now she is talking with Howie Franklin about a book, a reprint of **What To Do After You Hit Return or PCC's First Book of Computer Games**, on which CCC and PCC are working together.

Howie is the most dedicated of the several CCC volunteers. There are only two full time paid staff members — Joanne, and Doug Faunt who is System Manager, programmer, technician and part-time teacher. For this entire week, Doug is in Las Vegas attending a computer conference. CCC also pays Mark Freedman, a Palo Alto High School student, to work as a programmer and supervise groups of independent computer users on nights and Saturdays.

Joanne and Howie are anticipating the arrival of Albert Bradley to assist in the preparation of the games book. They also talk about how to get foundation grants, one of the several revenue projects that the cash-starved CCC is considering. In particular, they are seeking foundation help to place computer terminals in public places to tie in with the CCC computer.

11 A. M.

Today's schedule is not usual: it's part of a series of classes with students from Peninsula School in Menlo Park. Instead of a one-shot "turn on to computers" field trip these students are getting to explore the available games in some depth, and then go on to programming.

The first group from Peninsula School, about a dozen kids, a teacher, one parent, and a dog have arrived. This is the eighth visit to the center for most of this group. Having already written their programs, the students are now entering them into the computer. What does a 9 year old's programming look like? Here is one example:

```
10 PRINT "I THINK YOUR'E CUTE"
20 PRINT "CALDOR IS CUTE"
30 PRINT "DOU YOU LIKE THIS"
40 PRINT "I HOPE YOU LIKE THIS"
100 END
```

Indifferent to flattery, Caldor lolled on the floor, not even bothering to wag his tail in appreciation of the compliment.

These children are learning how to input and output strings of literal data. They write programs that require the person using them to answer questions by inputting names and other facts.

Here is what one 9 year old's program looks like as it is run:

```
WHAT IS THE NAME OF A FRIEND?
? BOB
WHAT IS THE NAME OF ANOTHER FRIEND?
? SUSAN
WHAT IS THE NAME OF A THIRD FRIEND?
? ROCHELLE
WHAT KIND OF CANDY DOU YOU LIKE?
? ALMOND JOY
```

The computer then assembles the story:

```
THE CANDY STORE
ONCE UPON A TIME BOB, SUSAN AND ROCHELLE
WENT TO THE CANDY STORE
BOB SAID I WANT TO BUY THIS
SUSAN SAID WHAT IS IT?
BOB SAID IT'S A ALMOND JOY
SUSAN SAID OH!
ROCHELLE SAID WOW I WANT ONE!
SUSAN SAID DON'T BE SO WOW-EE!
```

Not all of the students concentrate exclusively on programming. Some of them play computer games or some of the Center's board games or a Pong game.

Noon

The first group has left. Howie is using a video terminal to program one of the games for the Center's use. CCC is perhaps best known for its collection of computerized games and simulations which are used in teaching students about computers. Joanne has described educational uses of games in previous issues of PCC.

However, behind all of the fun and games lies a lot of work. Doug, Howie and Mark spend much time programming new games, or making old ones work.

12:30 P.M.

Another Peninsula School group has arrived, and is being taught some games. They are all clustered around a small video terminal; Joanne is explaining its use. Field trips are often a combination of lecture/discussions and individual practice at the terminal. Each of the six terminals available at the Center can accomodate as many as three students. With large groups, some students work at the terminals while others are busy at different activities.

Joanne and a group discuss this program.

Howie and Albert looking through tapes for the games book. Joanne is in the background.

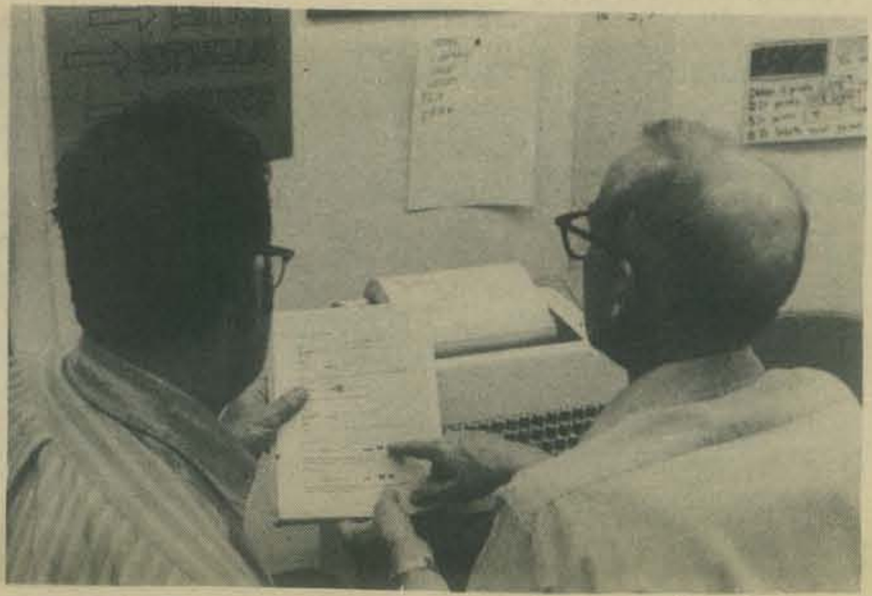
1:30 P.M.

As the field trips continue, Albert arrives to discuss some work he is doing on a revision of the PCC games book.

3 P.M.

The last of the field trips has been completed, yet the Center is not quiet. Several adults have come in to rent time on the computer.

The Community Computer Center is sometimes thought of as being primarily for children. While the majority of the visitors to the Center are youngsters, the Center has activities that are of interest to adults as well. Many of the games on the computer are sophisticated enough to challenge adult players. In addition to the games, CCC has software and hardware resources for homebrew computer users. Those interested in learning about computers can take classes at the Center. For novices there are classes that give a general introduction to the computer; for knowledgeable computer users there are more advanced classes in hardware and software design. Anybody can rent time on the computer for $1.50 an hour.



These people came in to ask about computers and sat down to learn the program.

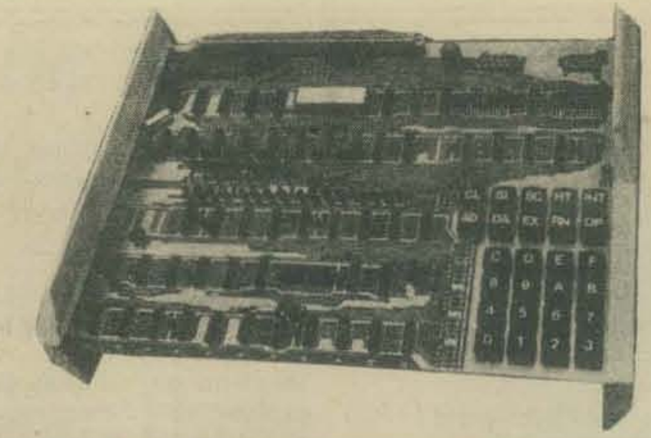
5 P.M.

Except for the staff itself, the Center is now empty. However, even when the Center is quiet, there is a lot of work to do. Phone calls and visitors come in at all hours of the day, letters must be written and answered, and projects discussed and developed. Usually, the Center closes down at 5 p.m., with Joanne leaving for the day. At 7 p.m. on weeknights, the Center reopens, with either Doug, Mark or one of the volunteers in charge. At this time the computer becomes available to individuals or groups for their own education or amusement. Sometimes Joanne or another staff member will stay on to teach one of the classes that the Center sponsors at night. ✱

"What'll happen next?"

THE DATA HANDLER USERS MANUAL

BY DON INMAN



The DATA HANDLER is a complete microcomputer system on a single P. C. board combining the Mos Technology 6502 microprocessor with the latest state of the art technology. Superior design enables the DATA HANDLER to operate at very high speeds as a stand along microcomputer or dedicated controller for even such high speed devices as disk peripherals. External TTY or terminals are not needed since the DATA HANDLER contains 26 keyboard switches for full function hardware front panel control. Personal expandability of the system is achieved by using the Altair/IMSAI peripherals. Anyone of which may be used directly on the DATA HANDLER P. C. board or further system expansion may be accomplished with the use of the Altair mother board.

The Data Handler, a one-board 6502 microcomputer by Western Data Systems, is being used to train teachers in the San Jose schools (see PCC Vol. 5 No. 3). "The Data Handler User's Manual," a summary of course material, will be serialized in PCC beginning with this issue. While written specifically for the Data Handler, the material could easily be modified to fit other 6502 computers such as Kim - l, EBKA, etc.

The first two San Jose sessions were devoted to constructing the computer from kit form. The material presented here picks up at that point (i.e. assuming the computer has been constructed and is operable.) The complete first semester manual (Volume I) will be available from the PCC bookstore in the near future.

INTRODUCTION

This User's Manual is designed to serve as both a self-teaching guide and as an outline for a course at the beginning level of computer science. While it deals specifically with the Data Handler, it can easily be adapted to other microcomputers using the MOS Technology 6502 microprocessor.

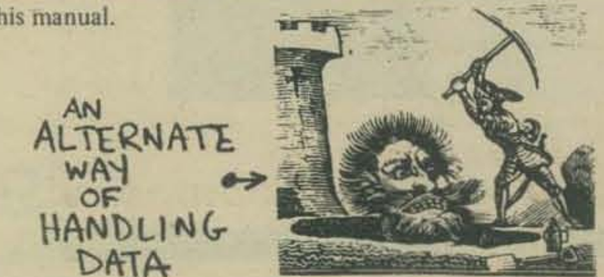
The Data Handler has been used along with the material in this manual in a teacher training course for the San Jose School District. This particular product was chosen because of its versatility, flexibility and low cost.

The San Jose teachers constructed the microcomputer from kit form during two two-hour class sessions plus whatever additional amount of time was needed outside of class. The operation of the Data Handler, the 6502 instruction set, and an introduction to machine language and assembly language were covered in seven additional class meetings.

Nine additional class meetings are planned for a second semester advanced course. At the completion of the year, the Data Handlers will be distributed to individual high schools in San Jose for further familiarization by the teachers. They will eventually be used to teach the high school students the fundamentals of computer operation.

Mike Indihar of Western Data Systems has been most gracious in helping to debug the problems encountered in the construction of the kits. Many of the teachers (approximately one-third of the class) had no previous experience with printed circuit board construction. The single-board Data Handler is quite tightly packed with components and traces, and beginning constructors did have problems.

The Programming Manual and the Hardware Manual of MOS Technology (*MOS 6500 Microcomputer Family Programming Manual;* MOS Technology, 950 Rittenhouse Road, Norristown, PA 19401; Jan. 1976, 2nd Ed.) were used as reference texts in the course and also for this manual.



The Data Handler is a complete microcomputer system on a single printed circuit board designed around the MOS Technology 6502 microprocessor. The system contains one thousand words (1K bytes) of random access memory (RAM), one eight bit parallel output port with clearable data flag. The Data Handler contains a 26 keyboard switch hardware controlled front panel which will load data, examine data, perform single cycle and single instruction, initialize the system, run and halt the system all in hexidecimal (base 16) format.

It contains a complete on-board discrete L.E.D. hexidecimal address and data display which displays the sixteen address bus lines and the eight data bus lines.

Also on the Data Handler printed circuit board are three discrete L.E.D.'s which indicate whether the Data Handler front panel is in the address or data entry mode and whether it is halted or running.

The left rear portion of the board contains two rows of fifty pads to facilitate the installation of a single 100-pin Altair/IMSAI type, edge connector. These pads contain the tri-state buffered data, address, memory, I/O(input/output) signals, and timing signals used by the 8800 Altair / IMSAI type peripheral boards. These bus lines are pin for pin compatible, and drive/receiver matched with the 8800 Altair/IMSAI microcomputer peripherals. These peripheral boards are the least expensive, most widely available, and offer the greatest variety in microcomputing peripherals on the face of the earth.

The right rear portion of the printed circuit board contains one 8 bit patching parallel input port with interrupt strobe, and one 8 bit parallel output port with a data flag. Additional I/O devices may be connected through these ports.

SPECIFICATIONS

- Uses the MOS Technology 6502 microprocessor
- Directly addresses 65K bytes of memory
- Contains 1K bytes of 500ns RAM on the board (address FC00 - FFFF)
- Slow down circuitry for slow memories
- Direct memory access (DMA) circuitry and DMA acknowledge control lines
- Dual interrupt lines – one maskable and one non-maskable
- On board dual I/O ports
- Variable speed R/C clock, or optional crystal may be used for frequency stability to 2 MHZ
- 8800 Altair/IMSAI identical tri-state address and data bus
- Single +5 voltsupply @ 1.8 amps (User provided)
- Large on board ground plane for noise suppression at high operating frequencies
- Discrete L.E.D.'s for address and data display
- Full function hardware controlled front panel
- Keyboard type data and control switches in hexadecimal format

The DATA HANDLER Bare Bones Kit which includes the DATA HANDLER PC board, PC board stand, 26 keyboard switches, and a complete documentation package is being offered at a price of $89.95.

The complete kit is priced at $179.95. This includes the DATA HANDLER PC board, PC board stand, 26 keyboard switches, the complete set of IC's, 1 6502 MOS Technology microprocessor, sockets, LED's, resistors, capacitors, 500 ns memory, and a complete documentation package.

COMPUTER NOTATION

Before we turn to the use of the Data Handler, a review (or introduction) to number systems is in order. Most of us have used the decimal number system (with its base of ten) so much that our familiarity with other systems is slight.

The computer is an electronic device using switch-type signals. A signal is either present, or it is not present. Therefore, it is a user of binary symbolism. Whereas the decimal system has ten symbols (0, 1, 2, 3, 4, 5, 6, 7, 8, 9), the binary system has only two (0 and 1). We may assign the 1 to correspond to a switch closed, a light on, or a signal high. The 0 then corresponds to a switch open, a light off, or a signal low.

We refer to a 0 or 1 in a binary symbolism as a bit. It provides one bit of data or information. The Data Handler uses an eight-bit word for communication. LED's on the printed circuit board are used to show what is going on during programming and operation. The row of data lights show an eight-bit word, XXXX XXXX, where each X represents a zero (light off) or a one (light on). Our eight-bit word will be referred to as a byte. Eight bits = one byte. Binary numbers become very long if we try to represent a large decimal number. For ease and simplicity in reading data into the machine and in interpreting results, the Data Handler (as most other computers) uses a hexadecimal representation of the machine's binary numbers.

The chart, which follows, shows the relationship between decimal, binary, and hexadecimal numbers. The Data Handler's keyboard is arranged in hexadecimal format (using the symbols: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F). The address and data LED's are arranged in groups of four so that they may be quickly converted to HEX notation. Therefore, begin to "THINK HEX".

DECIMAL	BINARY	HEXADECIMAL
0	0000 0000	0
1	0000 0001	1
2	0000 0010	2
3	0000 0011	3
4	0000 0100	4
5	0000 0101	5
6	0000 0110	6
7	0000 0111	7
8	0000 1000	8
9	0000 1001	9
10	0000 1010	A
11	0000 1011	B
12	0000 1100	C
13	0000 1101	D
14	0000 1110	E
15	0000 1111	F
16	0001 0000	10
17	0001 0001	11
18	0001 0010	12
19	0001 0011	13
20	0001 0100	14

etc.

SESSION III – PRELIMINARY CHECKOUT

(Sessions I and II were kit construction)

L. E. D. DATA AND ADDRESS LIGHTS

Four binary bits (xxxx) may be used to represent one hexadecimal symbol (0 through F). As seen from the chart on the previous page, F (the highest valued hex symbol) is represented by 1111 in four binary bits. If one is added to F, we get a carry into the next place value (F+1=10) which means we have one 16. Since four bits can be represented with one hex symbol, a full 8-bit byte can be represented by two hex symbols. When using the Data Handler, we will code our instructions and data in hexadecimal symbols using two symbols per byte. Not only does the Data Handler use the hex format, but the MOS Technology Programming Manual presents the 6502 instruction set in hex notation also.

As an example: If our eight data lights on the computer were lighted in this manner

○●○● ●●●●

we would evaluate the number as 5F. The leftmost group of four lights show the 5, and the rightmost group of lights show the F.

A memory location shown on the address lights might look like

●●●● ●●○○ ○●●○ ●○○●

This configuration would be evaluated as memory location FC69. The leftmost Group showing the F, the next group show the C, the next group show the 6, and the rightmost group show the 9.

USING THE DATA HANDLER TO COUNT IN HEX

Now let's put the Data Handler to use by demonstrating the counting process to practice hexadecimal conversion. With the power on, perform these steps:

1. Press the CLEAR KEY. CL
2. Now press the ADDRESS KEY. AD
3. Press the EXAMINE KEY next. EX

The result of the last step should have turned on the light in the rightmost position of the address lights (this one is called bit zero). The lights should look like this:

○○○○ ○○○○ ○○○○ ○○○● = 1 HEX

4. Press the EXAMINE KEY again.

Now the lights should look like this:

○○○○ ○○○○ ○○○○ ○○●○ = 2 HEX

5. Press the EXAMINE KEY AGAIN. EX

Now we see:

○○○○ ○○○○ ○○○○ ○○●● = 3 HEX

By repeatedly pressing the EXAMINE key you will see:

○○○○ ○○○○ ○○○○ ○●○○ = 4 HEX
○○○○ ○○○○ ○○○○ ○●○● = 5 HEX
○○○○ ○○○○ ○○○○ ○●●○ = 6 HEX
○○○○ ○○○○ ○○○○ ○●●● = 7 HEX
○○○○ ○○○○ ○○○○ ●○○○ = 8 HEX
○○○○ ○○○○ ○○○○ ●○○● = 9 HEX
○○○○ ○○○○ ○○○○ ●○●○ = A HEX

AND SO ON

By repeatedly pressing the EXAMINE key you can count as high as you wish. WAIT A MINUTE!!! What happens when all the lights are on? Eventually the lights look like this:

●●●● ●●●● ●●●● ●●●● =???? HEX

You are right. The number is FFFF in hexadecimal. What is FFFF hex equal to in the decimal number system? It is:

$$15(16^3) + 15(16^2) + 15(16^1) + 15(16^0) \text{ or}$$

$$61440 \quad + 3840 \quad + 240 \quad + 15 = 65{,}535$$

This is the largest decimal number that can be expressed in sixteen binary bits. You may be wondering how an eight bit computer can reach such a high memory location. It does this by using two 8-bit bytes to reference memory locations. We will discuss this at a later time.

USING THE KEYBOARD

The Data Handler hardware controlled keyboard is composed of 26 keyboard switches which are divided into one group of ten switches and one group of sixteen switches.

The group of sixteen keys in the lower right hand corner of the Data Handler are used for data and address loading. They contain the hexadecimal humbers 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F.

C D E F
8 9 A B
4 5 6 7
Ø 1 2 3

The group of ten switches are located just above the 16 data entry keys. These are function switches which are used to control the computer's operations. The keys are labelled:

CL (Clear) SC (Single Cycle) SI (Single Instruction)
HT (Halt) INT (Initialize) AD (Address) DA (Data)
EX (Examine) RN (Run) DP (Deposit)

CL SC SI HT INT
AD DA EX RN DP

Let's use the keyboard to discover how it operates. We will first work only with the address lights. While we are doing this ignore the data lights completely.

1. Press the CLEAR key.

2. Now press the ADDRESS key.

3. Press the 5 key. You see:

5
○○○○ ○○○○ ○○○○ ○●○●

4. Press the C key. You see:

5 C
○○○○ ○○○○ ○●○● ●●○○

5. Press the F key. You see:

5 C F
○○○○ ○●○● ●●○○ ●●●●

6. Press the 1 key. You see:

5 C F 1
○●○● ●●○○ ●●●● ○○○●

You have entered the hex number 5CF1. You probably noticed that as each key was entered, the previous hex digit was shifted one place to the left to make room for the next digit. If you key in another digit, each of the digits, 5CF1 are shifted left again. The 5 will disappear as the new digit enters on the right. Practice entering other numbers. Here are a few to try. Shade in the lights which appear.

3067	○○○○	○○○○	○○○○	○○○○
FC29	○○○○	○○○○	○○○○	○○○○
FC04	○○○○	○○○○	○○○○	○○○○
FC20	○○○○	○○○○	○○○○	○○○○
FFFC	○○○○	○○○○	○○○○	○○○○
10DA	○○○○	○○○○	○○○○	○○○○

Just for practice, let's go through the process of entering data. We will use these steps over and over again as we enter our programs in the computer.

1. Press the CLEAR key DATA lights and ADDRESS lights go off

DATA ○○○○ ○○○○
ADDR ○○○○ ○○○○ ○○○○ ○○○○

2. Press the ADDRESS key

The ADDRESS MODE light came on. These are next to DATA KEYS

DATA ○
→ ADDR ●
RUN ○

3. Press the F key

ADDR ○○○○ ○○○○ ○○○○ ●●●●

4. Press the C key

ADDR ○○○○ ○○○○ ●●●● ●●○○

Notice F has shifted left and we have address 00FC.

5. Press the 0 key

ADDR ○○○○ ●●●● ●●○○ ○○○○

Once again the numbers shifted left. We have address 0FC0.

6. Press the 0 key again

ADDR ●●●● ●●○○ ○○○○ ○○○○

We have now accessed address FC00 and are ready to enter some data into this memory location.

7. Press the DATA key

The ADDRESS MODE light goes off and the DATAMODE light came on

→ DATA ●
ADDR ○
RUN ○

8. Press the A key

DATA ○○○○ ●○●○
ADDR ●●●● ●●○○ ○○○○ ○○○○

9. Press the 2 key

DATA ●○●○ ○○●○
ADDR ●●●● ●●○○ ○○○○ ○○○○

The data available now shows A2.

10. Press the DEPOSIT key

DATA ●○●○ ○○●○
ADDR ●●●● ●●○○ ○○○○ ○○○●

Continued on page 13

CALCULATOR CALCULU∫ by George McCarty

Professor George McCarty
Department of Mathematics
University of California, Irvine, CA 92717

INTRODUCTION

The pocket calculator is rapidly replacing the slide rule and books of tables for scientific calculation in higher education. But this marvelous gadget also has a wholly new use as a teaching-learning tool: it can illustrate the limit processes of the calculus. Hence the calculator can differentiate, integrate, sum series, and solve differential equations.

This implies a classroom revolution; it also suggests that a scientist or engineer now has a second chance to really understand the calculus through self study on his own machine.

SOME EXAMPLES

A computation that requires hours of labor when done by hand with tables is quite inappropriate as an example or exercise in a beginning calculus course. But that same computation can become a delicate illustration of the theory when the student does it in seconds on his calculator. Furthermore, the student's own personal involvement and easy accomplishment give him reassurance and encouragement. A quick example of such a computation is 1.1^{10}, 1.01^{100}, 1.001^{1000}, ... Another example is $t = 0.1, 0.01, 0.001, \ldots$ in the function $(\sqrt{3t+9} - 3)/t$.

The machine is like a microscope, and its magnification is a hundred millionfold. Thus it can display for us almost infinitesimal regions near a point on a graph. The derivative of $f(x) = \sin x$, for instance, acquires real meaning when a student first appreciates its value at $x = 1$ radian by calculating the difference quotients $[\sin(1+t) - \sin(1)]/t$ for $t = 0.1, 0.01, 0.001, 0.0001$. The sequence of numbers that he calculates can be identified as the slopes of chords on the graph of $\sin x$, and then abstract trigonometric manipulations will relate his sequence to the number $\cos 1$. Similarly, the fun and excitement a student has in calculating for himself some approximations to a few definite integrals, such as

$$\int_0^1 \sqrt{1-x^2}\,dx \quad \text{or even} \quad \int_0^1 \frac{\sin t}{t}\,dt \quad \text{will}$$

give reality to their definition as limits of Riemann sums. When the usual algebraic manipulation of the sums for the integrands

$1, x, x^2$, and perhaps x^3 is augmented by such calculations, the Fundamental Theorem of the Calculus is seen in a new light. Instead of being misunderstood to be part of the definition of the integral, it becomes a genuine theorem that usefully relates two disparate mathematical objects.

Such an approach to the calculus will necessarily introduce a student at once to the elements of numerical analysis: roundoff and truncation errors, numerical quadrature, finding zeros, evaluating functions, and solving differential equations numerically. The student learns respect here for calculation in problems where theoretical methods fail and only numerical solutions exist. However, in other problems after he labors to form a few partial sums for a series like $1 - 1/3 + 1/5 - \ldots$, he will appreciate the ease and power with which the theory gives the limiting value. Perhaps now the calculator's buttons and twinkling lights can seduce the student to a *balanced* understanding of the theory and practice of the calculus.

A TEXTBOOK

I have written a book which is designed to guide such a course: *Calculator Calculus*, available through the PCC Bookstore. It can be used as the principal text for a semester having freshman calculus as prerequisite, or as a supplementary text for a first course. This book is also quite suitable for a self-study review. It begins with an easy introduction to algorithms via a new method for extracting square roots that is suitable even for four-function machines. The second chapter shows how to apply an algorithm of successive substitutions to find the roots of a polynomial equation of degree greater than 2. These first topics accustom the student to his own machine and start him thinking about the limiting processes of sequences. They are easy enough for brighter high-school students to grasp readily. Yet every student experiences a real thrill of early achievement here as he actually solves cubic, quartic and quintic polynomial equations on his own calculator.

Subsequent chapters follow the standard pattern for a calculus course, with heavy emphasis on infinite series and an extra, final chapter treating differential equations.

Each topic begins with a discussion of the theory illustrated by several examples worked out in detail on the calculator. But the spirit of the book is in its many exercises plus some harder and deeper problems.

WHICH MACHINE?

There is an enormous variety of pocket calculators available with features in many different combinations. Many of these calculators are suitable for learning the calculus with this book. *The recommended machine for such work is one which has buttons to calculate trig and log functions, displays at least eight digits, and has an adapter-recharger.* Of course, methods for approximating logarithmic and trigonometric functions are given in this book. Nevertheless, a student who attempts to do the work with a four-or seven-function machine will become distracted by the copious arithmetic and eventually will despair.

Most calculators that satisfy these minimal requirements also have square root and reciprocal functions, a memory, and the internal constant π. Most will also accept arguments for trig functions, in radians as well as degrees. In addition, some such calculators have the ability to convert a result into "scientific notation," with a mantissa and an exponent. These machines also offer superior logical systems called either "Parenthetical" or "Reverse-Polish." These systems are very useful; they enable the machine to accept more complex formulas without the user having to rewrite equations or write down intermediate results. We recommend that students who can afford it use a machine having scientific notation and such a system of logic.

There are even more elaborate calculators. Some have multiple memories, and some may be programmed to perform repetitive computations automatically. Such features could be useful in our work, but they will not be necessary.

SKILL AT CALCULATION

Every student must spend an hour or two with his first calculator just becoming acquainted with its logic and the feel of its buttons. We have found that calculus students can do this on their own by reading the explanations in the owner's manual and working out the simple examples given there. The text has an appendix which teaches about invisible registers and rewriting formulas for computation, evaluating polynomials and Taylor series. It includes some simple suggestions about errors and about handlingnumerical problems.

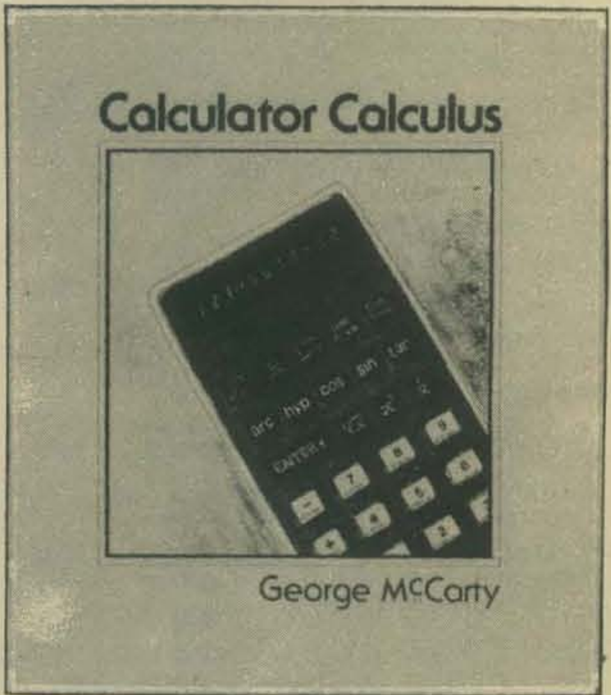
While the book emphasizes the theory of limits and not techniques, its students inevitably acquire considerable skill at getting accurate results with minimal effort.

THE TEACHER'S CALCULATOR

Of course, an instructor cannot merely hold a pocket calculator up to show the numbers on it, even before a small class. The digits are just too little. And every teacher instinctively feels a loss of attention and interest when he reads out the numbers or writes them on the blackboard. We have used closed circuit TV with fair results - there was a close-up lens on the camera right over the calculator. But the best method we have seen for communicating results is a new visual aid called the EduCALC, a teaching calculator with two display units. It has the usual small red numerical readout facing you as you operate it, and also it has another display unit with big neon digits showing the same number on the side of the machine toward your students. It is made by Educational Calculator Devices, Inc., P.O. Box 974, Laguna Beach, CA 92652.

SUMMARY

There is a whole new system available for teaching the calculus. It is based on the precision and large number of significant figures of the pocket calculator. Each limit we study is an arrow pointing to the infinite; a student can understand that arrow by calculating for himself many stages along the way in its direction.



CALCULATOR CALCULUS by George McCarty
$8.95
Page-Ficklin Palo Alto, CA 1975

Available from PCC Bookstore

KALKULATOR KORNER

gene hegedus

I want to tell you how I got to be the Editor of Calculator Lib. After more than 50 years (yes fifty plus!) during which to do high precision mathematics I lugged from the library a 20 pound 15 digit function table (form of giant book) to my desk, fished out of it two adjacent numbers and interpolated for a value between them, I was understandably fascinated by a $50 SR-50 which had more numbers in the 6 ounce box than I had in any of the overweight tables and all those number accessible immediately.

When I mentioned to a "Mathematician" friend (I am only a user of Math not a mathematician) that I am fascinated with the arithmetics I can do with my calculator he answered tersely: "Your statement is invalid because arithmetics and fascination are mutually exclusive."

However I did find a kindred soul in T. C. Mits (The Common Man In The Street) I found that each one of them was fascinated and each one had some trick or wizardry up his sleeve demonstrating proudly with varying degrees of sophistication. As a matter of fact the general approach to math underwent an amazing change during these last three years. An 80 years old gentleman told me that he has been liberated, because he used to spend more than a day to calculate every month the interest percentage value of the checks he received from his investments. Now he does it in an hour or so.

In the beginning I had a dream. Going back in time to the 1920's with an SR-51 in my pocket and became the RECHENMEISTER of the century.

Suddenly my old tricks became colorless. Like getting the sum of log (a + b). You can: log (a + b) = log b + log (a/b+1), with this identity crossing over to trigonometry you do wind up with the sum of a and b. Save time and loss of accuracy through the otherwise necessary conversions. But today this is passè, there is so much more you can do with a calculator. And that is how Calculator Lib was born.

So, the first issue is out. Tips and techniques, it turns out that a long exponential series has a surprisingly simple algorithm for a below $10 calculator. A dictionary of words you can read upside down. New developments from the industry. Episodes from the history of calculation. For the math buff, number theory: a new approach to determine the divisibility of numbers, no matter how many digits (hundreds or thousands). It brings unexpected insights. And a program for a (pseudo) random integer generator between 0 and 36 to simulate a roulette wheel. The same as an algorithm for an inexpensive calculator.

Calculator Lib brings many things for many people!

Gene Hegedus

Some operations are illegal on the calculator. Just to mention a few, trigonometrical functions cannot have a value of more than one; negative numbers do not have logarithms; and division by zero is prohibited.

Those of you who operated the old electromechanical calculators of a few years ago will remember that it was easy to induce a nervous breakdown in those machines. All you had to do was to enter a division by zero. The gears would start to rotate and would have gone on and on until judgement day or till the gears wore out--whichever came first--unless you used a special "stop division" key.

Computers and calculators work with logic instead of gears and a computer can be programmed to print out: "You blockhead, don't you know that you cannot divide by zero." Hand-held calculators are not that spectacular. Some of them do not even act, others display flashing numbers or a special error sign. In any case, it is a good idea to keep in mind: do not try to divide by zero.

The problems may be different with legal operations. If at some place in your report the equation 1 + 1 = 2 appears, you know it and I know it, it is not a big deal. In fact, there are some kindergarten students who know it.

But if you are smart you still will be able to show off that you have a good education in spite of the above equation.

If we cross over into the area of algebra, trigonometry, calculus or whatever, your equations will have letters instead of numbers, many of them Greek. The purpose of using Greek letters in calculations is that if the uninitiated looks at them and exclaims: "This is Greek to me!" you can answer truthfully and courteously: "You are so right."

When you diligently scan mathematical reference books, you can pick up such jewels of identities like $\cos \phi + j \sin \phi = 1$. Or the little trick of calculating e (the base of natural logarithms)

$$e = \lim_{n \to \infty} \left(1 + \frac{1}{n}\right)^n$$

and since ln e = 1 we substitute and can legally write

$$\ln\left[\lim_{n \to \infty} \left(1 + \frac{1}{n}\right)^n\right] = 1$$

Furthermore, take a series like

$$2^0 + 2^{-1} + 2^{-2} + 2^{-3} + \cdots + 2^{-n}$$

this comes awfully close to 2 and in shorthand we write

$$\sum_{n \geq 0} 2^{-n} = 2$$

Now let us set up our equation. How much more impressive it looks if instead of 1 + 1 = 2 we write:

$$\ln\left[\lim_{n \to \infty} \left(1 + \frac{1}{n}\right)^n\right] + \cos\phi + j \sin\phi = \sum_{n \geq 0} 2^{-n}$$

Voila and Q.E.D. (Quite Easily Done.)

Do you have a calculator problem or trick?

Write to Gene Hegedus, Box 2151, Oxnard, CA. 93034 Enclose stamped self-addressed envelope, please.

Data Handler User's Manual, continued from page 11

The data A2 has now been deposited into memory location FC00. Notice that the address has now been changed to FC01. It is ready for new data at the next sequential memory location.

These are the steps that we will use when we start to actually enter programs into the computer. Program steps are entered sequentially. Steps 8 through 10 would be repeated until the entire program had been entered. What you have just accomplished might well be the first step of a computer program in machine language (the language which our machine understands. i.e. binary 8-bit words).

Before we start writing programs, we must learn how to manipulate data inside the computer.

* * * * * * * * *

COURSE OUTLINE FOR FIRST SEMESTER

SESSION I	*Kit construction*
SESSION II	*Kit construction*
SESSION III	*Preliminary checkout of computer* *Discussion of computer specifications* *Computer Notation* *Binary-Hex-Decimal conversion* *Use of address lights for counting* *Use of Keyboard*
SESSION IV	*Data Transfer* *Use of registers (accumulator, x, y)* *6502 instructions (LDA' STA' JMP)* *Memory locations for use* *Use of simple data transfer program*
SESSION V	*Arithmetic Logic Unit* *Addition instructions (ADC, CLD, CLC)* *Use of immediate mode addressing* *Use of absolute addressing*
SESSION VI	*Indexed addressing instructions (LDX, INX, CPX, BNE)* *Looping concept* *Simple data transfer* *8 bit addition*
SESSION VII	*Writing Programs* *Flow charts* *Double precision addition via flow chart* *8 bit multiply using a Trace* *New instructions (ASL,BCS,DEX)*
SESSION VIII	*Writing Programs* *Review of 8 bit multiply* *Development of 8 bit divide through flow chart* *Trace of divide program*
SESSION IX	*Output Devices using output port* *Simple L.E.D. device* *Seven segment display* *Special purpose*

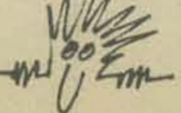
A second semester outline will be provided with Volume 2 of the Data Handler User Manual. This will provide for a one year course for either teacher or student training.

MORE TINY BASIC

by The DRAGON

1975 was the year of build-it-from-a-kit personal computer – the MITS ALTAIR 8800. Others soon followed; today, two dozen companies offer personal computers . . . for the skilled hobbyist or computer expert.

Beginner, beware! Most of the 20,000 to 30,000 people who have acquired personal computers are highly skilled technologists: computer programmers, electronics engineers and technicians, experienced hobbyists, or people who have somehow acquired the high-technology skills necessary to assemble, trouble-shoot, repair, program, operate and understand their sophisticated cybernetic gadgets.

Now for the good news; your turn is coming! 

1977 *could be* the year of the very inexpensive, off-the-shelf-ready-to-use personal computer for beginners. As easy to assemble as a hi-fi system; simply connect a couple of cables, plug in the Tiny Language of your choice, settle down with a good "how-to" book, and learn to use your own computer. Or . . . plug in programs to play the games you have been reading about in PCC for almost 5 years (and more games are on the way!). Plug in a program to convert your color TV to a personal graphics machine, or music machine. Or a program to teach a child to read or do arithmetic or learn music.

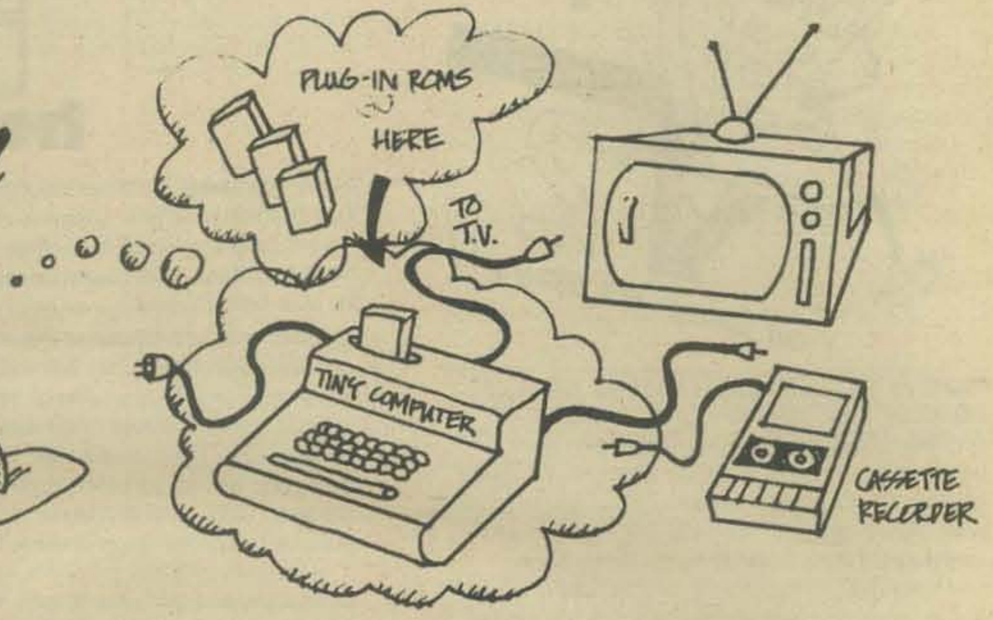
Your ready-to-use personal computer might look like the one we will describe in this article and might cost less than $500, if some intrepid manufacturer would build and sell quite a lot of them. The same low cost computer could be a powerful tool for learning in elementary schools, or a home, or both. All we need are some tools: hardware, software, instructional materials, and lots of new ideas . . . the same system at home, or at school.

Here we go . . . *our* idea of a low-cost, ready-to-use, home/school/personal computer system. Our computer has four components.

1 The computer, which we will describe
2 A black and white or color TV
3 An inexpensive cassette tape recorder
4 One or more plug-in ROMs

Plug-in ROMs? A ROM is a Read Only Memory. Each ROM holds up to 4096 bytes* of pre-programmed memory . . . making it super easy for *beginners* to use the system. A ROM might contain:

- Tiny BASIC, which we will describe here
- Tiny PILOT, described elsewhere in this issue
- Tiny Languages of the Future – to be described in the future – help us invent them!
- A program to convert your TV into a fantastic light/sound show, with *you* at the controls.
- A program to play STAR TREK or Hunt the Wumpus or The Don Quixote Starship or . . .
- A program to convert your system into a powerful scientific, mathematical, statistical or financial calculator
- More! More! More! Let your imagination run rampant. Read about *our* realizable fantasy and send us *your* realizable fantasy.

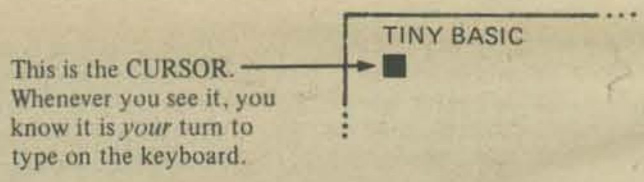


Tiny BASIC

And now . . . imagine that you have just unpacked your Tiny Computer. You attach the TV cable to your TV and plug the power cord into a 110 volt outlet. Don't hook up the cassette recorder yet. We won't need it for awhile. Instead, flip the power on-off switch to ON. You are ready to begin.

> DO THIS: Plug in the Tiny BASIC ROM
> Press READY, then press GO

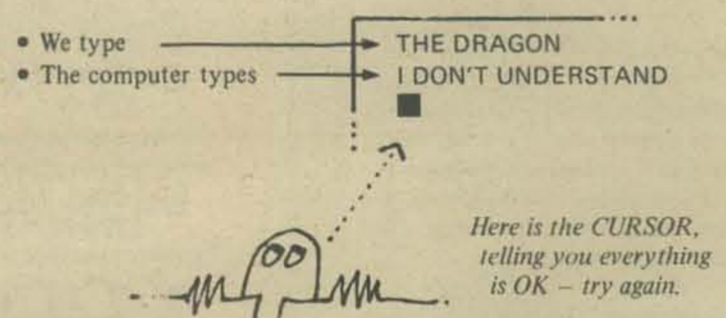
Watch the TV screen.



```
TINY BASIC
■
```

This is the CURSOR. Whenever you see it, you know it is *your* turn to type on the keyboard.

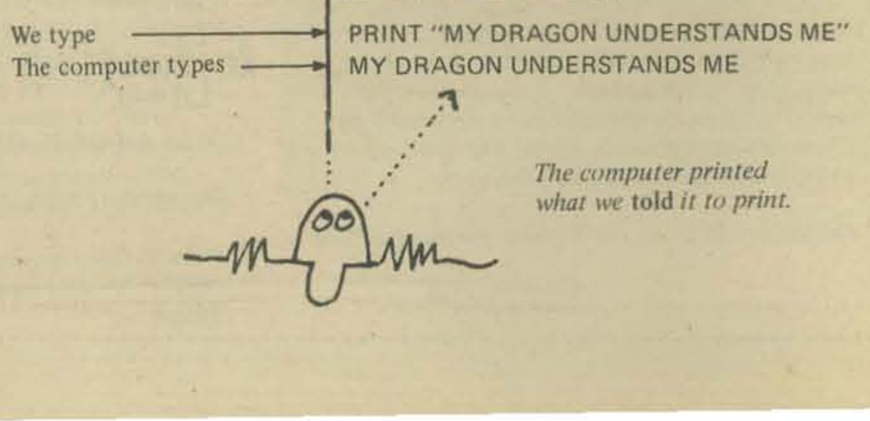
DO THIS: Type your name and press the RETURN key. The computer will probably type: I DON'T UNDERSTAND



```
THE DRAGON
I DON'T UNDERSTAND
■
```

Here is the CURSOR, telling you everything is OK – try again.

Avoid misunderstandings! Learn Tiny BASIC and converse successfully with your friendly personal computer. Follow along as we slowly reveal the mysteries of Tiny BASIC.



```
PRINT "MY DRAGON UNDERSTANDS ME"
MY DRAGON UNDERSTANDS ME
```

The computer printed what we told *it to print.*

* *A byte is a bunch of bits. A bit is a binary digit, 0 or 1. Usually, a byte consists of 8 bits.*

We typed a Tiny BASIC *statement*, called PRINT, which tells the computer to print something on the TV screen.

> The statement: PRINT "$\underbrace{\text{MY DRAGON UNDERSTANDS ME}}$"
>
> tells the computer to print this message on the TV screen.
> The message is *enclosed* in quotation marks.
> The quotation marks are *not* printed.

The message, enclosed in quotation marks, is called a *String*.

PRINT "$\underbrace{\text{MY DRAGON UNDERSTANDS ME}}$"

- This is a string
- The string is enclosed in quotation marks.
- The quotation marks *enclose* the string, but are not part of the string.

Have we boggled, befuddled and bewildered you? Here is another example:

```
• We type         ——► PRINT "LET'S CONVERSE IN TINY BASIC!"
• Computer types  ——► LET'S CONVERSE IN TINY BASIC!
                      ■
```

Now do you know how to get the computer to type your name?

Next, Tiny BASIC does Tiny Arithmetic.

```
• We type         ——► PRINT 3+4
• Computer types  ——► 7
                      ■
```

Note - **no** *quotation marks*

The statement: PRINT 3+4
tells the computer to compute the value of 3+4 and print the result. Since 3+4=7, the computer prints 7.

```
• We type         ——► PRINT 3 – 4
• Computer types  ——► – 1

• We type         ——► PRINT 3 * 4
• Computer types  ——► 12
                      ■
```

As you have probably guessed by now, we can use Tiny BASIC to tell the computer to add, subtract and multiply numbers.

To tell the computer to add, use +
To tell the computer to subtract, use –
To tell the computer to multiply, use *

But what about division?

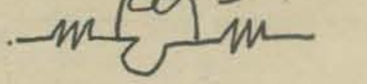
```
• We type         ——► PRINT 3/4
• Computer types  ——► 0
                      ■
```

To divide, use /

What? 3 divided by 4 is zero? Let's try another division example.

```
• We type         ——► PRINT 22/7
• Computer types  ——► 3
                      ■
```

But I thought 22/7 = 3.14 . . .



When the Tiny BASIC ROM is in charge, our computer does *integer* arithmetic.

```
              3 ◄ - Quotient. Tiny BASIC computes this
Remember?  7)22
             21
              1 ◄— Remainder. Tiny BASIC does not compute this
```

Later, we will show you how to compute the remainder on division. In the meantime, read on!

Since we are on the subject of division, let's see what happens if we violate the ancient taboo on division by zero:

```
• We type         ——► PRINT 8/0
• Computer types  ——► I CAN'T DIVIDE BY ZERO
                      ■
```

Tiny BASIC numbers must be *integers* in the range – 32767 to 32767, inclusive. We will say it again.

> Tiny BASIC numbers must be *integers* in the range: – 32767 to 32767

If you type a number larger than 32767 or smaller than – 32767, the computer will refuse to cooperate.

```
• We type         ——► PRINT 40000
• Computer types  ——► MY NUMBER RANGE IS –32767 TO 32767

• We type         ——► PRINT –32768
• Computer types  ——► MY NUMBER RANGE IS –32767 TO 32767
                      ■
```

A similar error message occurs if the true result of an arithmetic operation (+, –, * or /) is less than –32767 or greater than 32767.

```
• We type         ——► PRINT 1000 * 1000
• Computer types  ——► ANSWER IS NOT IN MY NUMBER RANGE
                      MY NUMBER RANGE IS –32767 TO 32767
                      ■
```

In the above example, the true result is 1,000,000, which is much bigger than 32767.

BOXES

Deep down inside the computer, there are 26 little boxes, labelled A through Z.

Each box can store one number at any one time. We show numbers in some of the boxes.

7 is in box A
5 is in box B

What number is in F?
in L?

Zero (0) is in box ____
?

A	7	N	
B	5	O	
C		P	0
D		Q	
E		R	
F	2	S	25
G		T	
H		U	
I		V	
J	–4	W	
K		X	32767
L	–32767	Y	
M		Z	

DO THIS: Put 8 in box C. That is, write the numeral "8" in the box labelled "C".

DO THIS: Using a *pencil*, put 12 into N.
Then, put 37 into N. But wait! A box can hold only one number at a time. Before you can enter 37 into box N, you must first erase the 12 that you had previously put into N. Do it.

When the computer puts a number into any box, A through Z, it first erases the previous number (if any) in that box.

Oh yes – the boxes are called *memory locations* and the labels, A through Z, are called *variables*. Putting a number in box A is equivalent to *assigning a value* to the variable A. Likewise for B through Z.

➔

Here is how we use Tiny BASIC to put a number in box A (Assign a value to the variable A) and then print the value of A on the TV screen.

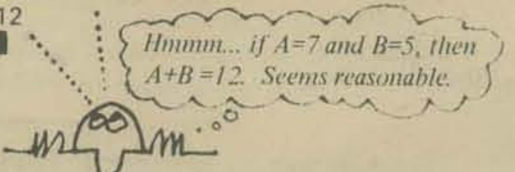
```
• We type          ──────►  LET A = 7
• We type          ──────►  PRINT A
• Computer types   ──────►  7
                            ■
```

The statement: LET A = 7
tells the computer to put 7 into box A

The statement: PRINT A
tells the computer to print the value of A (number in box A) on the TV screen. This statement does not erase the number from box A – it simply causes the computer to (1) peek into box A, (2) see what is there, and (3) type the contents of box A on the TV screen.

Now – follow along, carefully, as we perform some more complicated arithmetical maneuvers.

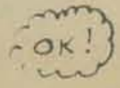
```
                        { LET A = 7
• We type          ─────{ LET B = 5
                        { PRINT A + B
• Computer types   ──────► 12
                           ■
```



Now remember – we have already (above) set A=7 and B=5.

```
• We type          ──────► PRINT 2*A+3*B
• Computer types   ──────► 29
                           ■
```

Let's see now, if A=7 and B=5,
then 2*A+3*B = 2*7+3*5 = 14+15 = 29



MAKING IT EASIER

Our problem for today: Compute the squares of 23, 37, 53 and 88. That is, compute 23^2, 37^2, 53^2 and 88^2. First, we will do it the hard way.

```
• We type          ──────► PRINT  23 * 23
• Computer types   ──────► 529

• We type          ──────► PRINT  37 * 37
• Computer types   ──────► 1369

• We type          ──────► PRINT  53 * 53
• Computer types   ──────► 2809

• We type          ──────► PRINT  88 * 88
• Computer types   ──────► 7744
                           ■
```

Well, OK, but we can give more of the work to the computer. We will do this by *storing a program* in the computer's memory. We will store a Tiny BASIC program to compute N^2 for any value of N.

> We will *store* a Tiny BASIC *program* in the computer's memory.

We will store the following program (don't do it yet).

```
10 LET N = 23
20 PRINT N*N
```

- The above *program* has two *statements.*
- Each *statement* begins with a *line number.*

```
10 LET N=23
```

Line number. A line number must be an integer from 1 to 32767.

When we type statements with line numbers, they are *not* executed immediately. Instead, the statements are stored in memory for later execution.

Next, we will store the above program, but first we must clear out, or erase, any old program that might already be stored in memory.

```
• We type          ──────► CLEAR
• Computer types   ──────► I'M CLEAR
                           ■
```

The computer has CLEARed its memory. So . . . let's type in our program.

```
                   { 10 LET N=23
• We type   ───────{ 20 PRINT N*N
                     ■
```

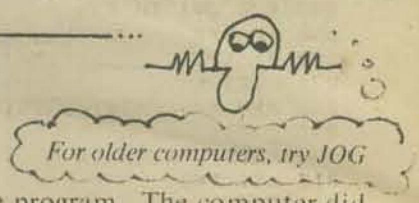
The program is stored in memory. Or is it? Let's find out.

```
• We type          ──────► LIST
                         { 10 LET  N = 23
• Computer types   ──────{ 20 PRINT  N * N
                           ■
```

Fantastic! Whenever we want to see the program in the computer's memory, we type LIST. The computer then lists (prints) the program (if any) on the TV screen.

Secure in the knowledge that our program is, indeed, properly stored in the computer, we decide to tell the computer to do the program. The word for this is RUN.

```
• We type          ──────► RUN
• Computer types   ──────► 529
                           ■
```



When we typed RUN, the computer ran the program. The computer did exactly what the program told it to do

(1) First, the computer did this: 10 LET N=23

(2) Then, the computer did this: 20 PRINT N*N

If you have guessed that the computer does statements *in line number order,* you are *absolutely correct.*

That takes care of 23^2. Now, follow along carefully while we *replace* Line 10 in the program with a new Line 10. **!!!Do NOT type CLEAR!!!**

```
• We type          ──────► 10 LET N=37  ◄─┐
                                          ├── NEW Line 10
• We type          ──────► LIST           │
                         { 10 LET N=37  ◄─┘
• Computer types   ──────{ 20 PRINT N*N  ◄── OLD Line 20
                           ■
```

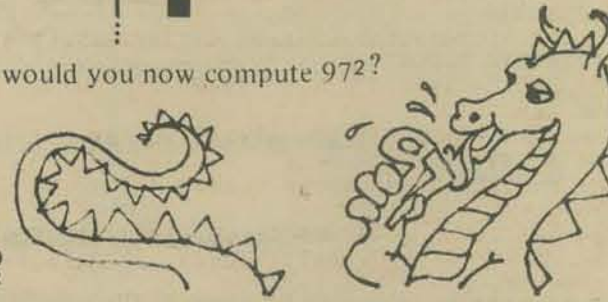
You see? We can replace any line in the program by typing a new line *with the same line number.* Let's RUN the modified program and compute 37^2.

```
• We type          ──────► RUN
• Computer types   ──────► 1369          So, 37²= 1369
                           ■
```

Well, let's complete today's work.

```
                         { 10  LET N=53
• We type          ──────{ RUN
• Computer types   ──────► 2809          Yup, 53²=2809

                         { 10  LET N=88
• We type          ──────{ RUN
• Computer types   ──────► 7744          That's right, 88²=7744
                           ■
```

And so it goes. How would you now compute 97^2?



TO BE CONTINUED

Next time, we will tell you about the INPUT and GO TO statements and loops that go around and around and around and . . . □

REVERSE REVERSE

by Peter Sessions
Reprinted from VOL 1 No.5

```
100 REM *** REVERSE - A GAME OF SKILL
110 REM *** PEOPLE'S COMPUTER COMPANY, MENLO PARK CA
120 RANDOM
130 DIM A(20)
140 REM *** N = NUMBER OF NUMBERS (1 THRU N)
150 LET N=9
160 PRINT "DO YOU WANT THE RULES (1=YES  0=NO)";
170 INPUT A
180 IF A=0 THEN 210
190 GOSUB 710
200 REM *** MAKE A RANDOM LIST A(1) TO A(N)
210 LET A(1)=INT((N-1)*RND(0))+2
220 FOR K=2 TO N
230 LET A(K)=INT(N*RND(0))+1
240 FOR J=1 TO K-1
250 IF A(K)=A(J) THEN 230
260 NEXT J
270 NEXT K
280 REM *** PRINT ORIGINAL LIST AND START GAME
290 PRINT
300 PRINT "HERE WE GO ... THE LIST IS:"
310 LET T=0
320 GOSUB 610
330 PRINT "HOW MANY SHALL I REVERSE";
340 INPUT R
350 IF R=0 THEN 520
360 IF R<=N THEN 390
370 PRINT "OOPS! TOO MANY - I CAN REVERSE AT MOST";N
380 GO TO 330
390 LET T=T+1
400 REM *** REVERSE R NUMBERS AND  PRINT NEW LIST
410 FOR K=1 TO INT(R/2)
420 LET Z=A(K)
430 LET A(K)=A(R-K+1)
440 LET A(R-K+1)=Z
450 NEXT K
460 GOSUB 610
470 REM *** CHECK FOR A WIN
480 FOR K=1 TO N
490 IF A(K)<>K THEN 330
500 NEXT K
510 PRINT "YOU WON IN";T;"MOVES!!!"
520 PRINT
530 PRINT "AGAIN (1=YES  0=NO)";
540 INPUT A
550 IF A<>0 THEN 210
560 STOP
600 REM *** SUBROUTINE *** PRINT LIST A(1) TO A(N)
610 PRINT
620 FOR K=1 TO N
630 PRINT A(K);
640 NEXT K
650 PRINT
660 PRINT
670 RETURN
700 REM *** SUBROUTINE *** PRINT THE RULES
710 PRINT
720 PRINT "THIS IS THE GAME OF 'REVERSE'. TO WIN, ALL YOU HAVE"
730 PRINT "TO DO IS ARRANGE THE LIST OF NUMBERS (1 THROUGH";N;")"
740 PRINT "IN NUMERICAL ORDER FROM LEFT TO RIGHT. TO MOVE, YOU"
750 PRINT "TELL ME HOW MANY NUMBERS (COUNTING FROM THE LEFT) TO"
760 PRINT "REVERSE. FOR EXAMPLE, IF THE CURRENT LIST IS:"
770 PRINT
780 PRINT " 2 3 4 5 1 6 7 8 9"
790 PRINT
800 PRINT "AND YOU REVERSE 4, THE RESULT WILL BE:"
810 PRINT
820 PRINT " 5 4 3 2 1 6 7 8 9"
830 PRINT
840 PRINT "NOW, IF YOU REVERSE 5, YOU WIN!"
850 PRINT
860 PRINT " 1 2 3 4 5 6 7 8 9"
870 PRINT
880 PRINT "NO DOUBT YOU WILL LIKE THIS GAME OF SKILL, BUT"
890 PRINT "IF YOU WANT TO QUIT, REVERSE 0 (ZERO)."
900 PRINT
910 RETURN
999 END
```

If you want a longer or shorter list, just change the value of N in Line 150. N can be anything from 1 to 20. The limit of 20 is set by the DIM statement, Line 130.

For very young people, start with N = 2 or 3, then increase N as they learn the game.

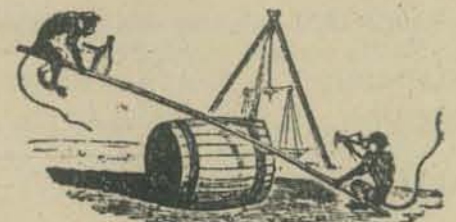
For REVERSE type games for you to program, see the next two pages.

```
RUN
DO YOU WANT THE RULES (1=YES  0=NO)? 0

HERE WE GO ... THE LIST IS:

 3  9  7  5  8  1  4  6  2

HOW MANY SHALL I REVERSE? 6

 1  8  5  7  9  3  4  6  2

HOW MANY SHALL I REVERSE? 9

 2  6  4  3  9  7  5  8  1

HOW MANY SHALL I REVERSE?

 9  3  4  6  2  7  5  8  1

HOW MANY SHALL I REVERSE? 7

 5  7  2  6  4  3  9  8  1

HOW MANY SHALL I REVERSE? 3

 2  7  5  6  4  3  9  8  1

HOW MANY SHALL I REVERSE?
```

Hey! I got 1 in the first place. Too bad the rest of the list is still scrambled.

Would you believe a list with 9 numbers can always be put into order within 15 moves?

I give up! You take over . . .



REVERSE is a simple-minded game, but dangerously addicting.

In an earlier version, we used another way of creating the randomized list of numbers, as follows:

```
210 FOR I=1 TO N
215 LET A(I)=I
220 NEXT I
225 FOR I=1 TO INT(19*RND(0))+19
230 LET X=INT(N*RND(0))+1
235 LET Y=INT(N*RND(0))+1
240 LET Z=A(X)
250 LET A(X)=A(Y)
260 LET A(Y)=Z
270 NEXT I
```

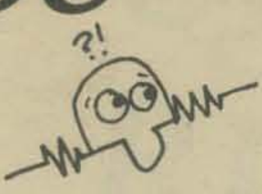
There are many ways to beat the game, but approaches tend to be either algorithmic or heuristic. The game thus offers the player a chance to play with these concepts in a practical (rather than thoretical) context.

An *algorithmic* approach guarantees a solution in a predictable number of moves, given the number of items in the list. For example, one method we found (invented?) guarantees a solution in 2N − 3 moves when the list contains N numbers. The essence of an algorithmic approach is that you know in advance what your next move will be. We could easily program a computer to do this.

A *heuristic* approach takes advantage of "partial orderings" in the list at any moment. Using this type of approach, your next move is dependent on the way the list currently appears. This way of solving the problem does not guarantee a solution in a predictable number of moves, but if you are lucky and clever you may come out ahead of the algorithmic solutions. We could not so easily program this method.

In practice, many players adopt a "mixed" strategy, with both algorithmic and heuristic features. Is this better than either "pure" strategy?

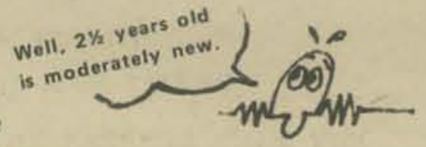
Games for YOU to Program: by the Dragon



ЯƎVƎЯƧƎ

The game of REVERSE was invented by Peter Sessions in the early days of People's Computer Company. REVERSE was first described in the May 1973 issue of PCC (Volume I, number 5) which, alas, is out of print. So . . . we have reprinted the original description of REVERSE on the previous page of this issue. It is written in "Standard" BASIC and should run (or at least jog) on most BASIC systems. If your computer doesn't like the RANDOM statement in Line 120, just delete Line 120 and try again.

For a somewhat different version of REVERSE, let your eyes and brain devour the listing in *What To Do After You Hit Return: PCC's First Book of Computer Games.* And, if you read *Creative Computing,* you will find REVERSE in the Sept.-Oct. 1975 issue, curiously labeled as "Another New Game from Creative Computing . . ."



Here's how REVERSE *WAS* done

- In playing REVERSE, you start with a scrambled list of numbers 1 through N. For example, suppose N = 7.

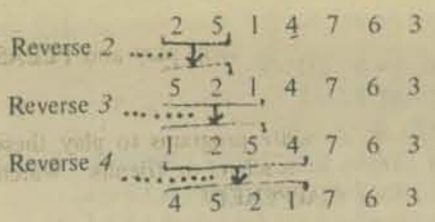
SCRAMBLED LIST: 2 5 1 4 7 6 3

- Your task is to unscramble the list and arrange the numbers in numerical order.

UNSCRAMBLED LIST: 1 2 3 4 5 6 7

- But . . . here is the catch . . . your only operation is *reverse.* You can *reverse* a set of numbers, starting from the left end of the list. You can *reverse* the first two numbers or the first three numbers or the first four numbers, or *reverse* as many as you want, but only from the left end of the list.

Hmmm . . . would you like to see some examples?



```
              2 5 1 4 7 6 3
Reverse 2 ...
              5 2 1 4 7 6 3
Reverse 3 ...
              1 2 5 4 7 6 3
Reverse 4 ...
              4 5 2 1 7 6 3
```

and so on . . .

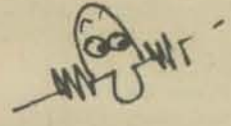
Below, for your reading pleasure, is a complete game, starting with a list of seven numbers.

Scrambled List:	2 5 1 4 7 6 3
Reverse 5:	7 4 1 5 2 6 3
Reverse 7:	3 6 2 5 1 4 7
Reverse 2:	6 3 2 5 1 4 7
Reverse 6:	4 1 5 2 3 6 7
Reverse 3:	5 1 4 2 3 6 7
Reverse 5:	3 2 4 1 5 6 7
Reverse 3:	4 2 3 1 5 6 7
Reverse 4:	1 3 2 4 5 6 7
Reverse 2:	3 1 2 4 5 6 7
Reverse 3:	2 1 3 4 5 6 7
Reverse 2:	1 2 3 4 5 6 7

And so, by diligent application of a simple rule, we transform chaos into order.

Now it's *YOUR* turn

OK, now try creating variations on Peter Sessions' REVERSE. We suggest several easy-to-do variations: you (or your kids) just need to change a line or so of the original program to create the variation. We'd like to see your programs for these games or your ideas for other games.



- Send a Game to the PCC Dragon

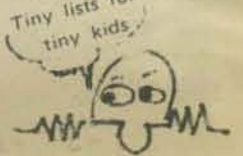
or

- Send an Idea for a Game

We will suggest several variations of REVERSE and show what a partial RUN might look like. However, we will lazily avoid providing you with complete "How to Play" text for each variation. Instead, we eagerly await *your* program for each variation so we can revel in the cogent, lucid and imaginative style in which *you* instruct the computer to describe *your* version of the game to the impatient player.

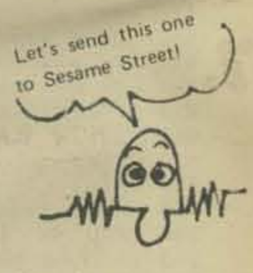
1. The original program provides a scrambled list of 9 numbers, determined by a LET statement in Line 150. To change the number of numbers, rewrite Line 150. If you make the number of numbers larger than 20, also rewrite the DIM statement in Line 130. But . . . instead of all that fuss, why not let the *player* select the number of numbers?

```
HOW MANY NUMBERS DO YOU WANT? 5
HERE WE GO ... THE LIST IS:

4  1  5  3  2

HOW MANY SHALL I REVERSE?...
```



2. Use letters instead of numbers . . . don't forget to change the instructions to the player.

```
HOW MANY LETTERS DO YOU WANT? 7
HERE WE GO ... THE LIST IS:

C  F  A  D  G  B  E

HOW MANY SHALL I REVERSE? 5

G  D  A  F  C  B  E

HOW MANY SHALL I REVERSE?...
```

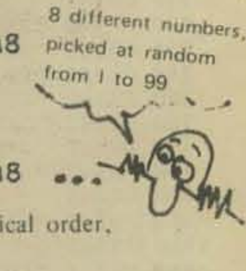


The object, of course, is to put the list in alphabetical order, like this:

```
A  B  C  D  E  F  G
```

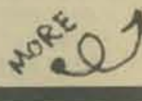
3. Instead of numbers from 1 to N, use N distinct numbers, selected from a larger set. For example, use N numbers selected from the set $\{1, 2, 3, \ldots, 99\}$

```
HOW MANY NUMBERS (UP TO 20)?8
HERE WE GO ... THE LIST IS:
37   12   94   66   75   3   20   48

HOW MANY SHALL I REVERSE?3

94   12   37   66   75   3   20   48  ...
```



and so on. The object, as before, is to put the list in numerical order, as follows:

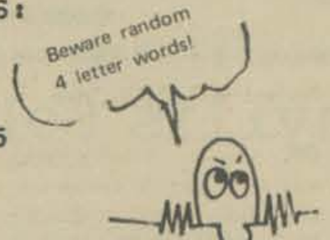
```
3   12   20   37   48   66   65   94
```



»ЯƎVƎЯƧƎ«

4. Use N letters selected from the entire alphabet.

```
HOW MANY LETTERS DO YOU WANT (UP TO 26)?6
HERE WE GO ... THE LIST IS:

P  D  W  E  X  M

HOW MANY SHALL I REVERSE?5

X  E  W  D  P  M  ...
```



and so on. To win, put the list in alphabetical order.

```
D  E  M  P  W  X
```

5. Allow redundant numbers or letters. Here is an example, using numbers:

```
HOW MANY NUMBERS DO YOU WANT?12
HERE WE GO ... THE LIST IS:

2  10  2  4  7  10  5  3  8  8  9  4 ...
```

To win, put the list in the following numerical order:

```
2  3  4  4  5  7  8  8  8  9  10  10
```

6. Why not include negative numbers?

```
HOW MANY NUMBERS DO YOU WANT?7
HERE WE GO ... THE LIST IS:

2  -3  1  3  0  -2  -1  ...
```

The properly unscrambled list should look like:

```
-3  -2  -1  0  1  2  3
```

Since we asked for an *odd* number of numbers, the program we have in mind generates a list symmetric about zero. If we ask for an *even* number of numbers, the list will not be symmetric about zero. For example, if we ask for 8 numbers, we might get a list obtained by scrambling the set
-3, - 2, -1, 0, 2, 3, 4 or a list obtained by scrambling the set
-4, -3, -2, -1, 0, 1, 2, 3 .

7. Fractions! Why not use fractions? Do try this one!

```
HOW MANY NUMBERS DO YOU WANT?6
HERE WE GO ... THE LIST IS:

5/9  1/2  7/8  3/4  2/3  1/7

HOW MANY SHALL I REVERSE?3

7/8  1/2  5/9  3/4  2/3  1/7

HOW MANY SHALL I REVERSE? ...
```

Eventually . . . we want to put the list into the following order:

```
1/7  1/2  5/9  2/3  3/4  7/8
```

↖ ↗

Smallest . . . . to . . . . Largest

8. A different kind of game, using letters. Store a bunch of words in DATA statements. Pick one at random. Scramble the letters and print a list of letters letters. Object? Using the operation of REVERSE, unscramble the letters into the original word.

```
HERE WE GO ... THE LIST IS:

 A  D  O  N  G  R

HOW MANY SHALL I REVERSE?4

 N  O  D  A  G  R

HOW MANY SHALL I REVERSE? ...
```

By luck, design, or ESP, eventually you will get:

```
 D  R  A  G  O  N
```

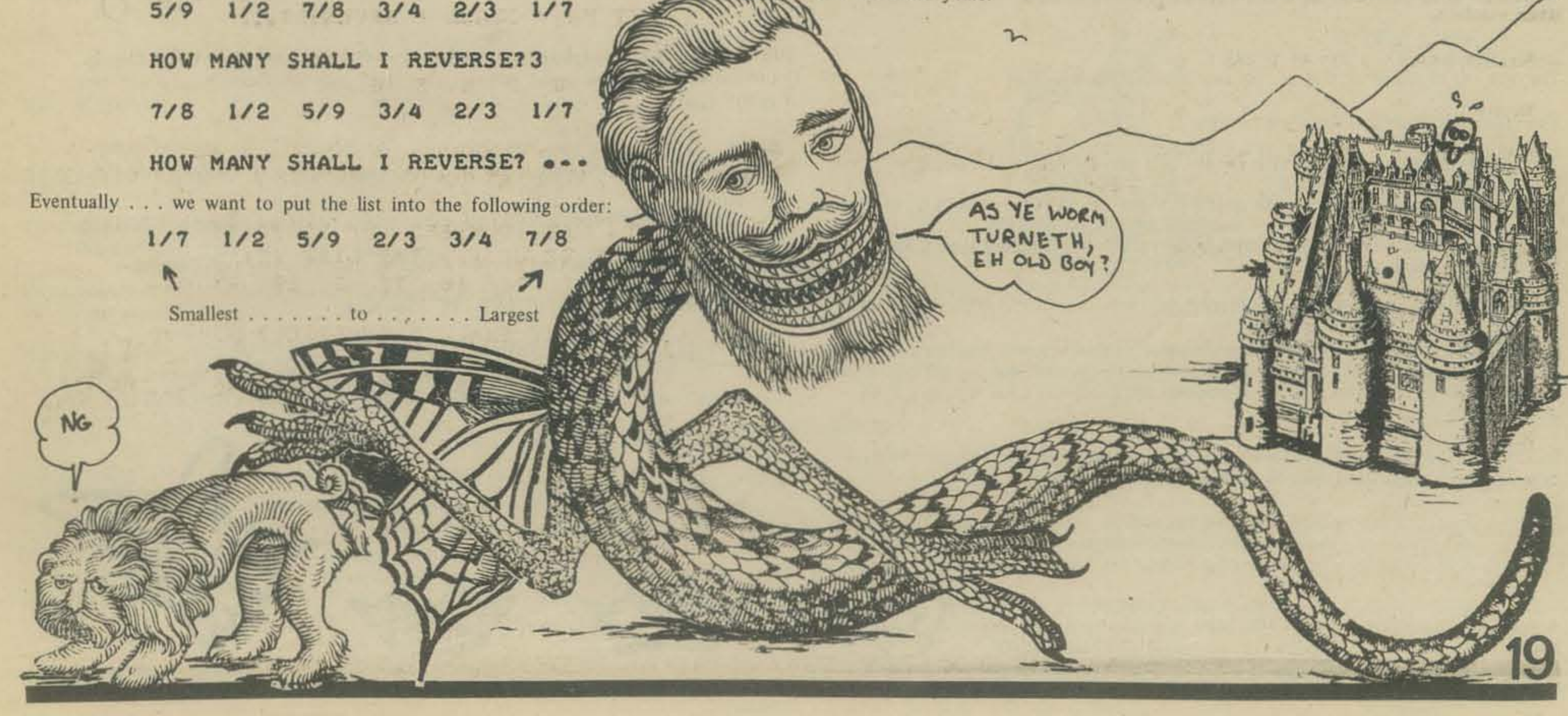
9. Program the computer to play REVERSE.

```
RUN

HOW MANY NUMBERS IN YOUR LIST??7

I'M GAME ... TYPE THE LIST

? 2,5,1,4,7,6,3

HERE WE GO ... THE LIST IS:

2 5 1 4 7 6 3

I REVERSE 5

7 4 1 5 2 6 3
 ...
```

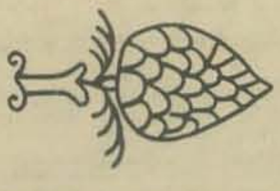
If you are a parent or teacher of an elementary school student, *do* write programs (or help the kids write programs) to play these games. Try them out, with your daughters, sons, or students; watch them play and PLEASE TELL US WHAT HAPPENED!

If you are an elementary school student, *do* write programs to play these games and try them out with your parents or teachers or friends. Watch them play and PLEASE TELL US WHAT HAPPENED!

Whoever you are . . . please help the PCC Dragon enjoy his golden days. Send computer games and ideas for games to the Playful Old PCC Dragon, P. O. Box E., Menlo Park CA 94025. Hurry! Dragons only live for a few thousand years.



tiny pilot



by Dennis Allison & Phyllis Cole

"Tiny" languages were first described in People's Computer Company, Vol. 3, No. 4, March, 1975; since then interest in tiny languages has grown rapidly. Tiny languages are dialects of widely used languages; they are easy to learn and use since only integer arithmetic and a small number of statement types and functions are available. They don't use much memory space and are suitable for implementation on very low cost computers in the neighborhood of $1500. PCC is currently exploring both Tiny BASIC (useful for applications involving a fair amount of computation, such as math recreations and various games) and Tiny PILOT (especially useful for instructional dialogue and reading games). An introduction to PILOT was published in the last issue of PCC, Vol. 5, No. 3.

This article describes what we think ought to be in Tiny PILOT from the user's point-of-view; at the same time we've worried about how to implement the language easily and efficiently on microprocessor based computers. Some useful features have been left out because we want to be able to run in very small memory configurations.

We expect several different implementations of Tiny PILOT will be made if our experience with Tiny BASIC is repeated. We hope that all of these will follow our definition so we'll have a common base language to exchange programs in.

Most of this article is aimed at experienced programmers. Those of you just learning programming should pay attention to the detailed examples: the computerese can sink in gradually.

PROPOSED TINY PILOT

0. INTRODUCTION

This note describes a proposed syntax and semantics for a programming language based on PILOT 73 but suitable for a minimal resources implementation on a microprocessor. The language is embedded in a program creation and editing facility suitable for interactive program development.

To specify even a simple language clearly one needs some notation. We use a variant of Backus-Naur form. Here a symbol in the language or a class of symbols is written as a lower case word, possibly hyphenated.
We write

    name = letter digit

to denote that a name is a letter followed by a digit,

    name = letter digit?

to denote that a name is a letter followed optionally by a digit,
and we write

    name = letter digit*

to denote that a name is a letter followed by zero or more digits. Choices between objects are denoted by writing the objects separated by a slash.
Thus

    letter = A/B/C/D

says a letter is an A or a B or a C or a D.
The or is the exclusive-or; that is, only one item may be selected.

Parenthsis can be used to collect as needed.

Parentheses can be used to collect as needed. Thus a conventional California license plate may be defined as

    CA. license numbers = (three-letters three-digits) / (three-digits three-letters)

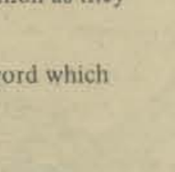
Language specification is further complicated by the need to include typical and systematic specifications.

1. PROGRAMS: INFORMAL DESCRIPTION

A program in TINY PILOT (TP) is a collection of numbered lines. The line numbers impose an ordering on program statements and must be in the range 1 ... 32767. Line numbers serve as names for program statements in exactly the same fashion as they do in BASIC.

In TINY PILOT, all program statements begin with a single letter keyword which determines the statement type. The statements supported are:

T	Type text from terminal
A	Acquire answer from terminal
M	Test answer for match
C	Compute
U	Use as subroutine (GOSUB)
J	Jump (GOTO)
E	Return (or terminate) (RETURN/END)
!	Comment



Almost all TINY PILOT statements have an identical syntactic form:

    line-number keyword condition? colon operand?

A keyword is one of the single letters listed above. A condition is either a parenthetical expression or one of the special conditions denoted by the letters Y and N. These special conditions test the success of the last match operation. A colon is just the character colon, ":". The operands required depend upon the nature of the statement and are discussed below. The one exception to these rules is the comment convention. All material following the keyword "!" is ignored as a comment.

VARIABLES

TP provides 26 numeric variables and 26 string variables. String variables are dynamically allocated storage from a common pool.

Numerical variables are represented as A ... Z, or, in open strings, (explained below) as A# through Z#. String variables are represented as A$ ... Z$. Default values are zero for numeric variables and the null string for string variables.

There are two kinds of strings in TP: *open strings* and *closed strings*. An open string is simply a sequence of characters. A closed string is enclosed in quotes. There are significant differences in interpretation. In an open string, any instance of a variable (string or numeric) is replaced by its current value. In open strings, all numeric variables must be followed by "#" to avoid confusion between variables and one letter words. For example, we want the word "I" and the numeric variable I# to be clearly differentiated.

Example: Suppose these values have been assigned:

```
A$ is "MOBY DICK"
B$ is "MICROCOMPUTERS"
I# is 40
```

open string: A$ IS I# YEARS OLD AND INTERESTED IN B$
Prints as: MOBY DICK IS 40 YEARS OLD AND INTERESTED IN MICROCOMPUTERS

Examples:

```
closed string:  "HOWDY, FRIENDS!"
prints as:       HOWDY, FRIENDS!

closed string:  "USE I# AND A$ IN YOUR PROGRAM."
prints as:       USE I# AND A$ IN YOUR PROGRAM.
```

In a closed string, no replacement takes place; all characters are printed literally. Quote marks in a closed string are represented by two immediately adjacent sets of quote marks.

```
    closed string:
"THE WORD ""I"" MUST NOT BE CONFUSED WITH A NUMERIC VARIABLE."
    prints as:
     THE WORD "I" MUST NOT BE CONFUSED WITH A NUMERIC VARIABLE.
```

Closed strings may be catenated to open strings by simply writing them adjacent.

Examples: Suppose the value of A$ is "HI" then this:

```
                                               string variable
NOW IS THE TIME TO PRINT "A$" WHICH CONTAINS A$
          open string    closed string   open string
```

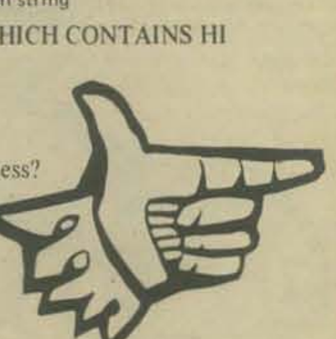
prints as this: NOW IS THE TIME TO PRINT A$ WHICH CONTAINS HI

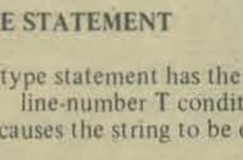
TYPE STATEMENT

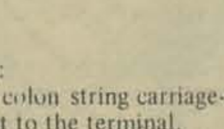
The type statement has the form:

    line-number T condition? colon string carriage-suppress?

and causes the string to be output to the terminal.







Examples:	*What happens when statement is executed:*
100 T: HI THERE!	No condition: "HI THERE!" is printed.
200 TY: WELCOME BACK, A$.	A conditional statement, which is only executed if the flag is 1 (i.e. the last match was successful.) Suppose the last match was successful and the value of A$ is "JENNY." Then this prints "WELCOME BACK, JENNY."

The carriage-return line-feed which is usually produced may be suppressed by using a terminating at-sign. The at-sign was chosen because it will probably be little used in most TP programs. It's particularly useful to suppress the carriage return when user input is to appear directly after the text:

Examples:	*What happens when statement is executed:*
100 T: WHAT'S YOUR NAME@	Text is printed then the printhead (or cursor) remains at the end of "NAME." If input is now requested the input will be printed just after the text.
200 TN: I'M CONFUSED; A$.@	If the flag is zero (i.e. the last match was not successful) and the value of A$ is "TYPE A NUMBER" then "I'M CONFUSED; TYPE A NUMBER." is printed, and the printhead (or cursor) remains in position after the period.

ANSWER STATEMENT

The answer statement has the form:
line-number A condition? colon variable
and causes "? " to be printed and either a string value to be read from the terminal and stored in the string variable or a numeric value to be read and stored in the numeric variable. This command is the one that requests user input. Input requests are shown to the user as a question mark followed by a space. For example, if the user inputs "NO" to an input request then "? NO" will be shown. If the answer statement requests a numeric value and the user types a non-numeric value the message "I NEED A NUMBER FROM – 32767 to 32767" will be given, and a retry allowed.

An answer variable must always be specified in an A statement. If the answer variable is a string, a separate copy is made into an invisible variable for the use of the match statement.

Examples:	*What happens when statement is executed:*
100 A:S	"? " is printed to request input; only numeric input is allowed, and will be stored in numeric variable S.
200 A:Q$	"? " is printed; input is stored in string variable Q$.
300 AY:T	If the flag is 1 (i.e. last match was successful) then "? " is printed; numeric input stored in numeric variable T.
400 A(X=0):U$	If the number stored in numeric variable X is zero then "? " is printed; input is stored in string variable U$.

JUMP

The jump statement has the form:
line-number J condition? colon line-number

Execution of a jump statement causes the program to begin executing statements at the line number specified.

An attempt to use a non-existent statement number is a run-time error and causes a diagnostic to be output and control to revert to program mode.

Examples:	*What happens when statement is executed:*
100 W: 1000	Statement 1000 will be executed next.
200 JY: 79	If flag is 1 (i.e. last match was successful) then statement 79 is executed next.
300 JN: 25	If flag is 0 (i.e. last match was not successful) than statement 25 is executed next.
400 J (X<Y): 88	If the number in X is less than the number in Y, then statement 88 is executed next.

END

The end statement has the form:
line-number Econdition? colon

Execution of an end statement causes the program to begin executing statements at the statement following the statement whose number appears at the top of the subroutine return stack. That number is deleted from the stack. If the stack is empty. the system reverts to program mode.

Confused, you non-programmer? For now, just think "E: for end." Later examples will clarify this.

MATCH STATEMENT

The match statement has the form:
line-number M condition? colon pattern-strings

It causes each of the pattern-strings (see below) to be tested for substring match against the most recently entered answer. If a match is found the flag is set true (=1), if not, it is set false (=0).

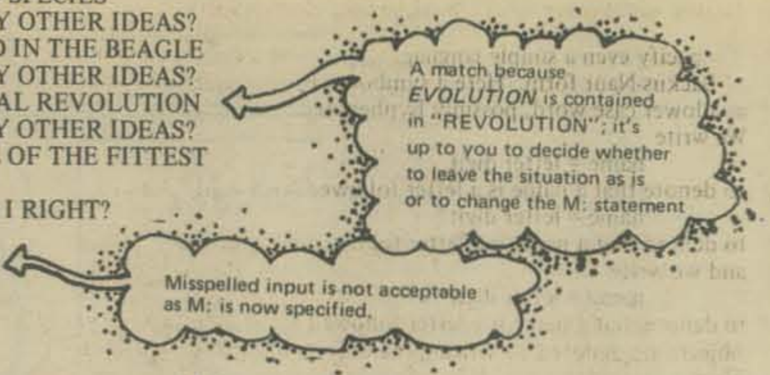
Pattern-strings are lists of open and closed strings; pattern strings are delimited by commas, therefore a comma cannot appear in a pattern except in a closed string. Leading blanks are ignored in open strings.

The match statement makes a character by character string comparison of the latest user input with each M: item, one item at a time. To get a clearer idea of how M: works, consider the following program:

```
100 T:  WHAT WAS DARWIN FAMOUS FOR?
200 A:  X
300 M:  SPECIES, EVOLUTION, BEAGLE
400 TY: RIGHT!  ANY OTHER IDEAS?
500 TN: TRY AGAIN.
600 J:  200
700 E:
```

The program is executed by typing "RUN." This program is written as a loop: it will never stop. We use it to illustrate how M: works.

```
RUN

WHAT WAS DARWIN FAMOUS FOR?
? INVENTED SPECIES
RIGHT!  ANY OTHER IDEAS?
? EXPLORED IN THE BEAGLE
RIGHT!  ANY OTHER IDEAS?
? THE SEXUAL REVOLUTION
RIGHT!  ANY OTHER IDEAS?
? SURVIVAL OF THE FITTEST
TRY AGAIN
? WELL WAS I RIGHT?
TRY AGAIN
? SPESES
TRY AGAIN
etc.
etc.
etc.
```


A match because *EVOLUTION* is contained in "REVOLUTION"; it's up to you to decide whether to leave the situation as is or to change the M: statement
Misspelled input is not acceptable as M: is now specified.

To allow for mispelling, program expected errors or partial words, e.g.
300 M: EVOL, SPEC, SPES, BEAGL, BEEG

To check for specific words, program leading and/or trading blanks: You'll need to use closed strings, since leading blanks are ignored in open strings.
300 M: " EVOL", " SPEC", " BEAGLE "

COMPUTE

The compute statement has the form:
line-number C condition? colon numeric-variable equals expression

The expression is evaluated and the value stored in the numeric-variable. All arithmetic is integer with values in the range -32767 ... 32767. Overflows result in an error message ("NUMBER OUT OF RANGE") and a return to program mode.

All the usual operators (+, -, *, /, AND, OR, XOR, NOT, < , <=, =, >, >=, < >) are provided with the usual precedence. Partial evaluation depth caused by parenthesis or operator precedence is limited to 6.

TINY PILOT continued

****

Examples:	*What happens when statement is executed:*
100 C: X = X + S	5 is added to the value of X and result stored in numeric variable X
200 C (A<0): A= –A	If the value of A is negative then store the absolute value of the number in numeric variable A.
300 C: A=A + (B * 24)/E	Computes the value of A + (B * 24)/E and stores it in numeric variable A.
400 CY: B = (A=3) OR (V=9)	if the last match was successful, then if the value of A is 3 or the value of V is 9, then B is set to 1 (true) otherwise B is set to 0 (false).

USE

The use statement has the form:
    line-number U condition? colon line-number

Execution of a use statement causes the line-number of the current statement to be placed into a subroutine return stack. Program execution continues at the line number specified.

An attempt to use a non-existent statement number is a run-time error and causes a diagnostic to be output and control to revert to program mode.

Here's a program that illustrates both U: and E:. You get to figure out what a run looks like.

```
10  T:  HOW MANY YEARS OLD ARE YOU?
20  A:  X
30  T:  WANNA KNOW HOW MANY MINUTES THAT IS?
40  U:  400
50  T:  HOW MANY YEARS OLD IS YOUR MOTHER?
60  A:  X
70  T:  WANNA KNOW YOUR MOTHER'S AGE IN MINUTES?
80  U:  400
90  T:  HOW MANY YEARS OLD IS YOUR FATHER?
100 A:  X
110 T:  WANNA KNOW YOU FATHER'S AGE IN MINUTES?
120 A:  X
130 U:  400
140 T:  THAT'S ALL!
150 E:
400 A:  W
410 M:  YES, OK, /UP
420 UY: 800
430 E:
800 C:  X = X * 12 * 30 * 24 * 60
810 T:  THAT'S ABOUT X# MINUTES
820 E:
```

Places 40 in subroutine return stack.

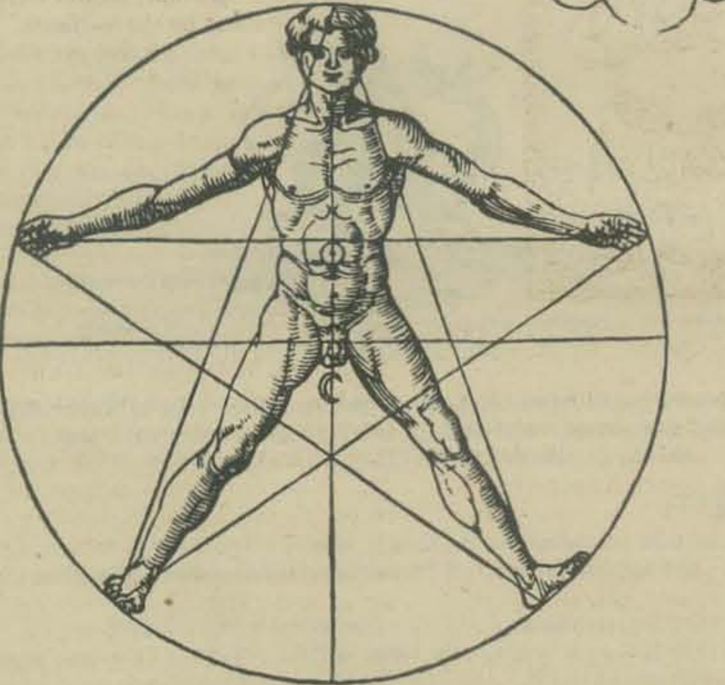
Places 80 in subroutine return stack.

Places 130 in subroutine return stack.

Final end of program.

If flag is 1, then statement 800 is executed next. Otherwise, statement 430 is next.

Next statement after 430 will be 50 or 90 or 140, depending on whether 40, 80, or 130 is in subroutine return stack.



2. EDITING AND CONTROL: INFORMAL DESCRIPTION

TP programs are built and modified in program mode. They are invoked by command available in program mode which initialize and convert to run mode.

The several commands which are available are:

NEW	Clear space and initialize for new program
RUN	Run a TP program
LIST	List the TP program
LOAD	Load a TP program from external media
SAVE	Save a TP program on external media
DEL	Delete lines specified from TP program
SEE	Display value of a TP variable or expression
CONT	Continue after interruption

Lines with line numbers are inserted or appended to the program.

At line entry time, the usual backspace and line delete functions are available just as in BASIC.

RUN takes an optional line-number parameter to indicate the starting line.

LIST and DEL both have the possibility of specifying a range or line numbers or a starting line-number.

SEE displays a single string variable or an arbitrary numeric expression. It is provided as a debug aid.

3. PROGRAM INTERRUPTION

The program can be interrupted (returned to program mode) by typing a break-character at any time. This produces a diagnostic message ("STOPPED BEFORE LINE nnn"). The program can be edited and run again following an interruption.

4. IMPLEMENTATION

One needs a slightly different interpreter mechanism than that used for Tiny BASIC because of the presence of strings, but the same interpretive intermediate language approach would be a plausible implementation strategy. Alternatively, one might use a procedural approach or a table-driven approach which would be somewhat faster.

More on this later as the language firms up.

5. CONCLUSIONS

A very simple version of PILOT has been defined together with some simple editing facilities for program development. Some of the features of PILOT 73 have been eliminated in favor of what we feel is a more consistent and simpler approach.

Comments and counter proposals are solicited. Send them to us c/o PCC, P. O. Box E, Menlo Park, CA 94025.

COMPUTER STORE SURVEY

by Larry Press

PCC *and* **SCCS Interface Magazine** *recently sent a questionnaire to computer stores across the country - it was a joint effort (***PCC** *paid for the printing and* **Interface** *for postage) and is being reported on in both magazines. Any store which would like to be included in our next survey and added to our oft-published, good-publicity list should contact Larry Press, 1702 Ashland, Santa Monica, CA 90405.*

I recently sent a questionnaire survey to computer stores in order to get a feel for what sorts of services they offer, who their customers are, and what sorts of things they are doing in the areas of school and business applications. I have received replies from 35 stores and will summarize the results here.

The first question to be dealt with is: why were there only 35 replies? The obvious answer is laziness–I've been told that 35 out of 105 is a good return rate for questionnaires; however, I believe that the fact that many of the stores are only a few weeks old is also a factor here since they had little to report. Furthermore, a number of the "stores" on our (and other) lists are not stores but OEMs (Original Equipment Manufacturers) or mail order stores for whom the survey was inappropriate.

So much for those who didn't reply, how about those who did? The first area the survey dealt with was schools. We asked if elementary or secondary schools in their areas were using micros (23% said "yes") and if any schools were assembling kits (20% said "yes"). We also found out that onlv 1 store was offering classes for teachers in their area and that 49% were interested in materials to support classes for teachers (authors take note).

The next area of interest was the customers of the stores. We asked people to break down their CPU sales into several categories and received the following response:

	Overall Percent	Range
Hobbyists	62%	6–80%
Professionals	14%	5–75%
Schools	9%	0–25%
Business	15%	5–82%

It appears that some stores are hobbyist oriented and others slant their efforts toward systems for business. Seventeen percent make over half of their sales to business and 40% make more than half of their sales to individuals (hobbyists and professionals).

The average number of CPU sales reported thus far was 48 per store. Half of the stores responding have sold 10 or fewer CPU's and sales ranged from 0 to over 250. A number of stores (large and small) declined to give the absolute numbers, but only reported percentages. It is still early and many of the stores in the "under ten CPU" group may be just opened, or it may be that the public is not yet ready to support retail computer stores outside of high technology urban areas.

I was surprised to learn that book sales account for an average of only about 8% of the business in the stores carrying them and that many do not even carry books. Furthermore, computer store customers buy few tools (perhaps because they're not stocked) and, except for the cases where computers are a sideline at an electronic supply store, components are not very hot items.

Another focus of the survey was the services available at stores. This is what we found:

Service	Percent of Stores
Programming services	49
Maintenance	46
Systems analysis services	43
Software classes	37
Space for local clubs	29
Construction classes	17
Electronic theory classes	9
Chocolate chip cookies	3
Service bureaus	3
PROM burning	3
Kit construction	3
Leasing	3

Finally, several questions dealt with the development of business applications software. We found that 54% of the stores knew of business software projects and that 43% are doing some sort of business software development themselves. About 23% seem to be fairly serious about their business software development. Disc based systems predominate, though a few are working on tape as well.

The following summarize the activity in this area:

Industry Packages	Percent
Doctors offices	17
Lawyers offices	9
Dentists offices	9
Accountants	11
Engineers	6
Retail stores	6
Manufacturing companies	9
Distribution companies	3
Retail businesses	6
Service businesses	6
Industrial control	0
Data processing services	3
Management Information Systems	**Percent**
Production control	9
Finance	3
Marketing	3
Planning	3
General Business	**Percent**
Payroll	26
Accounts receivable	26
Accounts payable	26
General ledger	23
Inventory control	26
General Purpose	**Percent**
Sort and merge	6
Word processing	14
Utilities	6
Program file management (tape)	6
Program file management (disc)	14
Data file management (tape)	6
Data file management (disc)	11

A few of these packages are being developed either by end users or by independent programmers acting as consultants, but the great majority are being developed by the stores themselves.

That's it for now–we'll follow up with another survey soon. □

What Is a computer?

By Jim Day

What is a computer, really? This is an easy question to ask but a very tricky one to answer, even when one "knows" what a computer is. A computer is a lot like the elephant in the fable about the blind men and the elephant. One man grasped the elephant's tail and said that the elephant was like a piece of rope. Another man took hold of the elephant's trunk and said that the elepant was more like a snake. Another encountered one of the elephant's legs and declared that the elephant was very much like a tree. Still another ran into the side of the elephant and insisted that the elephant was a lot like a wall. Each man spoke truly from his own experience but none could agree on what the elephant was really like. So it is with computers. What a computer "really is" depends a lot on one's point of view.

From the point of view of the logician a computer is a mathematical abstraction expressible in Boolean equations. The logic designer deals with the same equations, but expressed as logic diagrams depicting gates and flip-flops in the form of graphic symbols connected by lines. The product engineer has his own schematics, similar to those of the design engineer but with the addition of parts lists, testpoints, and pin connections. He thinks of the computer as a collection of wiring diagrams.

The design draftsman translates the diagrams and parts lists of the product engineer into scaled drawings of printed circuit boards, card cages, wiring harnesses, and panel layouts. His computer is a hierarchy of assembly drawings. Many other people are involved in the design and construction of a computer, and each has his or her own view of what a computer is.

When a computer has been built and delivered to its final destination it still may be viewed quite differently by different people, depending on the nature of their relationship with the computer. Some people may be concerned mainly with the physical operation of a computer. To them, the computer is an operator's console plus one or more control panels and assorted peripheral devices with which they interact in various ways.

There are many different kinds of computer programmers, and each views the computer in a slightly different way, but they can be lumped into three general categories: 1) system programmers, 2) processor programmers, and 3) applications programmers.

The system programmer creates software to extend the facilities provided by the hardware. The functions of such software cover a broad spectrum and include such things as program loaders, editors, and monitors. Most of this software is written in assembly language, partly for speed of execution and partly because system programmers tend to mistrust software written by others–often with good reason. Perhaps in compliance with Parkinson's law, system software seems to expand in size at a more or less constant rate. Since the complexity of the system quickly overshadows that of the hardware, the system programmer quite naturally begins to consider the hardware a mere appendage to the system. To him, the system and the computer are one.

One step removed from the system programmer is the processor programmer who creates software for languages like APL, BASIC, COBOL, and FORTRAN. Typically, the processor programmer isn't too impressed by the magnum opus of the system programmer. He regards the monitor as a necessary evil at best, and thinks of his own brainchild as the crowning glory of the computer–nay, the very computer itself.

Finally, there is the applications programmer, who writes programs in COBOL or some other high-level language. He too identifies his own program with the computer. Well, after all, it *is* doing the computing, isn't it? □

COMPUTER STORES

UPDATE

CALIFORNIA

Action Audio Associates
Westlake Shopping Center
323 S. Mayfair
Daly City, CA 94015

Correction from PCC Vol. 5, No. 3: there is no "Byte Westchester" at 8711 La Tijera Blvd. Los Angeles.

Computers and Stuff
644 Via Alama
San Lorenzo, CA 94580

Computerware
830 First St., Suite B
Encinitas, CA 92024

Correction from PCC Vol. 5, No. 3. CTI Data Systems of Long Beach, CA is not a computer store.

Cyberdux
Microcomputer Applications
1210 Santa Fe Drive
Encinitas, CA 92024

Databus
354 Springfield St.
Claremont, CA 91711

Correction from PCC Vol. 5 No. 3:
The Data Center
136 N. Maryland Ave.
Glendale, CA 91206
(Not in Los Angeles)

Lewis Fite
620 Illinois Court No. 9
El Segundo, CA 90245

Microprocessor Marketing
943 Industrial Avenue
Palo Alto, CA 94303

New Address:
Sunshine Computer Co.
Carson Mall
20700 So. Alameda
Carson, CA 90745

Vogel Associates
Box 4145
Mountain View, CA 94040

FLORIDA

Computer Associates Inc.
6900 N. Kendall No. A103
Miami, FL 33156

Electronics for Yachting, Inc.
2001 SW 20th St.
Fort Lauderdale, FL 33315

Sunny Computer Stores
1238 A South Dixie Highway
Coral Gables, CL 33146

GEORGIA

Atlantic Computer Mart
5091 Buford Highway
Atlanta, GA 30340

ILLINOIS

American Microprocessor s
241 Indian Creek Box 515
Prairie View, IL 60069

Data Domain
42 W. Roosevelt
Lombard, IL 60148

(Updated from PCC Vol. 5 No. 3)
itty bitty machine co. inc.
1316 Chicago Ave.
Evanston, IL 60201

The Lillipute Computer Mart
4446 Oakton
Skokie, IL 60076

INDIANA

(Updated from PCC Vol. 5 No. 3)
Data Domain
7027 Michigan Rd.
Indianapolis, IN 46268

(Updated from PCC Vol. 5 No. 3)
Data Domain
406 S. College Ave.
Bloomington, IN 47401

Data Domain
7027 Michigan Rd.
Indianapolis, IN 46268

Data Domain
219 Columbia
West Lafayette, IN 47905

Hobbytronic Distributors
1218 Prairie Drive
Bloomington, IN 47401

KANSAS

The Computer Hut Inc.
521 N. Hillside Ave.
Wichita, KS 67214

KENTUCKY

Data Domain
506½ Euclid Ave.
Lexington, KY 40502

Data Domain
3028 Hunsinger Lane
Louisville, KY 40220

LOUISIANA

Baxter's TV
7924 Jefferson Highway
Baton Rouge, LA 70809

MARYLAND

New Address:
The Computer Workshop
5709 Frederick Ave.
Rockville, MD 20852

MASSACHUSETTS

The Computer Mart of MA
1097 Lexington
Waltham, MA 02154

MICHIGAN

Newman Computer Exchange Inc.
1250 North Main St.
Ann Arbor, MI 48104
(used stuff; microcomputer catalog)

MISSOURI

The Computer Workshop
6903 Blair Rd.
Kansas City, MO 64152

NEVADA

Johnson TV Inc.
2607 E. Charleston
Las Vegas, NV 89104

NEW HAMPSHIRE

Aircon, Inc.
Route 16B
Union, NH 03887

NEW JERSEY

Midwest Enterprises Inc.
815 Standish Ave.
Westfield, NJ 07090

NEW YORK

Computer Microsystems
6 Wooleys Lane
Great Neck, NY 11023

MJB Research and Development
36 W. 62nd St.
New York, NY 10023

Waco Trading Co. Inc.
239 Park Ave. S.
New York, NY 10003

OHIO

Els Systems
2209 N. Taylor Rd.
Cleveland Heights, OH 44112

PENNSYLVANIA

Caravan
116 S. Garner
State College, PA 11711

Philadelphia Heritage
P.O. Box 1170
Philadelphia, PA 19105

TENNESSEE

MCED Company
1600 Hayes Street No. 101
Nashville, TN 37203

TEXAS

Printing & Office Supply Co. of Texas
130 One Shell Plaza
Houston, TX 77002

Young Electronics Service
Box 5515
Pasadena, TX 77505

VIRGINIA

Jack M. Williams
2415 Ansdel Court
Reston, VA 22091

WASHINGTON

KBC Computer Shop
2735 152 Ave. NE
Redmond, WA 98052

CANADA

Computer Store Division of Hart's
6000 Bourassa E. No.25
Montreal 462, Quebec

First Canadian Computer Store, Inc.
44 Eglinton Street W.
Toronto, Ontario M4R 1A1

The Computer Shop
3515 18th St. S.W.
Calgary, Alberta T2T 4T9

John Crawford
66 Castlefield Ave.
Toronto, Ontario M4R 1G4

The Pacific Computer Store
4509-4511 Rupert St.
Vancouver, British Columbia VSR 2J4

SDS Technical Devices Ltd.
1138 Main St.
Winnipeg, Manitoba R2W 3F3

OTHER

Comicro AG
Badenstrasse 281
CH 8003 Zurich
SWITZERLAND

Computer Resources
Chausee de Charieroi 80
1060 Brussels
BELGIUM

Entryphone Co. Ltd.
172 field Rd.
London SW 10 9AG
ENGLAND

MANUFACTURERS

MICROCOMPUTER & PERIPHERAL

Ai Cybernetic Systems
P.O. Box 4691
University Park, NM 88003
(505) 526-6842

American Microsystems, Inc.
3800 Homestead Rd.
Santa Clara, CA 95051
(408) 249-4550

Apple Computer Co.
770 Welch Rd.
Palo Alto, CA 94304
(415) 326-4248

Applied Micro Technology
100 N. Winchester Blvd.
Santa Clara, CA 95050
(408) 244-8695

Burkeshire Systems
P.O. Box 512
Mountain View, CA 94040

Comp-Sultants
P.O. Box 1016
Huntsville, AL 35800
(205) 837-5100

Comptek
P.O. Box 516
La Canada, CA 91011
(213) 790-7957

Cromemco
2432 Charleston Rd.
Mountain View, CA 94043
(415) 964-7400

The Digital Group
Box 6528
Denver, CO 80206
(303) 777-7133

Dutronics
P.O. Box 9160
Stockton, CA 95208
(209) 951-0516

Ebka Industries Inc.
6920 Melrose Lane
Oklahoma City, OK 73127
(405) 787-3671

E&L Instruments Inc.
61 First St.
Derby, CT 06418
(203) 735-8774

Electronic Control Technology
P.O. Box 6
Union, N.J. 07083

Electronic Product Assoc.
1157 Vega St.
San Diego, CA 92110
(714) 276-8911

Electronic Tool Co.
4736 W. El Segundo Blvd.
Hawthorne, CA 90250
(213) 644-0113

Computalker Consultants
P.O. Box 1951
Santa Monica, CA 90406
(213) 392-5230

Fairchild Micro Systems
1725 Technology Dr.
San Jose, CA 95110
(408) 998-0123

Godbout Electronics
Box 2355
Oakland Airport, CA 94614
(415) 562-0636

Gnat Computers
8868 Balboa
San Diego, CA 92123
(714) 560-0433

Hal Communications Corp.
807 E. Green St.
Urbana, IL 61801
(217) 367-7373

Hamilton/Avnet Electronics
10950 West Washington Blvd.
Culver City, CA 90230
(213) Enterprise 15317

ICOM
6741 Variel Ave.
Canoga Park, CA 91303
(213) 348-1391

IMS Associates
14860 Wicks Blvd.
San Leandro, CA 94577
(415) 483-2093

Infinite Inc.
P.O. Box 906
Cape Canaveral, FL 32920

Info-Tech Inc.
20 Worthington Dr.
St. Louis, MO 63043
(314) 576-5489

Intel
3065 Bowers Ave.
Santa Clara, CA 95051
(408) 246-7501

Intelligent Systems Corp.
4376 Ridgegate Dr.
Duluth, GA 30136
(404) 449-5961

Intersil
10900 N. Tantau Ave.
Cupertino, CA 95014
(408) 257-5450

Lear Siegler Inc.
Data Products
714 N. Brookhurst St.
Anaheim, CA 92803
(714) 774-1010

M&R Enterprises
P.O. Box 61011
Sunnyvale, CA 94088
(408) 738-3772

Martin Research
3336 Commercial Ave.
Northbrook, IL 60062
(312) 498-5060

Matrox Electronic Systems
P.O. Box 56, Ahuntsic Syn.
Montreal, Quebec
CANADA
(514) 481-6838

Microcomputer Assoc. Inc.
2589 Scott Blvd.
Santa Clara, CA 95050
(408) 247-8940

Micro Peripherals Inc.
P.O. Box 22101
Salt Lake City, UT 84122
(801) 566-0201

Micro-Term Inc.
P.O. Box 9387
St. Louis, MO 63117
(314) 645-3656

Mikra-D Inc.
30 Main St.
Ashland, MA 01721
(617) 881-3111

MITS Inc.
2450 Alamo S.E.
Albuquerque, NM 87106
(505) 243-7821

Monolithic Systems Corp.
14 Inverness Dr. East
Englewood, CO 90110
(303) 770-7400

MOS Technology Inc.
950 Rittenhouse Rd.
Norristown, PA 19401
(215) 666-7950

MPI
P.O. Box 22101
Salt Lake City, UT 84122
(801) 566-0201

National Multiplex Corp.
3474 Rand Ave.
S. Plainfield, NJ 07080
(201) 561-3600

National Semiconductor
2900 Semiconductor Dr.
Santa Clara, CA 95051
(408) 732-5000

Ohio Scientific Instruments
11679 Hayden St.
Hiram, OH 44234
(216) 569-7945

Oliver Audio Engineering
7330 Laurel Canyon Blvd.
N. Hollywood, CA 91605
(213) 874-6463

PCM Company
Box 215
San Ramon, CA 94583
(415) 837-5400

Polymorphic Systems
737 S. Kellogg
Goleta, CA 93017
(805) 967-2351

Pronetics Corp.
P.O. Box 18582
Dallas, TX 75228

Processor Technology
6200 Hollis St.
Emeryville, CA 94608
(415) 652-8080

RCA - Solid State Div.
Route 202
Somerville, NJ 08876
(201) 685-6000

Scientific Research Instruments Co.
P.O. Drawer 2096
Ashland, VA 23005

SGS/ATES Semiconductor Corp.
435 G Newtonville Ave.
Newtonville, MA 02160
(617) 969-1610

Signetics
811 E. Arques Ave.
Sunnyvale, CA 94086
(408) 739-7700

Southwest Technical Products
219 W. Rhapsody
San Antonio, TX 78216
(512) 344-0241

Sphere Corp.
P.O. Box 213
Bountiful, UT 84010
(801) 292-8466

S.T.M. Systems
P.O. Box 248
Mount Vernon, NH 03057
(603) 673-2581

Sykes Datatronics
375 Orchard St.
Rochester, NY 14606
(716) 458-8000

Tarbell Electronics
144 Miraleste Dr.
Miraleste, CA 90732
(213) 832-0182

Technical Design Labs Inc.
342 Columbus Ave.
Trenton, JH 08629
(609) 392-7070

The Digital Group
P.O. Box 6528
Denver, CO 80206
(303) 861-1686

Tycom System Corp.
26 Just Rd.
Fairfield, NJ 07006
(201) 227-4141

Vector Electronics Co. Inc.
12460 Gladstone Ave.
Sylmar, CA 91342
(213) 365-9661

Veras Systems
Div. Solid State Sales
P.O. Box 74 E
Somerville, MA 02143
(617) 547-7053

Wave Mate
1015 W. 190th St.
Gardena, CA 90248
(213) 329-8941

Western Data Systems
3650 Charles St.
Santa Clara, CA 95050
(408) 984-7804

Wintek Corp.
902 N. 9th St.
Lafayette, IN 47904
(317) 742-6802

MASTERMIND - A GAME OF LOGIC

by JESSE HEINES & GAY ROSSER

Jesse Heines is a Training Course Developer in the Educational Services Department at Digital Equipment Corporation in Maynard, Massachusetts.

Gay Rosser is Headmistress of St. Mary's Junior and Infant School in Henley-in-Arden, Warwickshire, England.

MASTERMIND is a game that challenges you to develop a logical approach to problem solving. The computer generates a secret code of four colored pegs in a specific order, using any one of six colors in each of four positions. This gives a total of 1296 (6x6x6x6) different codes that the computer can generate. Each time you guess a code, the computer will tell you how many of your guessed pegs match the computer's in both color and position, and how many are correct colors in incorrect positions. If you apply logic to this data, you can usually figure out the secret code with as few as 6 or 7 guesses.

The logic in MASTERMIND involves two steps. First, each guess must be calculated to "tell" you something: it can either verify that a peg of a specific color exists in the secret code, or it can identify the position of a peg for which you already know the color. Second, you must deduce the colors and positions of pegs from the data that the computer gives you on your guesses.

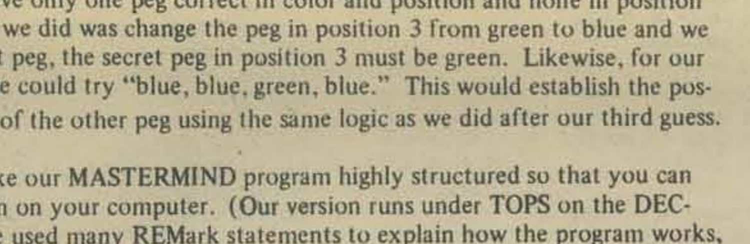
Here is an example. Suppose the secret code is "green, yellow, green, red." (Specific information on the possible colors and the method for telling how many guessed pegs are correct is contained within the program's instructions.) For our first guess, we might try "green, green, blue, blue." The computer would tell us that one of our pegs is correct in both color and position and one is correct in color but not position. The problem is that we don't know which pegs are the correct ones. To find this out, we might try "green, blue, green, blue" as our second guess. Now we would have two pegs correct in both color and position and none in color only. But which ones are they? Since we changed the two middle pegs, we can deduce that one of the correct pegs is in position 1 or 4 and the other is in position 2 or 3. To establish once and for all which of the middle pegs is correct, we could guess "green, blue, blue, blue." The computer would tell us that we now have only one peg correct in color and position and none in position only. Since all we did was change the peg in position 3 from green to blue and we lost one correct peg, the secret peg in position 3 must be green. Likewise, for our fourth guess, we could try "blue, blue, green, blue." This would establish the position and color of the other peg using the same logic as we did after our third guess.

We tried to make our MASTERMIND program highly structured so that you can modify it to run on your computer. (Our version runs under TOPS on the DECsystem-10.) We used many REMark statements to explain how the program works, arranged the program in a logical sequence, supplied a list of the program's variables and functions and their uses, and indented FOR/NEXT loops and conditional code (routines following IF statements).

In addition, we tried to make our program "human-engineered." It contains detailed instructions (lines 1580-2150), allows abbreviations for YES and NO (1510-1520) and colors (1400, 2210-2220, 2670-2680), checks for typing errors (2590), provides an "out" if you can't figure out the secret code (2420-2500), keeps count of the games and guesses (3090-3130, 3210-3290), and lets you rerun the game as many times as you like(3140-3190). These features make the game much more enjoyable for players who are naive about computers by forgiving their errors and helping to convince them that they, not the computer, are in control.

Here's a hint to help you win at MASTERMIND: use four pegs all of the same color as your first guess. This will put you in the best position to apply logic in figuring out the secret code quickly and efficiently (see the sample run). If you'd like to find out about MASTERMIND tournaments, write to Invicta Plastics (USA), Ltd., 200 5th Avenue, New York, N.Y. 10010.

MASTERMIND *is copyrighted by Invicta Plastics (USA), Ltd.; the game is widely available at a cost of $2 to $4.*



D. E. Knuth analyzed the four peg, six color version of Mastermind via computer and published a strategy which guarantees a win in five or fewer moves in the Journal of Recreational Mathematics, Vol. 19, No. 1.

NOTE: Lines that were typed by the player in the sample run are underlined.

```
M A S T E R M I N D


WOULD YOU LIKE TO SEE THE INSTRUCTIONS ?YES


     THE GOAL OF MASTERMIND IS TO FIGURE OUT THE COMPUTER'S SECRET  CODE
WITH  AS  FEW GUESSES AS POSSIBLE.  THE CODE IS MADE UP OF 4 PEGS.  EACH
PEG MAY BE ANY OF 6 COLORS:  RED, YELLOW, GREEN, BLUE, BLACK, OR  WHITE.
A SINGLE COLOR MAY BE USED 2 OR EVEN MORE TIMES IN THE SAME CODE.

     YOU GUESS THE CODE BY TYPING 4 COLORS IN SEQUENCE.  THE COLORS MUST
BE SEPARATED BY COMMAS, AND YOU MUST ALWAYS PRESS THE  RETURN  KEY  WHEN
YOU  FINISH  TYPING.  THE  COMPUTER  WILL  TELL YOU HOW YOU HAVE DONE BY
INDICATING HOW MANY 'KEYS' YOU HAVE SCORED.  BLACK  KEYS  INDICATE  THAT
YOU  HAVE  MATCHED  A  CODE PEG EXACTLY -- YOU HAVE A CORRECT COLOR IN A
CORRECT POSITION.  WHITE KEYS INDICATE THAT YOU HAVE A CORRECT COLOR  IN
AN INCORRECT POSITION.

     FOR EXAMPLE, IF THE SECRET CODE IS 'GREEN, YELLOW, GREEN, RED'  AND
YOU  GUESS 'BLUE, GREEN, RED, RED', YOU WOULD SCORE 1 BLACK KEY FOR YOUR
'RED'  GUESS  IN  POSITION  4  AND 1 WHITE KEY FOR YOUR 'GREEN' GUESS IN
POSITION 2.

     THE  GAME  IS  OVER WHEN YOU HAVE FIGURED OUT THE CODE AND SCORED 4
BLACK KEYS.  IF YOU CANNOT FIGURE OUT THE CODE, TYPE  'UNCLE,X,X,X'  AND
THE COMPUTER WILL TELL YOU IT.  YOU CAN THEN START ANOTHER GAME.

     NOTE:  YOU  MAY  TYPE  THE COLOR NAMES IN FULL OR USE THE FOLLOWING
TWO-LETTER ABBREVIATIONS:
          RD - RED                YW - YELLOW              GN - GREEN
          BU - BLUE               WH - WHITE               BK - BLACK

     GOOD LUCK!  PRESS THE 'CR' KEY WHEN YOU ARE READY TO PLAY.  READY ?
G A M E   # 1

LINE 1 -- YOUR GUESS ?BLUE, BLUE, BLUE, BLUE
   YOU GET 0 BLACK KEYS AND 0 WHITE KEYS.

LINE 2 -- YOUR GUESS ?GREEN, GREEN, GRENE, GREEN
   YOUR 'GRENE' GUESS IN POSITION 3 IS NOT VALID.  PLEASE TRY AGAIN.

LINE 2 -- YOUR GUESS ?GREEN, GREEN, GREEN, GREEN
   YOU GET 1 BLACK KEYS AND 0 WHITE KEYS.

LINE 3 -- YOUR GUESS ?GREEN, RED, RED, RED
   YOU GET 0 BLACK KEYS AND 2 WHITE KEYS.

LINE 4 -- YOUR GUESS ?RED, GREEN, BLACK, BLACK
   YOU GET 1 BLACK KEYS AND 1 WHITE KEYS.

LINE 5 -- YOUR GUESS ?RED, YELLOW, GREEN, YELLOW
   YOU GET 1 BLACK KEYS AND 1 WHITE KEYS.

LINE 6 -- YOUR GUESS ?RED, WHITE, WHITE, GREEN

   CONGRATULATIONS, YOU GET 4 BLACK KEYS!  YOU FIGURED OUT THE CODE IN
   6 LINES!

   THE CORRECT SEQUENCE WAS:  RED - WHITE - WHITE - GREEN

WOULD YOU LIKE TO PLAY AGAIN ?EYS
   PLEASE RESPOND WITH 'YES' OR 'NO'.  YOUR CHOICE ?YES


G A M E   # 2

LINE 1 -- YOUR GUESS ?WH,WH,WH,WH
   YOU GET 0 BLACK KEYS AND 0 WHITE KEYS.

LINE 2 -- YOUR GUESS ?BK,BK,BK,BK
   YOU GET 1 BLACK KEYS AND 0 WHITE KEYS.

LINE 3 -- YOUR GUESS ?BK,BU,BU,BU
   YOU GET 1 BLACK KEYS AND 1 WHITE KEYS.

LINE 4 -- YOUR GUESS ?BU,BK,GN,GN
   YOU GET 1 BLACK KEYS AND 2 WHITE KEYS.

LINE 5 -- YOUR GUESS ?BU,GN,BK,YW
   YOU GET 0 BLACK KEYS AND 3 WHITE KEYS.

LINE 6 -- YOUR GUESS ?GN,BU,RD,BK
   YOU GET 1 BLACK KEYS AND 3 WHITE KEYS.

LINE 7 -- YOUR GUESS ?GN,BU,BK,RD
   YOU GET 2 BLACK KEYS AND 2 WHITE KEYS.

LINE 8 -- YOUR GUESS ?GN,BK,BU,RD

   CONGRATULATIONS, YOU GET 4 BLACK KEYS!  YOU FIGURED OUT THE CODE IN
   8 LINES!

   THE CORRECT SEQUENCE WAS:  GREEN - BLACK - BLUE - RED



WOULD YOU LIKE TO PLAY AGAIN ?NO



YOU FIGURED OUT THE SECRET CODE IN 2 OUT OF 2 GAMES WITH AN AVERAGE
OF 7 LINES PER GAME.

BYE!
```

→

MASTERMIND listing

```
1000 REM        M A S T E R M I N D  --  A  G A M E  O F  L O G I C
1010 REM
1020 REM        WRITTEN BY:  JESSE HEINES AND GAY ROSSER, SEPTEMBER 1976
1030 REM
1040 REM
1050 REM        ADAPTED BY PERMISSION FROM THE ORIGINAL GAME COPYRIGHTED
1060 REM        BY INVICTA PLASTICS (USA), LIMITED, NEW YORK, N.Y. 10010
1070 REM
1080 REM
1090 REM
1100 REM  ****  VARIABLE DIRECTORY
1110 REM
1120 REM        A$      ALPHANUMERIC PLAYER INPUT
1130 REM        C$(X)   SECRET CODE
1140 REM        F0(X)   GUESS FLAGS  (1 = MATCH ON THIS CODE PEG)
1150 REM        F1(X)   CODE FLAGS   (1 = EXACT OR COLOR MATCH WITH THIS
1160 REM                  GUESS PEG)
1170 REM        G$(X)   PLAYER'S GUESSES
1180 REM        K0,K1   LOOP INDICES
1190 REM        L0      LINE COUNTER
1200 REM        P$(X)   COLORS OF PEGS
1210 REM        S0      EXACT MATCH COUNTER -- "BLACK KEYS"
1220 REM        S1      COLOR MATCH COUNTER -- "WHITE KEYS"
1230 REM        T0      TOTAL NUMBER OF LINES GUESSED
1240 REM        T1      TOTAL NUMBER OF GAMES PLAYED
1250 REM        T2      TOTAL NUMBER OF GAMES WON
1260 REM        X       CODE NUMBER OF RANDOMLY SELECTED PEG
1270 REM
1280 REM  ****  FUNCTION DIRECTORY
1290 REM
1300 REM        LEFT$(X$,Y)     RETURNS THE FIRST THROUGH YTH CHARACTERS
1310 REM                          OF X$
1320 REM        RND(0)          RETURNS A RANDOM NUMBER BETWEEN 0 AND 1
1330 REM
1340 REM  ****  DIMENSION ARRAYS AND INITIALIZE VARIABLES
1350 REM
1360     DIM C$(8), G$(4), P$(12), F0(4), F1(4)
1370     FOR K0=1 TO 12
1380       READ P$(K0)
1390     NEXT K0
1400     DATA RD,YW,GN,BU,WH,BK,RED,YELLOW,GREEN,BLUE,WHITE,BLACK
1410     RANDOMIZE
1420     T0=0
1430     T1=0
1440     T2=0
1450 REM
1460 REM  ****  PRINT TITLE AND ASK ABOUT DIRECTIONS
1470 REM
1480     PRINT "M A S T E R M I N D"
1490     PRINT
1500     PRINT
1510     PRINT "WOULD YOU LIKE TO SEE THE INSTRUCTIONS";
1520     INPUT A$
1530     IF LEFT$(A$,1)="Y" THEN 1600
1540     IF LEFT$(A$,1)="N" THEN 2000
1550       PRINT "   PLEASE RESPOND WITH 'YES' OR 'NO'.  YOUR CHOICE";
1560       GOTO 1520
1570 REM
1580 REM  ****  PRINT INSTRUCTIONS
1590 REM
1600     PRINT
1610     PRINT
1620     PRINT
1630     PRINT "     THE GOAL OF MASTERMIND IS TO FIGURE OUT THE COMPUT";
1640     PRINT "ER'S SECRET  CODE"
1650     PRINT "WITH  AS  FEW GUESSES AS POSSIBLE.  THE CODE IS MADE UP";
1660     PRINT " OF 4 PEGS.  EACH"
1670     PRINT "PEG MAY BE ANY OF 6 COLORS:  RED, YELLOW, GREEN, BLUE, ";
1680     PRINT "BLACK, OR  WHITE."
1690     PRINT "A SINGLE COLOR MAY BE USED 2 OR EVEN MORE TIMES IN THE ";
1700     PRINT "SAME CODE."
1710     PRINT
1720     PRINT "     YOU GUESS THE CODE BY TYPING 4 COLORS IN SEQUENCE.";
1730     PRINT "  THE COLORS MUST"
1740     PRINT "BE SEPARATED BY COMMAS, AND YOU MUST ALWAYS PRESS THE  ";
1750     PRINT "RETURN  KEY  WHEN"
1760     PRINT "YOU  FINISH  TYPING.  THE  COMPUTER  WILL  TELL YOU HOW";
1770     PRINT " YOU HAVE DONE BY"
1780     PRINT "INDICATING HOW MANY 'KEYS' YOU HAVE SCORED.  BLACK  KEY";
1790     PRINT "S  INDICATE  THAT"
1800     PRINT "YOU  HAVE  MATCHED  A  CODE PEG EXACTLY -- YOU HAVE A C";
1810     PRINT "ORRECT COLOR IN A"
1820     PRINT "CORRECT POSITION.  WHITE KEYS INDICATE THAT YOU HAVE A ";
1830     PRINT "CORRECT COLOR  IN"
1840     PRINT "AN INCORRECT POSITION."
1850     PRINT
1860     PRINT "     FOR EXAMPLE, IF THE SECRET CODE IS 'GREEN, YELLOW,";
1870     PRINT " GREEN, RED'  AND"
1880     PRINT "YOU  GUESS 'BLUE, GREEN, RED, RED', YOU WOULD SCORE 1 B";
1890     PRINT "LACK KEY FOR YOUR"
1900     PRINT "'RED'  GUESS  IN  POSITION  4  AND 1 WHITE KEY FOR YOUR";
1910     PRINT " 'GREEN' GUESS IN"
1920     PRINT "POSITION 2."
1930     PRINT
1940     PRINT "     THE  GAME  IS  OVER WHEN YOU HAVE FIGURED OUT THE ";
1950     PRINT "CODE AND SCORED 4"
1960     PRINT "BLACK KEYS.  IF YOU CANNOT FIGURE OUT THE CODE, TYPE  '";
1970     PRINT "UNCLE,X,X,X'  AND"
1980     PRINT "THE COMPUTER WILL TELL YOU IT.  YOU CAN THEN START ANOT";
1990     PRINT "HER GAME."
2000     PRINT
2010     PRINT "     NOTE:  YOU  MAY  TYPE  THE COLOR NAMES IN FULL OR ";
2020     PRINT "USE THE FOLLOWING"
2030     PRINT "TWO-LETTER ABBREVIATIONS:"
2040     PRINT "   RD - RED                 YW - YELLOW        GN - G";
2050     PRINT "REEN"
2060     PRINT "   BU - BLUE                WH - WHITE         BK - B";
2070     PRINT "LACK"
2080     PRINT
2090     IF LEFT$(A$,1)="N" THEN 2140
2100       PRINT "     GOOD LUCK!  PRESS THE 'CR' KEY WHEN YOU ARE READ";
2110       PRINT "Y TO PLAY.  READY";
2120       INPUT A$
2130       PRINT
2140     PRINT
2150     PRINT
2160 REM
2170 REM  ****  SELECT SECRET CODE FOR THIS GAME & RESET ROUND COUNTER
2180 REM
2190     FOR K0=1 TO 4
2200       X=INT(6*RND(0))+1
2210       C$(K0)=P$(X)
2220       C$(K0+4)=P$(X+6)
2230     NEXT K0
2240     L0=0
2250     PRINT "G A M E   #" T1+1
2260     PRINT
2270 REM
2280 REM  ****  RESET KEY COUNTERS AND ALL FLAGS TO ZERO
2290 REM
2300     S0=0
2310     S1=0
2320     FOR K0=1 TO 4
2330       F0(K0)=0
2340       F1(K0)=0
2350     NEXT K0
2360 REM
2370 REM  ****  ACCEPT A GUESS
2380 REM
2390     L0=L0+1
2400     PRINT "LINE" L0 "-- YOUR GUESS";
2410     INPUT G$(1), G$(2), G$(3), G$(4)
2420 REM
2430 REM  ****  CHECK FOR 'UNCLE'
2440 REM
2450     IF G$(1)<>"UNCLE" THEN 2540
2460       PRINT
2470       PRINT "   O.K., BETTER LUCK NEXT TIME."
2480       T0=T0-L0
2490       T2=T2-1
2500       GOTO 3020
2510 REM
2520 REM  ****  CHECK FOR BAD GUESSES
2530 REM
2540     FOR K0=1 TO 4
2550       FOR K1=1 TO 12
2560         IF G$(K0)=P$(K1) THEN 2620
2570       NEXT K1
2580       PRINT "   YOUR '" G$(K0) "' GUESS IN POSITION" K0 "IS NOT ";
2590       PRINT "VALID.  PLEASE TRY AGAIN."
2600       PRINT
2610       GOTO 2400
2620     NEXT K0
2630 REM
2640 REM  ****  CHECK FOR EXACT COLOR & POSITION MATCHES -- BLACK KEYS
2650 REM
2660     FOR K0=1 TO 4
2670       IF G$(K0)=C$(K0) THEN 2690
2680       IF G$(K0)<>C$(K0+4) THEN 2720
2690         F0(K0)=1
2700         F1(K0)=1
2710         S0=S0+1
2720     NEXT K0
2730 REM
2740 REM  ****  CHECK FOR COLOR MATCHES -- WHITE KEYS
2750 REM
2760     FOR K0=1 TO 4
2770       FOR K1=1 TO 4
2780         IF F0(K0)=1 THEN 2860
2790         IF F1(K1)=1 THEN 2860
2800         IF K0=K1 THEN 2860
2810         IF G$(K0)=C$(K1) THEN 2830
2820         IF G$(K0)<>C$(K1+4) THEN 2860
2830           F1(K1)=1
2840           S1=S1+1
2850           GOTO 2870
2860       NEXT K1
2870     NEXT K0
2880 REM
2890 REM  ****  CHECK FOR WIN (4 BLACK KEYS) & REPORT SCORE IF NO WIN
2900 REM
2910     IF S0=4 THEN 2980
2920       PRINT "   YOU GET" S0 "BLACK KEYS AND" S1 "WHITE KEYS."
2930       PRINT
2940       GOTO 2300
2950 REM
2960 REM  ****  CONFIRM WIN
2970 REM
2980     PRINT
2990     PRINT "   CONGRATULATIONS, YOU GET 4 BLACK KEYS!  YOU FIGURED ";
3000     PRINT "OUT THE CODE IN"
3010     PRINT "  " L0 "LINES!"
3020     PRINT
3030     PRINT "   THE CORRECT SEQUENCE WAS:  ";
3040     PRINT C$(5) " - " C$(6) " - " C$(7) " - " C$(8)
3050     PRINT
3060     PRINT
3070     PRINT
3080 REM
3090 REM  ****  INCREMENT TOTAL COUNTERS AND ASK ABOUT ANOTHER GAME
3100 REM
3110     T0=T0+L0
3120     T1=T1+1
3130     T2=T2+1
3140     PRINT "WOULD YOU LIKE TO PLAY AGAIN";
3150     INPUT A$
3160     IF LEFT$(A$,1)="Y" THEN 2130
3170     IF LEFT$(A$,1)="N" THEN 3230
3180       PRINT "   PLEASE RESPOND WITH 'YES' OR 'NO'.  YOUR CHOICE";
3190       GOTO 3150
3200 REM
3210 REM  ****  PRINT TOTAL SCORES AND STOP
3220 REM
3230     PRINT
3240     PRINT
3250     PRINT
3260     IF T2=0 THEN 3310
3270       PRINT "YOU FIGURED OUT THE SECRET CODE IN" T2 "OUT OF" T1;
3280       PRINT "GAMES WITH AN AVERAGE"
3290       PRINT "OF" T0/T2 "LINES PER GAME."
3300       PRINT
3310     PRINT "BYE!"
3320 END
```

by LEE SCHNEIDER & TODD VOROS

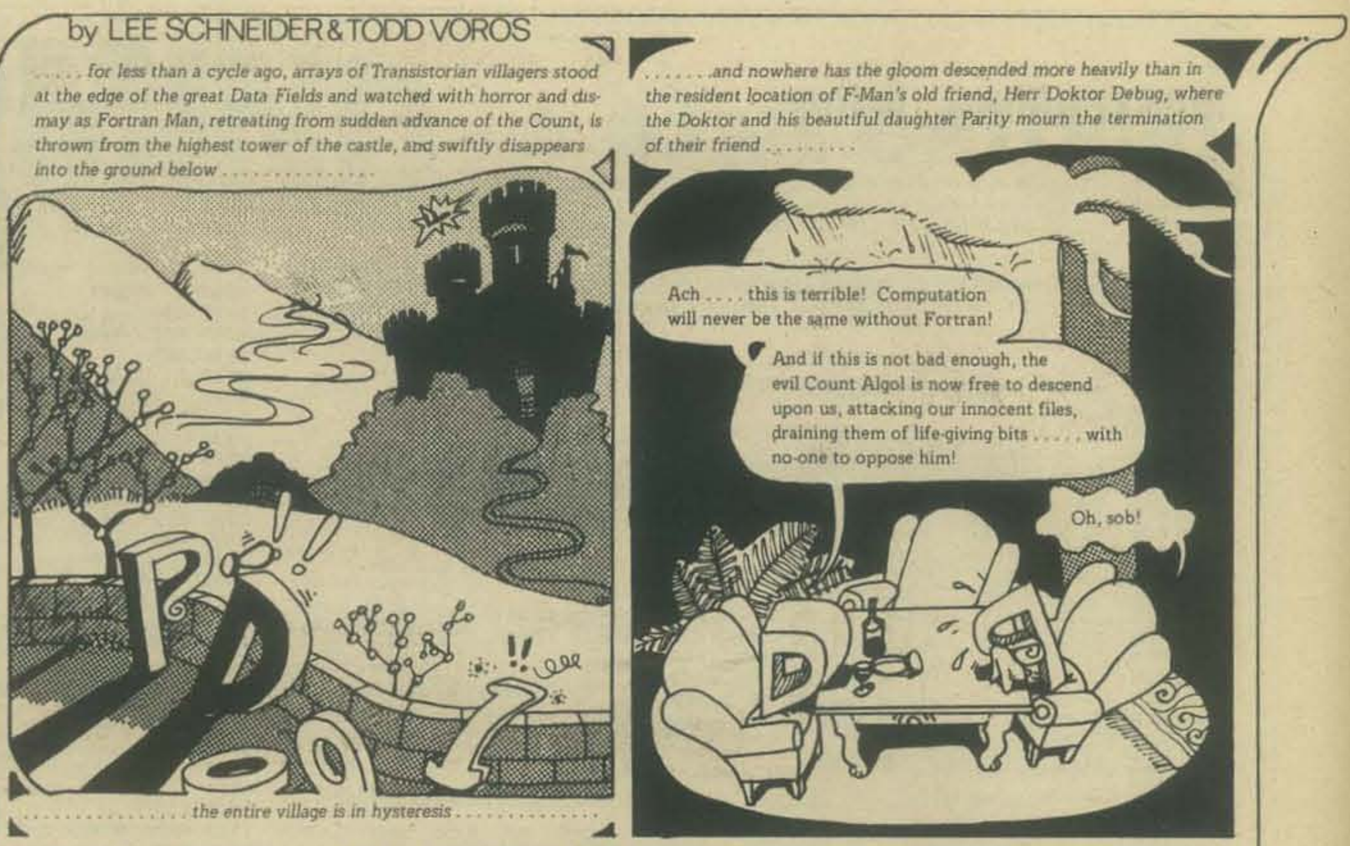
As you may recall from our last episode, things have not been running so well for Our Hero . . . .

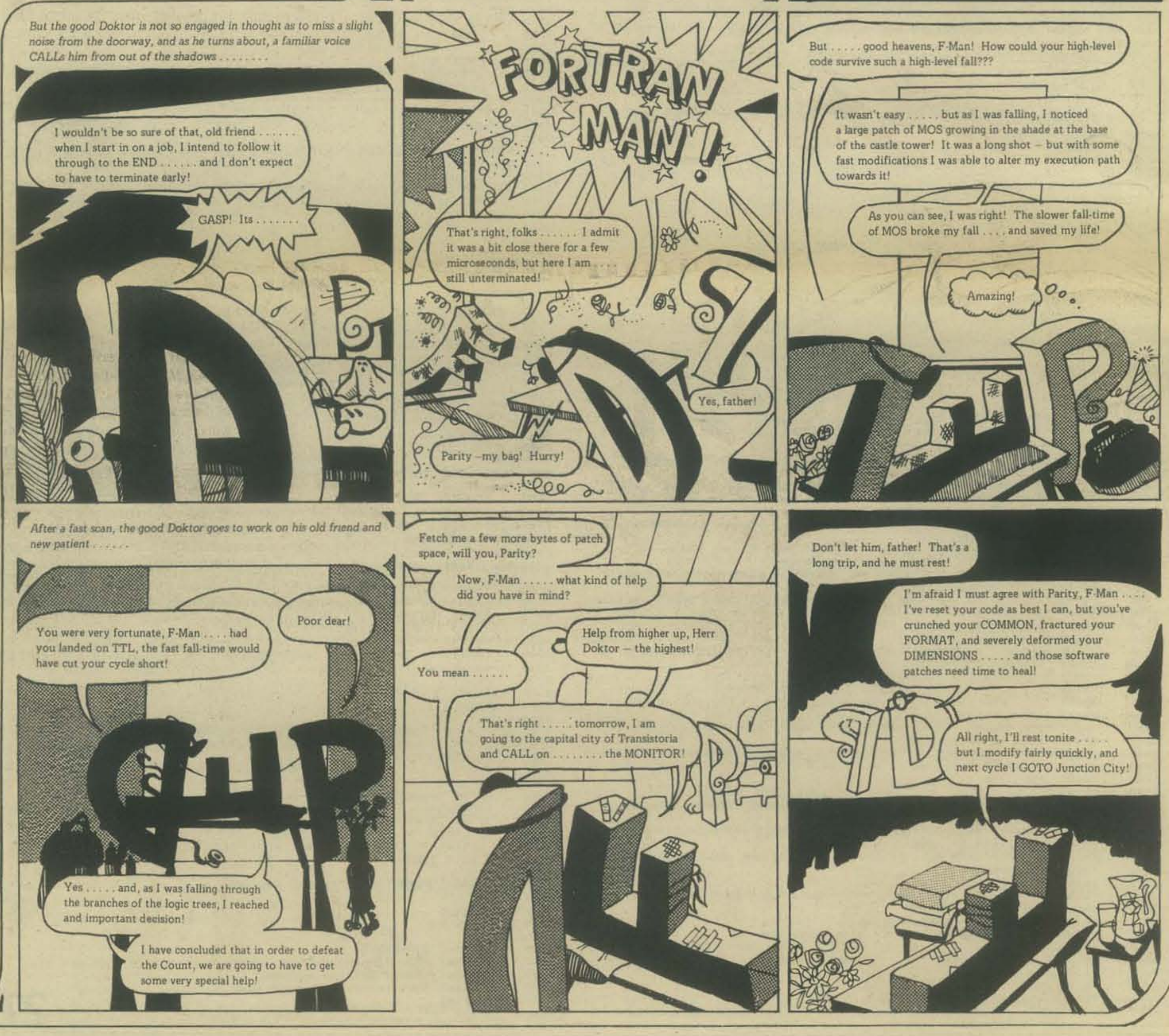
The terrible plague which struck the 'Old Country' of Transistoria had all the indications of being the work of a demented program, and Fortran Man (with the aid of his old friend Herr Doktor Debug) set a trap for the culprit – only to discover that his foe is far more formidable than he had imagined – none other than the ancient and powerful Count Algol!

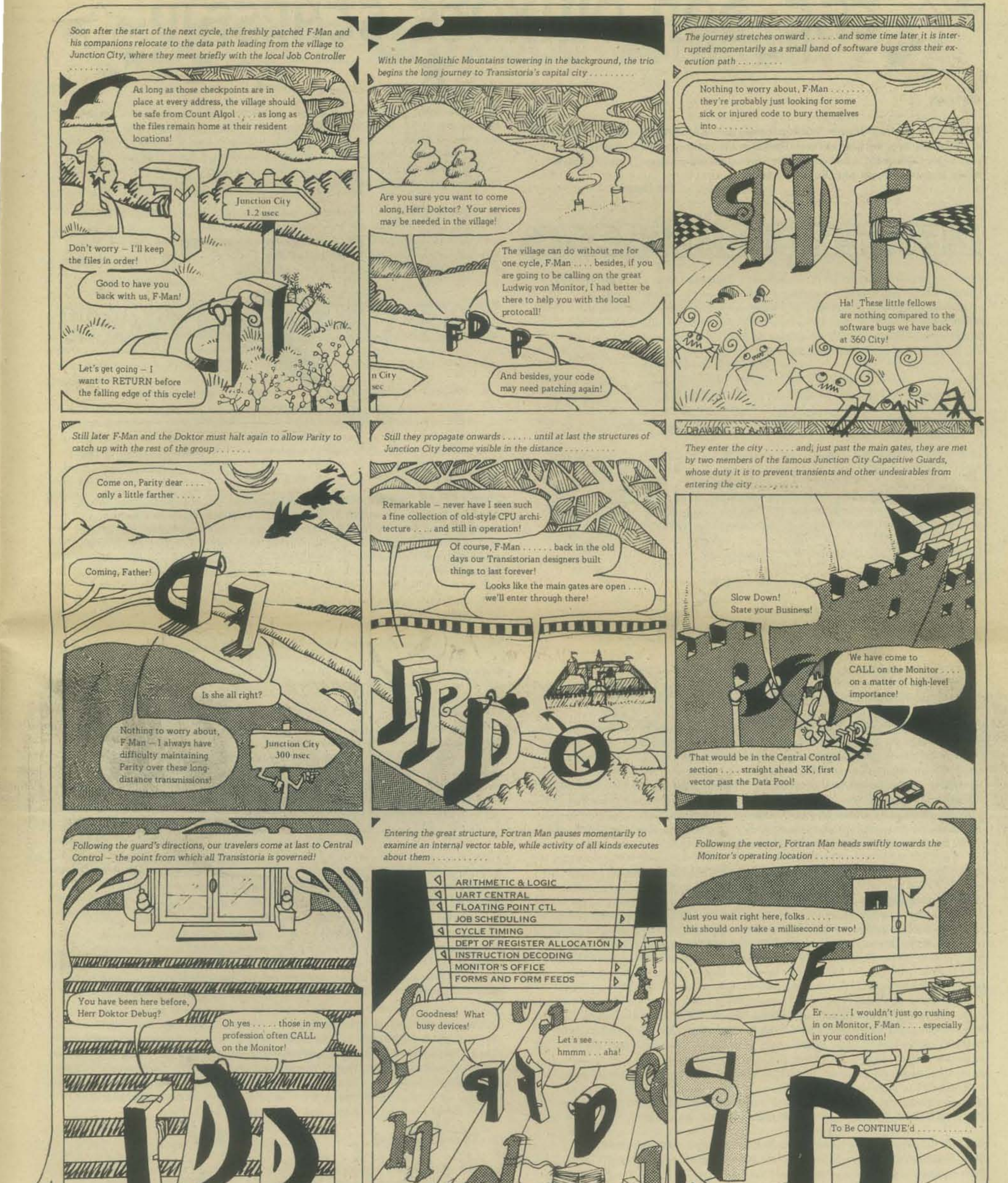
But Our Hero does not retreat in the face of adversity – he pursues his foe to within the very walls of Castle Algol, there to battle the evil Count on his own home ground!

The plan to terminate the Count almost works – but amongst the crumbling data structures of the unrefreshed castle, a slight mishap occurs . . . .

And now, alas – an undercurrent of doom saturates the land of Transistoria . . . .


. . . . . for less than a cycle ago, arrays of Transistorian villagers stood at the edge of the great Data Fields and watched with horror and dismay as Fortran Man, retreating from sudden advance of the Count, is thrown from the highest tower of the castle, and swiftly disappears into the ground below . . . .
. . . . the entire village is in hysteresis . . . .
. . . . . . .and nowhere has the gloom descended more heavily than in the resident location of F-Man's old friend, Herr Doktor Debug, where the Doktor and his beautiful daughter Parity mourn the termination of their friend . . . .
Ach . . . . this is terrible! Computation will never be the same without Fortran!
And if this is not bad enough, the evil Count Algol is now free to descend upon us, attacking our innocent files, draining them of life-giving bits . . . . . with no-one to oppose him!
Oh, sob!


But the good Doktor is not so engaged in thought as to miss a slight noise from the doorway, and as he turns about, a familiar voice CALLs him from out of the shadows . . . .
I wouldn't be so sure of that, old friend . . . . . . when I start in on a job, I intend to follow it through to the END . . . . . . and I don't expect to have to terminate early!
GASP! Its . . . .
FORTRAN MAN!
That's right, folks . . . . . . I admit it was a bit close there for a few microseconds, but here I am . . . . still unterminated!
Parity –my bag! Hurry!
Yes, father!
But . . . . . good heavens, F-Man! How could your high-level code survive such a high-level fall???
It wasn't easy . . . . . but as I was falling, I noticed a large patch of MOS growing in the shade at the base of the castle tower! It was a long shot – but with some fast modifications I was able to alter my execution path towards it!
As you can see, I was right! The slower fall-time of MOS broke my fall . . . . and saved my life!
Amazing!
After a fast scan, the good Doktor goes to work on his old friend and new patient . . . . . .
You were very fortunate, F-Man . . . . had you landed on TTL, the fast fall-time would have cut your cycle short!
Poor dear!
Yes . . . . . and, as I was falling through the branches of the logic trees, I reached and important decision!
I have concluded that in order to defeat the Count, we are going to have to get some very special help!
Fetch me a few more bytes of patch space, will you, Parity?
Now, F-Man . . . . . what kind of help did you have in mind?
Help from higher up, Herr Doktor – the highest!
You mean . . . . . .
That's right . . . . . tomorrow, I am going to the capital city of Transistoria and CALL on . . . . the MONITOR!
Don't let him, father! That's a long trip, and he must rest!
I'm afraid I must agree with Parity, F-Man . . . . I've reset your code as best I can, but you've crunched your COMMON, fractured your FORMAT, and severely deformed your DIMENSIONS . . . . . and those software patches need time to heal!
All right, I'll rest tonite . . . . . but I modify fairly quickly, and next cycle I GOTO Junction City!


Soon after the start of the next cycle, the freshly patched F-Man and his companions relocate to the data path leading from the village to Junction City, where they meet briefly with the local Job Controller . . . .
As long as those checkpoints are in place at every address, the village should be safe from Count Algol . . . as long as the files remain home at their resident locations!
Junction City 1.2 usec
Don't worry – I'll keep the files in order!
Good to have you back with us, F-Man!
Let's get going – I want to RETURN before the falling edge of this cycle!
With the Monolithic Mountains towering in the background, the trio begins the long journey to Transistoria's capital city . . . .
Are you sure you want to come along, Herr Doktor? Your services may be needed in the village!
The village can do without me for one cycle, F-Man . . . . besides, if you are going to be calling on the great Ludwig von Monitor, I had better be there to help you with the local protocall!
n City sec
And besides, your code may need patching again!
The journey stretches onward . . . . . . and some time later it is interrupted momentarily as a small band of software bugs cross their execution path . . . .
Nothing to worry about, F-Man . . . . they're probably just looking for some sick or injured code to bury themselves into . . . .
Ha! These little fellows are nothing compared to the software bugs we have back at 360 City!
DRAWING BY A.MIYA
Still later F-Man and the Doktor must halt again to allow Parity to catch up with the rest of the group . . . .
Come on, Parity dear . . . . only a little farther . . . . .
Coming, Father!
Is she all right?
Nothing to worry about, F-Man – I always have difficulty maintaining Parity over these long-distance transmissions!
Junction City 300 nsec
Still they propagate onwards . . . . . . until at last the structures of Junction City become visible in the distance . . . .
Remarkable – never have I seen such a fine collection of old-style CPU architecture . . . . and still in operation!
Of course, F-Man . . . . . . back in the old days our Transistorian designers built things to last forever!
Looks like the main gates are open . . . . we'll enter through there!
They enter the city . . . . . . and, just past the main gates, they are met by two members of the famous Junction City Capacitive Guards, whose duty it is to prevent transients and other undesirables from entering the city . . . .
Slow Down! State your Business!
We have come to CALL on the Monitor . . . . on a matter of high-level importance!
That would be in the Central Control section . . . . straight ahead 3K, first vector past the Data Pool!
Following the guard's directions, our travelers come at last to Central Control – the point from which all Transistoria is governed!
You have been here before, Herr Doktor Debug?
Oh yes . . . . . those in my profession often CALL on the Monitor!
Entering the great structure, Fortran Man pauses momentarily to examine an internal vector table, while activity of all kinds executes about them . . . .
ARITHMETIC & LOGIC
UART CENTRAL
FLOATING POINT CTL
JOB SCHEDULING
CYCLE TIMING
DEPT OF REGISTER ALLOCATION
INSTRUCTION DECODING
MONITOR'S OFFICE
FORMS AND FORM FEEDS
Goodness! What busy devices!
Let's see . . . . . . hmmm . . . aha!
Following the vector, Fortran Man heads swiftly towards the Monitor's operating location . . . .
Just you wait right here, folks . . . . . this should only take a millisecond or two!
Er . . . . . I wouldn't just go rushing in on Monitor, F-Man . . . . especially in your condition!
To Be CONTINUE'd . . . .
Tune in next issue, folks . . . . . . same time, same system . . . .

THE DON QUIXOTE STARSHIP

HISTORY

The idea for the Don Quixote Starship formed during a carefree Friday afternoon of synergistic beer drinking at Pete's Harbor in the summer of 1975. For the next year or so, DQS rested – and grew. The idea would not go away. Finally, it grew so big, it just had to get out. So . . . we begin.

The Dragon

LETTER FROM FRANK

Bob, you've destroyed my whole equanimity! Yesterday I was a reasonably contented *bleep* employee, bootlegging a little time on the *bleep* computer occasionally, teaching a class once in a while, paying off some debts from my business failure (P. B. S., you know) and generally getting through each day a day at a time. I had no really interesting problems to work on, except a book I'm co-authoring with Jean Pedersen at S. C. U., and no pressing time constraints.

Now, today, after coming to PCC's meeting last week, I'm restless, my bride says I don't listen when she talks to me, I was up way past my bedtime last night, I've started to procrastinate on preparing my lecture notes for tonight's talk. In general I'm distracted by the idea you so casually dropped at that meeting.

How in the hell can such a project ever succeed/get started/progress? You have no plan. You have no schedule. You have no budget. All you have is an idea!

A game of the magnitude you propose is ridiculous! Take two years to design it – preposterous! Involve a thousand or more players – unbelievable! Take a year or more to play, even with computer moderation – unthinkable!

I think the title of the first article in PCC ought to be:

"Get Your Ass in Gear, or We'll Leave Without You"

In the first announcement, we'll describe the project – a space – no, universe – exploration project. We'll establish a communication net first. Find out who will participate at what level. The goal will be to explore the known universe.

I think the first kinds of people we will need to involve will be problem-thinker-uppers; people who can dream of unknown worlds of the intellect and predict what kinds of problems might be encountered.

Then, when the problems start coming in, we will need some problem-solvers. We will also need some information directors who can sort and distribute information between the others, for example: a problem-thinker-upper may propose a whole series of anticipated problems. He or she sends all of these to the information-transferrer, who then sorts and sends some of them to a social-problem-solver, and others to a communications-problem-solver. If the several problem-solvers come up with any solutions or methods-toward-solutions, the information-transferrer sorts and stores the proposed solutions/methods in a form to be further distributed back to the problem-thinker-uppers and other problem-solvers.

Right off the bat I can see a problem of classifying the areas of problem-thinker-uppers [PTUs] and problem solvers [PSs], and perhaps getting some of them into direct communication in order to shorten the turn-around time for their interaction, so that they can report their results to the information-transferrers/storers [ITSs] in more complete form. Perhaps we need several levels of communication: way out [WO], far out [FO], and far freakin' out [FFO] suggest themselves as possible identifiers. Then the WO PTUs could communicate with the WC PSs and the FO PTUs with the FO PSs without having to bother the WO ITSs once they have been put into contact with one another.

I don't know how big the spacecraft ought to be, but I'll bet that somewhere between one cubic metre and 10 to the third cubic metres there's a number that probably makes some kind of logical sense. And I don't know how large the beginning colony ought to be, either, but your guess of ten thousand to ten million probably contains the answer. And if that's the case, capturing an asteroid and hollowing it out might make sense. How in the heck to you capture an asteroid? And once you get it, how do you hollow it out? And, assuming you have a plan that is technically feasible, how do you finance such an endeavor?

And will the game designers turn out to be good game players, or is there something different in the guts of those who are good designers which prevents them from being good players?

Hell, Bob, I just don't think it'll work! Nobody's going to respond to this letter, and if they do, they won't be willing to identify themselves, send cases, offer any problems to be considered, or to advise on how to get started. I bet we don't even get any information sorters/transferrers, who would be willing to read mail, and sort it by category. And who in his right mind will be willing to send in an FFO problem for which he hasn't a solution. And as for problem-solvers, I doubt if anyone even has the foggiest notion of how to determine how big a spacecraft ought to be that must travel indefinitely, let alone be able to estimate the size of the crew necessary.

So, unless we got a lot of response and a lot of people who are willing to think up problems, and a lot of people who will be willing to classify these problems, and then send them in bundles to a lot of other people who are willing to accept a batch of way out, far out, and far freakin' out problems to work on (without knowing where it's heading) I'd suggest you drop the whole project.

Frank Armbruster

MEMO FROM FLASH

Memo To: BOB ALBRECHT – People's Computer Company

Re: THE DON QUIXOTE STARSHIP (the impossible game)

From: the Mellow Flash

The Quixote Quest

Thanks Bob for suggesting such a stupendously delicious enterprise, a cooperative game that will take thousands of folk to design and many more thousands to play. . . to play for years. . . . perhaps forever. It is a space exploration game that would include as much detail and complexity as the knowledge and imaginations of your readers can bring to bear on the problem of exporting an Earth culture to explore and seed the stars. Its scope reminds me of the Glass Bead Game of Herman Hesse and the Lottery in Babylon of Jorge Borges. It would surely be an exercise in a vast communal consciousness, and endless odyssey into the mysterious reaches of the universe!

I foresee that it could be a new and alternative venture into space. It would not be the excessively Yang space race of cultures that project their Macho power and sex urges onto Nature, but rather a venture of Life, one that is both part of the incredibly inter-connected warp and woof of the Universe and an intimate and caring exploration of a living Cosmos.

Unfortunately, the exploration of the moon took place at a time and during a politic that denigrated the spirit of that venture. Although the media made it possible that we could all vicariously participate in the process, its mass exploitation rendered its content banal. TV does that; makes us indifferent to wonder and mystery. Only the explorers and a lucky few others still feel it as a spiritual adventure. It should be a sacred quest; the spirit of earth's consciousness reaching out to explore the chasms of wonder, searching the unknown without/within, and revelling in the delight of discovery of the ever-unfolding mysteries of the Universe.

There are so many questions to ask and to explore and to solve! Who, how many, how and when should we go? (all of us that want to, as soon as we can?). What kind of culture should we take with us, one or many; how will they work, sustain, and evolve? Will they look like the space colonies now being planned (see the CoEvolution Journal)? How will they cooperate; what will they value? Can we sustain enough variety for the homeostasis necessary for long-term survival in new and alien environments? Can we utilize technology and science humanely,. . . lovingly? Can we integrate a yang science with the yin intuition of direct experience (gnosis)? Can we expand our world-views, models, metaphysics, and tolerance in order to encounter the alien, the other . . . and integrating our lives with it??

Ted Nelson once told me of his paradigm for joyful activity in the flow; children who start playing with their own fantasies, that quickly blend together, merge and then expand anew into uncharted and surprising territories. That is what you have sprung on us Bob you old Dragon. It is a vast and stupendous vehicle for the evocation of all our dreams . . . a vast communal dreaming! For it is by dreams and fantasy that we can tap the enormous reservoir of intuition and knowledge that is suppressed by the rules, language, and conventions of our culture.

On COoperative Gaming

So many of the games we play are reflections of warfare, of battles, and of duels between antagonistic opponents. Sometimes this can be a healthy sublimation of the competitive sides of our nature; however, it can also be a sickly reinforcement of warlike tendencies to the neurotic and unwhole. It is a question of balance.

THE DON QUIXOTE STARSHIP

Anyhow, there are too many competitive space games, let us make the Quixote Starship as cooperative a game as we can.

Abstract "game theory" with its simplistic assumptions, implied competitiveness, minimax strategies, and its implied notion of games "against Nature", has a point of view and biases that I would rather not see encouraged . . . But how do we put cooperation into a gaming situation? Nature is not ultimately for or agin us . . . so instead we should game against something like death or non-life, or rather non-game forces. . . the game's the thing eh, Moriarity!

For the Quixote quest, there could be essentially 3 types of game components; autonomous organisms (orgauts), resources, and the processes of Nature. Orgauts differ from resources in that they are generally more complex, often mobile, and relatively negentropic (that is, they tend to pattern the world). The processes are really the laws of the Universe, the transition rules, the Dynamics of the Game.

Cooperation and other "human" side-effects come in because orgauts and resources have finite life-spans (cycles). In order to survive (play longer) orgauts will have to share and help each other as well as conserve resources. Ultimately, the only way to continue playing will be regard and treat the non-human components as a vast living creature that must be nurtured if we are to survive.

Orgauts will have different capacities and skills (resources for negentropic activity. . . i.e. decision-making, communication, organizations, mobility, and power to manipulate resources) mobilities, and organizations. These capacities will be continually consumed by the harshness of life, and will require cooperation, sharing, and trade in order for an individual to continue. Some skills will be more evident (more easily understood by others), for instance material power and possession versus love and wisdom. Different species (other forms of life) can be indicated by having a class of orgaut that requires a different profile of resource/capacities in order to continue playing the game. Inter-species cooperation should be required in most environmental circumstances; these interdependencies may be extraordinarily complex . . . living ecological webs. On an interstellar, intragalactic level, communication between species will be a must. Some of the resources necessary for very long-term survival will have to be sought on other stellar systems.

Game Components

Different species will have different capacities for co-op power (trade and communication), civilization (social grouping, network formation, knowledge systems), and different relationships to ambient resources (different economics, values, belief systems). The eventual success of a class of orgaut will depend heavily upon how material skills are integrated with the ability to communicate, wisdom, problem-solving, love, dreaming, organizational abilities, proclivity to risk and adventure, as well as other skills that are "alien" to us now.

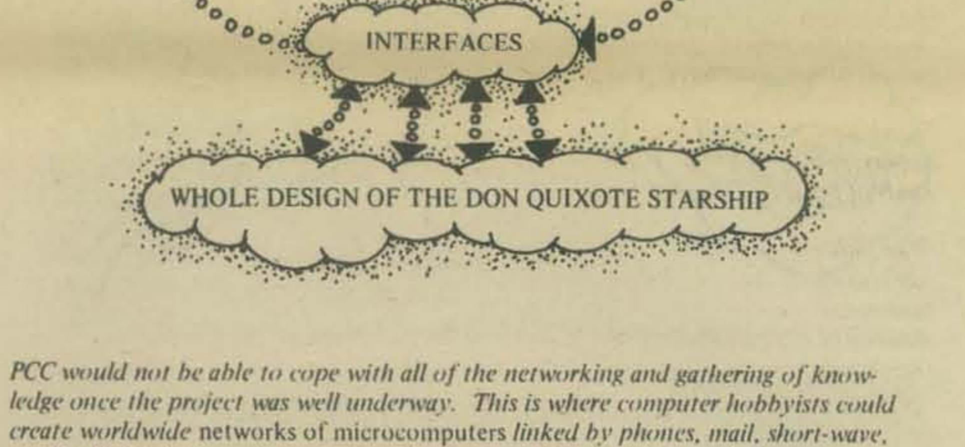
Resources cannot act autonomously, they only interact according to well-structured laws in an uniform manner. They do nothing to increase the order or predictability of the world whereas a certain degree of the latter is required by the orgauts (life). Resources generally have renewable cycles, from the very long to the very short. Since the orgauts require them for energy/mobility/nourishment/exploration, their ecological interdependencies will have to take these cycle times into their economic accounting. Resources can be totally wiped out if the conditions for their generation have been deteriorated (e.g. solar radiation level dropped, or the loss of cultural knowledge when a species dies without having established inter-species communication.) Depending on how one scales their view of the interdependent webs of orgauts and resources the levels of resulting whole systems constitute autonomous organisms in themselves.

The Processes of Nature are essentially the ways by which the components of the game can interact with each other and evolve through time. The possible interactions are a reflection of the interconnection of things . . . the warp and woof of the world. . . "everything is deeply intertwingled". Chance, always an important component to most gaming should appear to be present; but ultimately, I believe it should only be a reflection of the deep intertwingling . . . the microcosm within microcosmic complexity of the world. There could be the breaking of pseudo-random codes that predict the form of other pseudo-random codes . . . etc. What appears as chance events throughour linear-chain-causality spectacles could really be reflections of a vaster, synchronistic or jig-saw puzzle causality that requires the pattern recognition and twingle-logic-intuition of our right cortical hemispheres in order to perceive. The workings of the levels within levels of the vast cosmic cellular array.

Dynamics of Game Design

Creating a game is like designing a simulation, you are creating a model of some world (often confused with reality, wherever that is). The langugage (components) and semantics (dynamics) of your model description language specify what you perceive that world to be. Creating a vast exploration like the Quixote Starship has both the elements of world creation and world simulation; we should therefore be careful not to put blinkers on our vision of reality, especially when there are such incredible unknowns to encounter, such as the nature of alien intelligence, the ecology of alien worlds and stellar systems that are multiple-sunned or star-clustered, and subtle synchronistic laws of the universe. We should therefore encourage tremendous variety and ranges of alternatives in the game; variety in logic, language, universal and local laws, social groupings, networking, and approaches to the exploration of the universe.

A quest of the complexity of QUIXOTE will require many moons or years of design. There will have to be groups of folks with wide differences in expertise and wisdom exchanging, interacting, gathering knowledge about the various components of this endless expedition. Vast overlapping nets preparing to seed the stars. We will need media that will disseminate the activities and integrate the interconnections of these groups. . . a journal such as the Tralfamadore Times that you suggested. There should be several cycles of setting up problems, acquiring and distributing feedback, gestation periods, and setting the phases of the game (see the Lottery in Babylon). Who, then, and how often will this be done? Do we need some secrecy in setting final designs; after all, uncertainty is a prime requisite for the enjoyment of gameplay?



PCC would not be able to cope with all of the networking and gathering of knowledge once the project was well underway. This is where computer hobbyists could create worldwide networks of microcomputers linked by phones, mail, short-wave, or CB. Hobbyists could creatively solve the operational problems of these nets; their protocols, their teleprocessing, the creation of large distributed data-bases for the game. A vast reticulation of knowhow could spread across a whole earth in a World Community Memory. The pulsing cortex of Earth intelligence.

Would such networking subtly and effectively create an alternative and somewhat invisible college (see Teg's 1994) . . . an alternative to the production-oriented, authority-mongering, creativity and imagination-stifling current institutions of learning? Can a free and open network for knowledge and widom exchange make the inflexible and authoritarian institutions of our culture irrelevant?

The Times should facilitate and coordinate the formation of subnets of design activity and game play. Nets with random memberships could be created to percolate novel synergistic effects. Nets could be named after the fantasy lands and worlds of science-fiction. I would like to be in a net concerned with alien communication; call it the Earthsea net.

Feed-back-forward

We need feedback and feedforward from all those who want to contribute their imaginations and dreams to the design and/or the play of the DON QUIXOTE STARSHIP game. There are numerous areas where special knowledge and imaginations are required. We need the inspiration of those who have been engaged with space-colony design, like Johnnie S., and Gerald O'Neill, and whole-systems distributor Stewart Brand. We need the involvement of artists, humanists, spirits, iconoclasts, crazies and scientists.

Let us nurture a vast and loving gathering of the adventurous spirit of the Earth!

THE DON QUIXOTE STARSHIP ...is trembling on the pad !

Here're some of our ideas: we'd like your reactions to the form, content and whatever else you care to comment on.

We propose: The Don Quixote Starship

Purpose: **To facilitate participation in man's advent into outer space and to foster humane, diversified, and creative approaches to that process.**

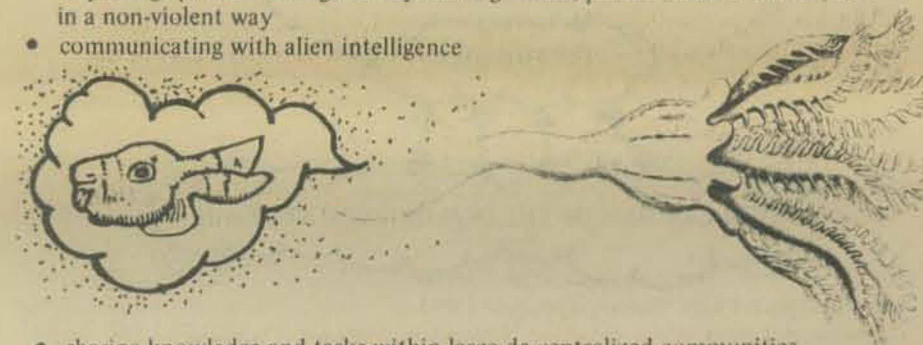
Activities: **Expand consciousness of space exploration with computer games and simulations, stressing cooperation, collecting and distributing information, etc. etc. etc.**

INTRODUCTION

Successful and humane space exploration depends on right technology and right attitudes — — and its the latter area, perhaps, in which we're most lacking. Violent space war and violent Star Trek games proliferate at a rapid rate, even though killing Klingons was *not* the message of the *Star Trek* TV series. These, as with most space games, stress violence and competition instead of the cooperative non-violent approaches we so sorely need.

The Don Quixote Starship (DQS) will stress the development needed for space exploration from as many viewpoints as we can nuster. We see DQS as a vehicle for developing cooperative games and providing opportunities for kids and adults to access science communications and space technology. . . and have fun! PCC will serve as a forum for this journey. We encourage you to

1) design cooperative space games (we'll publish suggestions as they come in) particularly involving:
   - building a starship
   - exploring space including solar systems, galaxies, planets, alien societies, etc. in a non-violent way
   - communicating with alien intelligence



   - sharing knowledge and tasks within large de-centralized communities

2) build resources for DQS
   - specify protocols for de-centralized (distributed) micro computer networks amongst hobbyists
   - build software for maintaining distributed data bases
   - we need systems for inputting games, data bases, etc.
   - we need people to act as clearing house(s) to create coherency from the input so that the information can be usefully accessed
   - Data bases need building (perhaps one real and one from science fiction/ personal invention?) that will describe star systems, planet forms, life forms, flora, fauna, societal structures, problems to solve in space, etc. etc.

How many ways can these games/resources be made available? Lots and lots and lots of ways are needed if we're to help create an environment in which cooperative exploration can thrive. Here are our first-pass suggestions:

- **micro computers** at home, school, etc. can serve one or more people for some games, including perhaps, building starships from "warehouses" of "parts" while overcoming "obstacles."
- use existing **timesharing networks** for multi-player on-line games.
- **cable TV** plus phone link
- **play-by-mail games**: such games are already happening; see information on Flying Buffalo in the next section of this article
- **ham radio** (networks?)
- **PISA** (Public Interest Satellite Association — — see article this issue)
- out-of-body space explorations (Psychics have performed out-of-body pre-Viking explorations of Mars; their reports are now being compared with recent NASA data)
- ?? FOR YOU TO COMPLETE.....

POSSIBLE RESOURCES

Here're a few ideas to get you started: we want your suggestions to extend these lists:

Ideas for designing games

Note on conflict & competition: These sorts of things are intrinsic in the dynamic of the world — further, they are relative to an observer's or participant's value systems — the trampled grass on a medieval field of battle might regard *both* sides of the conflict as equal entities — as mortally dangerous to individuals (blades of grass) and as a good source of fertilizer (the casualties) for the survivors. Clearly the humans involved will view it quite differently.

Games to be interesting *require* a dynamic effort expended and must somehow relate to benefit received — otherwise why play? Mostly, I think, the problem is with the sorts of *fantasies* we create to provide motivation for playing a given game — this is certainly an area which could use a little creativity and a little taste.

Consider this: simulating, hunting and killing an imaginary Wumpus gives someone direct contact with the predatory aspects of our nature (and prey aspects too — Wumpi eat back!) Eating MacDonald's hamburgers causatively involves *real* killing of *real* animals but is experienced in such a way as to disassociate us from that reality, to provide hallucinations of how we would *like* the world to be, as opposed to how it *is.*

Now here's a tough, sophomoric (yes, that's what I mean) question: which activity makes us wholer, in the sense of reducing our schizophrenia , hunting the Wumpus or grabbing a quick bite to eat at MacDonald's?

Enough of this — — on to specific suggestions as to where to get ideas for games which could be either used directly or adapted for DQS.

- See the Aug.-Sept. *PCC* (Vol. 5 No. 2) for several ideas on subject matter for games (page 6 "Science fiction Games") and on designing the programs (page 7 "Thoughts on St*r Tr*kking and Space Gaming").
- explore commercial science fiction and other board games; mostly violent/ competitive, but maybe they'll give you ideas
- Flying Buffalo, Inc. is a small company involved in a variety of play-by-mail games (see *PCC* Vol. 4 No. 4, January '76). Flying Buffalo's Favorite Magazine is published by Flying Buffalo Computer Conflict-Simulation, Inc., for $4/year, for 6 copies, or $1/ issue. Or write

  Flying Buffalo
  P. O. Box 1467
  Scottsdale, AZ 85252

  We suggest you get some experience in play-by-mail games by signing up for Flying Buffalo's STARWEB game.

- *Gamesemag:* sample copies 50 cents; 9 issue subscription $4. From

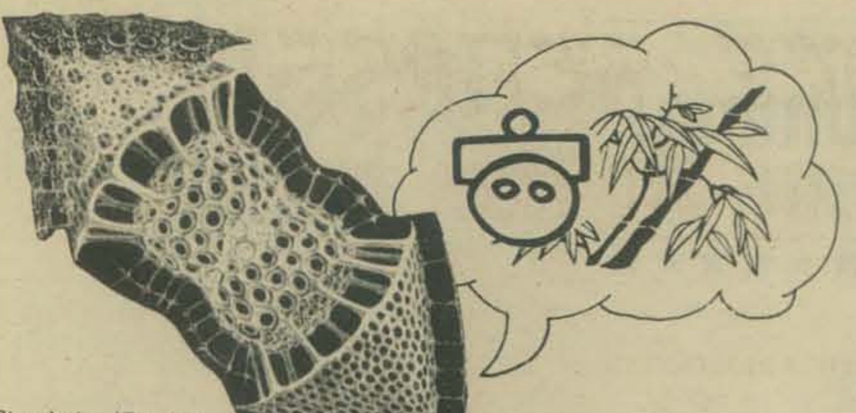
  The Center for Open Learning and Teaching
  P. O. Box 9434
  Berkeley, CA 94709

- Huntington II: computer simulation games; the "policy" game is of especial interest. Huntington materials are available from

  Digital Equipment Corporation
  Software Distribution Center
  146 Main Street
  Maynard, MA 01754

- Free literature on school-oriented simulation games "Elementary Simulation and Gaming Catalog" and "Jr. High Through College — — Social Studies Catalog featuring Simulations and Games" available from

  emi
  Box 4272 – G
  Madison, WI 53711



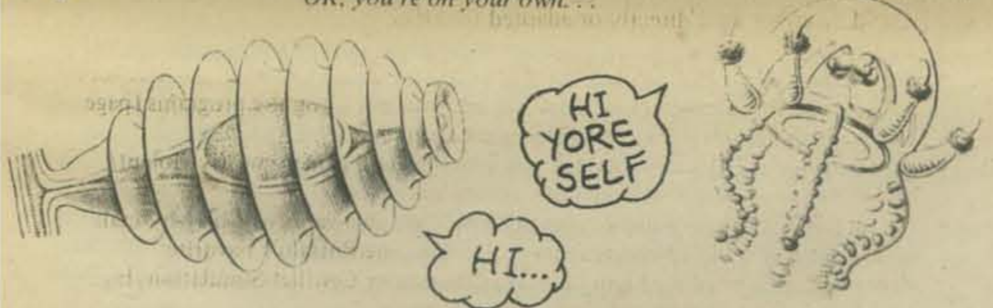
- Simulation/Gaming
  Box 3039
  University Station
  Moscow, ID 83843

- "Dungeon and Dragon" game mags – see 6 of them on page 42 of the last *PCC*, Vol. 5 No. 3.-
- We've heard that a very interesting space game exists on the Plato system, which sounds like it's worth checking out. We believe it involves a cube of space 100 light-years on a side. There's a whole bunch of star systems, with definite locations, of several types (all supposedly physically correct) such as single suns, binarys, red giants, white dwarfs and (of course) black holes. You start off somewhere in the form of a civilization. You gather resources, build space ships, explore, exploit other systems, etc. After a while of course, you run into other players (civilizations, races). What happens then depends . . . maybe war, maybe alliance.

  The game universe is always available on the system, players enter and leave as they wish (this is really an indefinitely long simulation) empires grow, consolidate, die, etc.

  Plato allows for this kind of thing, because it's a big, distributed system. They also have nation-wide dogfight (airplane) games of considerable sophistication – complete with displays of yours and their planes, landing fields, etc. (in perspective).

- New Games: "play hard, play fair, nobody hurt; games for lots of people to play cooperatively"; paperback; $4.95; Doubleday/Dolphin
- *OK, you're on your own. . .*



SPACEY AND/OR FUTURISTIC IDEAS

- Articles in *CoEvolution Quarterly* from Fall 1975 through the present provide provocative articles ranging over many subjects. Fall 1975 contains Gerard O'Neill's testimony before a subcommittee on space science and applications of the U. S. House of Representatives' committee on science and technology on July 23, 1975. Cost estimates, power sources, current research are a few of the topics discussed. O'Neill was also interviewed for that issue. Subsequent issues contain reactions to O'Neill's space colonies from many individuals, including R. Buckmaster Fuller, Paolo Soleri and Astronaut Russell Schweickart. Find the magazine at $2.50/issue at "organic" bookstores or P. O. Box 428, Sausalito, CA 94965.

- Gerard K. O'Neill, *The High Frontier: Colonies in Space*; $8.95; Morrow

- *L5 News*: "latest developments in space industrialization, satellite solar power, and space habitats;" $1/issue when available; subscription prices to non-members available upon request from

  L-5 Society
  1620 N. Park
  Tucson, AZ 85719

- *Earth/Space News*, a practical, technically oriented newsletter involving people actively planning the technological/human factors/financing of building a spaceship. Subscriptions are $10 yr; bimonthly. from

  Earth/Space Inc.
  4151 Middlefield Road
  Palo Alto, CA 94303

- Carl Sagan, editor, *Communications with Extraterrestrial Intelligence: CETI*: proceedings of a joint U. S. - Russia Conference on the subject, and contains many technical studies; paperback; $6.95; MIT Press

- The Foundation Institute, 85 E. Geranium Ave., St. Paul, MN 55117 is concerned with advanced scientific and engineering studies, with a primary thrust in astronautics, particularly in the utilization of outer space and the need for economical space transportation.

- *Communications Tomorrow*, from the Special Studies Division of the World Future Society; bimonthly publication; $9 yr. ($6 yr. to members)

  World Future Society
  P. O. Box 30369
  Bethesda Station
  Washington, D. C. 20014

- membership in the World Future Society (see above address) is $12 per year and includes a subscription to the Society's magazine, *THE FUTURIST: A Journal of Forecasts, Trends, and Ideas About the Future*

- *Speculative Anthropology* is a forum on cultural futuristics and space technology; it's published in September, December, March and June for $3/yr. For information, a sample copy, to subscribe or submit articles for consideration, write to

  Carol J. Motts
  Speculative Anthropology
  10151 Heather Court
  Westminster, CA 92683

- Nova Enterprises
  P. O. Box 149
  Parkville Station
  Brooklyn NY 11204
  Send 25 cents for a catalog of scifi and fantasy games and books.

- NASA has lots of excellent, free materials. Send for publication lists, film lists and information about other services from the Educational Office at the NASA Center serving your state.

  NASA Ames Research Center
  Moffett Field, California 94035
  Alaska, Arizona, California, Hawaii, Idaho, Montana, Nevada, Oregon, Utah, Washington, Wyoming

  NASA George C. Marshall Space Flight Center
  Marshall Space Flight Center, Alabama 35812
  Alabama, Arkansas, Iowa, Louisana, Mississippi, Missouri, Tennessee

  NASA Goddard Space Flight Center
  Greenbelt, Maryland 20771
  Connecticut, Delaware, District of Columbia, Maine, Maryland, Massachusetts, New Hampshire, New Jersey, New York, Pennsylvania, Rhode Island, Vermont

  NASA John F. Kennedy Space Center
  Kennedy Space Center, Florida 32899
  Florida, Georgia, Puerto Rico, Virgin Islands

  NASA Langley Research Center
  Langley Station, Hampton, Virginia 23365
  Kentucky, N. Carolina, S. Carolina, Virginia, West Virginia

  NASA Lewis Research Center
  21000 Brookpark Road, Cleveland, Ohio 44135
  Illinois, Indiana, Michigan, Minnesota, Ohio, Wisconsin

  NASA Lyndon B. Johnson Space Center
  Houston, Texas 77058
  Colorado, Kansas, Nebraska, New Mexico, N. Dakota, Oklahoma, S. Dakota, Texas

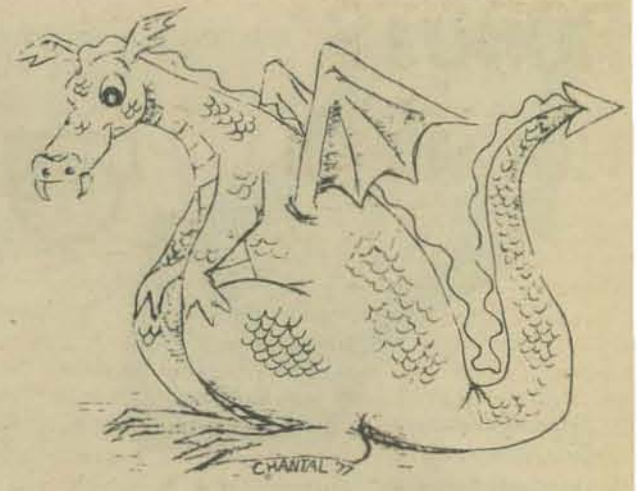
Editorial Contact:

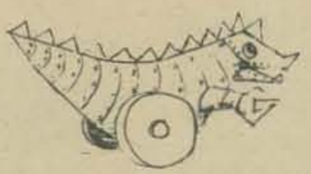
for direct communication to the editor on Don Quixote Starship material write to our alien correspondent:

Doug Seeley
833 W 18th Ave.
Vancouver, B. C.
Canada V52 1W2

The ROBOTS are Coming!

A few words from the United States Robotics Society





Dear Dragon,

Thanks for the invitation to say a few words about USRS in your pages. We weren't quite ready for it, but perhaps the material enclosed will be what you hope for.

There's very little to say about the United States Robotics Society at the moment, because the organization is new and comparatively small. We've been a bit taken aback by the response to our first real public exposure and are working diligently, not just to respond properly to our correspondents, but to prepare material of some value for distribution.

USRS was formed in the summer of 1975, when a handful of people who had been working in artificial intelligence and related fields realized that robotics had passed a milestone with the advent of the personal computer. Only within the past year has popular knowledge and appreciation of logic systems become so firmly established at the popular level that it is impossible to bottle up robotics activity by law.

The notion has been widespread for years that robotics is vaguely possible, but is an exotic activity dependent upon big-domed scientists working in vast research and development facilities with NASA-style budgets. The notion is mistaken, of course.

The founders of USRS believed that all information necessary to the implementation of machines with the characteristics of animals has already been developed, if not clearly recognized. The information is scattered about the world, lying in files both open and secret. All the thoughtful technical amateur needs to build satisfactory robotics systems is the technology available to him already, over the counter, and access to some of the various insights various workers have had with respect to robotics. He doesn't need NASA, Peenemunde, big budgets, or permission from a committee of experts.

Robots are inevitable. Unless we learn how to cope with robots—our own and those developed by people who mean to do us harm—we are in some danger of surprise. Dr. Richard Reuss commented as long ago as 1962: "If some tribe of Ubangis develops robots, and we don't, we're in serious trouble." No society has a monopoly on ideas and talent. Robots are on our doorstep, and some few of us may enjoy the experience of getting to know them.

USRS was formally established as a legal, non-profit corporation to gather, collate and disseminate information about robotics to those interested in the subject. The Society has few resources with which to begin the work of setting up communications channels among members, establishing a library, and generating useful activity that would give members a common effort in which to join.

Personal Computing Magazine* *was interested in our effort, and since the editorial office of that publication is here in Albuquerque where we coul wave our arms and talk persuasively face to face, agreed to publish an advertisement for the Society.*

Frankly, we didn't know whether there were as many as five more people in the country who agreed with our premises and thought the activity would be worthwhile. Somewhat to our surprise, memberships have begun to flow in at a modest, but steady rate, even from Europe, as well as from widely scattered portions of the United States. Obviously, this is not everyone's cup of tea, especially since our views on robotics are not sensational. We are dealing rather quietly with practical information, not chiefly with flights of fancy.

The membership has already grown to the point at which we feel the need to publish newsletters (on a modest scale, every two months, to begin with) and bulletins containing longer articles of specific interest to the membership. We expect the bulletins to alternate with the newsletters, so members will be receiving items of interest roughly monthly.

We are soliciting help from members in collecting clippings, papers, references that will permit development of a useful library. In time, as we have the resources, USRS will apply for a federal tax exemption, so that individuals and corporations will be able more readily to donate computer equipment for use in our own information processing. It's easier to put a computer to work in a brand new library than it is to adapt an existing library to computer treatment, so we have hope of accomplishing this fairly soon.

We are also generating a series of articles, either for internal publication or for release to other publications (like PCC*) that will, we hope, provide a framework in which robotics can be understood at a practical level. One such article, treating robots as household pets, is enclosed for your consideration. Our only reservation is that you publish the copyright notice with the article so that we may reserve the rights to USRS.*

A similar article will appear in the March/April issue of Personal Computing, *treating the establishment of a technical sport in which remotely controlled machines contend on a playing field. We feel that USRS sponsorship of the activity would provide us with a vast amount of practical information, so we are inviting our membership to participate in creation of the sport itself, the rules, regulations, traditions, customs, mores—everything involved in a full-fledged sport. We believe that this is the sort of creative effort that can give cohesiveness to a small, voluntary organization.*

We expect, of course, to develop robotics kits (the household pet would be a sensible place to start) that could be provided to members at a worthwhile discount. We are already offering some books of interest at a discount and doing those ordinary things that will gain financial support for the organization.

USRS is patterned after the National Geographic Society, for example, with a self-perpetuating board of directors charged to carry out the organization's objectives under the bylaws. Information developed by the Society must be made public and activities must meet the standards imposed for non-profit organizations. However, there will be no political activities among the members, no election of boards, officers, and leaders. If the organization fails in its objectives, the membership will diminish to a point at which operation cannot be maintained and activity will be terminated.

At the moment, however, membership is growing. It's easy to join. A $12 check to the United States Robotics Society with a request for membership does the job. Ideally, of course, members will wish to participate actively in the Society's work, though activity is not required.

That's where we stand, at least until we work our way through the mail that is already in hand. You're welcome to publish any or all of this in PCC. *Hope you find the enclosed article useful as well.*

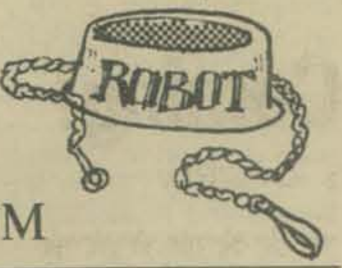
Sincerely,

GLENN R. NORRIS
President
United States Robotics Society
Box 26484
Albuquerque, NM 87102

* Personal Computing
401 Louisiana SE
Albuquerque NM 87108

Many thanks to Chantal for creating and contributing the fine dragon and robot on this page.

ROBOTS as HOUSEHOLD PETS



by ROBERT ROSSUM

Obviously, some decades must pass before we can expect any help from robots in driving buses, running city government, leading rescue teams to downed aircraft, and settling international disputes. In the meantime, more modest and, frankly, more interesting, applications of robotics are possible. Robots are usually imagined first as domestic servants that will do household chores without complaining, but such capability calls for feats of mechanics and judgement now far beyond our practical resources. (We'll discuss robots-as-domestics in another article.) We can, however, build robots in the form of household pets to which people may become fondly attached. Although some people will be repelled by these creatures, that need not discourage us from pursuing robotics; many people are repelled by dogs and cats.

The first step is easy—we speculate on the characteristics of household pets and the practicality of building these characteristics into robots.

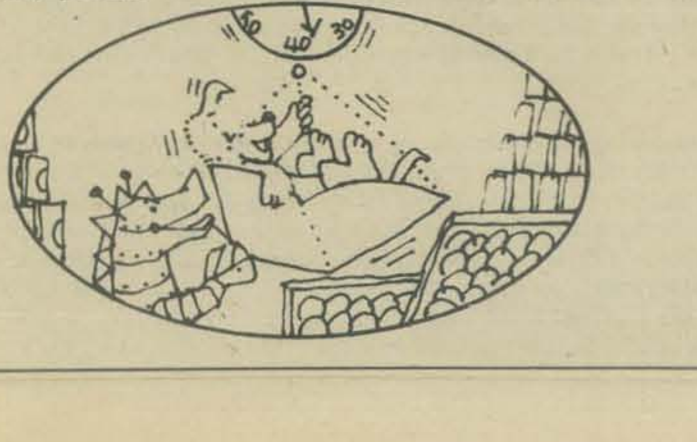
Consider cost. Assume that we're not interested in building just one or two toys for zealous rich people, but of creating a product that can be distributed fairly widely. What should it cost? Well, what does a pet cost? There's a range, to be sure, running from mutts that can be ransomed for a dollar from the dog pound up to rare, specialized pooches that have been developed over many generations of careful breeding, valued at several thousand dollars. Notice that the difference is not in the flesh, bone and sinew, but in the *information* that goes with the dog. In general, the price rises with the sophistication, the information content, of any item being traded.

Size matters, too. Indeed, we have access to a study that allows us to gauge the probable acceptable price of any given item in our society, given its weight and sophistication. Notice that a microcomputer with a fair amount of memory, power supply, cabinet, and small keyboard weights fifteen to twenty pounds and sells for about a thousand dollars. You can buy twenty pounds of bread for less than that.

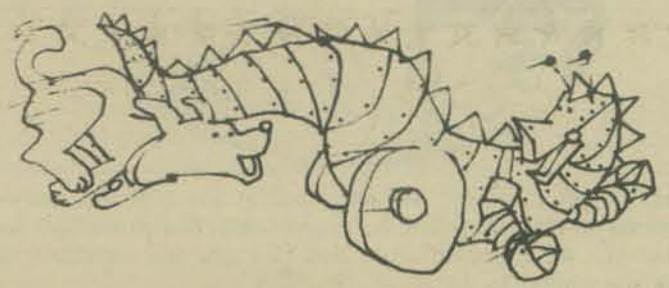
Our pet robot can't be picked up for a buck at the Robot Pound yet and it's obviously as sophisticated as a microcomputer. Its price, then, is related to its weight also. How much should it weigh?

Since this pet is to live in the house, we must consider the weight of things that commonly live in houses. People live in houses. From the time they are first mobile to the time they are grown, they weigh something between about fifteen and three hundred pounds. Their weight is distributed over a fairly large area on their big feet (and their hands and knees when they're small) so no terribly great forces are brought to bear on small points, even when they stamp their feet or jump. Mobile household pets share these characteristics.

Few indoor pets weigh more than a nundred pounds. Few have such sharp, pointed hooves that they mar the floor. (Ladies with spiked heels are often encouraged to take their shoes off to avoid damage to hardwood floors and carpets.) Though some German Shepards live inside, most acceptable household pets are smaller, ranging from fifteen to perhaps fifty pounds.



So, judging from floor-loading characteristics and common practice, we're shooting for a robot pet that weighs twenty-five to forty pounds. It's sophisticated, new, rare, but in detail not completely novel. Looking at our chart that plots the log of product weight against the log of price in dollars-per-pound for things in our society, such a product as ours should cost around a thousand dollars. If it sold for ten times that much, it would have little practical chance of popularity. If it sold for a tenth of that, it would have little practical chance of being manufactured.



After cost and weight, what about velocity? Things that live around the house seldom move more than five miles per hour. If a little kid gets up to that speed, somebody shouts "Quit that running in the house!" If the dog moves that fast, he's chucked outside and if the cat does it, he's assumed to be having a crazy spell and he's fended off with a broom. 4 mph is a very rapid walk for a man.

A pet that moves only slower than .5 mph probably can't move fast enough to play with the kids. If it moves faster than 2 mph, it's considered too frisky. So, .5 to 2 mph for the robot household pet may be a good range.

What can the pet do physically? It had better not do very much. Monkeys with grabby hands are seldom allowed to roam homes freely. Dogs and cats are not encouraged to meddle with the things that people use. If they push open a door that's ajar or pick up a newspaper when they're told, that's plenty. No fondling of the dishes, the furniture, or guests is acceptable. People don't want their pets to be capable of taking liberties with valued items.

If the pet robot can move about, that's probably enough. Refinements like the ability to climb stairs or to pick up small objects from the floor are probably desirable options, but that's a matter of important choice that is best left to the individual pet owner.

What about information content? Since we've backed light years away from the objective of simulating human behavior, we're dealing with a system that's just about comprehensible in state of the art technology. We can distinguish usefully here between structure and logic. That is, the information content of a stone ax, for example, lies completely in its physical structure. It is good for some things, not good for others, and not at all adaptable. You can't make a silk purse out of a stone ax. Everything it "knows" is determined at the moment of its completion.

More sophisticated machines, however, are equipped with some logic, with structure or circuitry that "learns" from the environment in which it operates. An automatic record changer, for example, adapts to the size of the records it is handling, one at a time, and may be equipped with elaborate sensory and judgemental apparatus to vary its performance greatly with varying circumstances.

Our pet robot must have certain sensory and response capabilities as part of its basic structure. Certain pattern recognition capabilities are part of the structure of living things: a frog's eye (even when it is disconnected from the frog's brain) puts out a jolt of signal when something that looks like a fly crosses its field at a distance of one frog-tongue. That's the way the system is built. A human baby fifteen minutes old, even before it has learned to "see" will turn to a crude portrayal of a human visage, away from other forms presented at the same time. That's the way the baby is built.

The pet robot, like the pet dog or cat, will have some such structural basic characteristics. What we add to that in the way of "logic," the ability to learn from experience, is a matter of taste. Perhaps we should remark early and often that the key to maintaining a healthy relationship with the pet is to keep total control of its learning. You don't want somebody teaching your parrot to say things that will offend you. You don't want your pet robot to learn habits that annoy or frighten you. The pet owner may have a key to the switch and turns the learning circuitry on and off and he may want to wear it like a dog tag around his own neck.

Of course, the pet robot can be "retrained" at any time, his memory utterly erased so that everything he ever learned, bad habits and good, are eliminated. We'd hesitate to do that to a dog, but we don't mind rebuilding a machine, do we?

Without dwelling on this in a brief speculation we may note that intelligence in animals seems to be related to the number of input channels to the brain, compared with the number of "thinking" cells in the brain—that is, the ratio of nerve inputs to the number of neurons. An ant, which can learn just a little, but relies mostly on structural information for his operations, has a ratio of about one to one. In a rat, the ratio is about eighteen to one. In a human being, the ratio is about five hundred to one. What should it be in the pet robot? What *can* it be? There's room for a lot of discussion and experimentation here.

What must the pet robot *be like* to find a welcome in the home? Again, what's welcome now?

Mammals are all roughly spherical in shape. When engineers calculate heat flow and the like for mammals, they use a spherical model without being too far off. The pet robot may want to be roundish, with a density of about 1, the density of water. Most of a mammal's weight *is* water, which weighs 64 pounds per cubic foot. It may be difficult to package a forty-pound pet in a cubic foot, but remember we may reasonably be off by a factor of two or three without exceeding practical tolerances.

All baby mammals have large eyes and high foreheads. People regard little critters with these characteristics as desirable. Poodles are ordinarily trimmed to exaggerate the big eyes and high foreheads. Probably owls are the most popular artistic subjects among birds because of the mammalian appearance. All baby mammals are cuddly and non-threatening and people feel at ease with them, while adults make us all uneasy until we get to know them. Narrow-eyed bull terriers are "mean looking" dogs. Surely, the pet robot must have big, baby eyes and a high forehead to be welcome. Dark eyes, too. Why? Angry, agitated people ordinarily have the pupils of their eyes stopped down to small apertures. Like the bull terrier, they look dangerous. Relaxed people, at ease with their surroundings, have large, dark pupils (except, of course, in harsh light) that put others around them at ease. Dark eyes are a sign of benign tranquility in the ordinary world. To put people at ease, the pet robot wants large, dark eyes.

The pet robot must look like a friendly mammal. It should also smell like a friendly mammal . . . or anyway, like something of which people ordinarily approve . . . new car smell? Musk? A soft pine smell bringing to mind the furry forest creatures? Experiment needed.

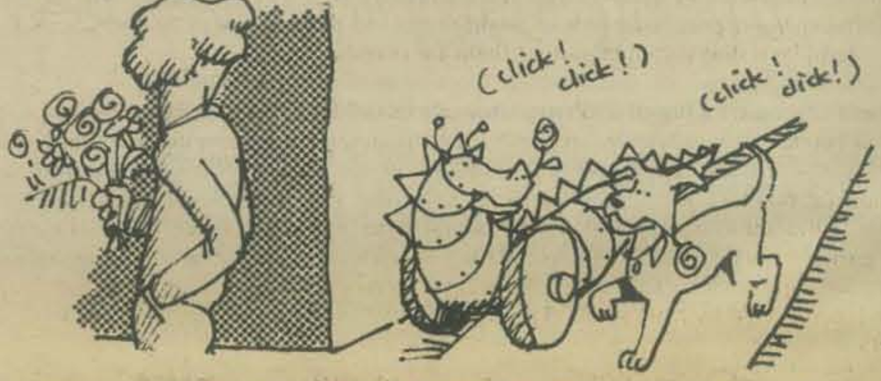
The pet robot should taste "right," not metallic, not like paint. You know that small children will get a lick of the thing once in a while and the taste should be neither attractive, nor offensive, though maybe faintly salty. You don't want child or dog to be slurping the thing constantly. Bad for all concerned. Children are typically discouraged from licking the cat and vice versa.

The pet robot should "feel right," probably warm (98.6°F would be a sensible internal temperature with an appropriately corresponding skin temperature.) and soft, not necessarily furry. Bare rat tails are not universally admired by people seeking pets.

People also like to know which way is up on their pets. Commonly, animals are light underneath, so things looking up under them mistake them for the sky, and dark on top, so that things looking down mistake them for the ground. The arrangement of colors varies, but among mammals, earthy tones predominate. Vivid reds and blues are uncommon. The combination of color and shape should reveal to small child and parent alike just where the "strong back" of the pet is. Obviously, small children will try to ride the pet robot, as they always try to ride dogs and cats. The dogs and cats ordinarily resist this pleasantry by squirming away. (In every generation, the dogs that bite small children get their brains knocked out, selecting for smarter and more tractable animals.) The robot need not squirm away, but people who disapprove of seeing kids try to sit on the stomach of the dog will want to be assured that they are not sitting on the stomach of the pet robot. That just wouldn't be right.

And – importantly, the pet robot must *sound* right. Strident, systematic sounds are unsettling. Threatening creatures throughout man's long history have always made purposeful, non-random sounds and our ancestors (who were the people who survived stalking sabre-toothed tigers, etcetera, long enough to have children) learned to spring to alertness when non-random sounds occurred. Babbling brooks, splashing surf, the wind in the branches, are relaxing. Footfalls and cyclic sounds are unnerving. The pet robot should sound random, purring, chortling, when it is not doing anything significant, and quietly systematic when it is doing something.

If the pet robot is trundling down the hall after you, it should let you know it is there. It has no nails (well, maybe it has) to make a sound on the floor, as a dog does, but maybe it makes a jolly soft clicking noise of self-importance while it's moving, so you won't stumble over it or be taken by surprise. In this day of high-quality audio systems, the sound of the pet robot is outstandingly controllable.



So, appearance, aroma, flavor, warmth, orientation, and sound are all significant factors in the pet robot. Presumably, some of these factors are variables that may be altered by the pet robot's environment . . . leading us to behavior. What might the pet robot do?

The household pet is much admired when he curls up in the sun and goes to sleep. A sleeping pet doesn't look dead, lifeless. He looks like a live, sleeping animal. He breathes, even snores, chuckles, twitches a bit, shifts his position occasionally, often following a moving sunbeam across the floor as the earth turns. It should be no problem to design the pet robot to do these things in the same, reassuring fashion.

People learned many thousands of years ago that when the family meathound hears something interesting, even when he's asleep, his ears stand up. Many a man has been warned of danger by the alert ear responses of a sleeping pet. Should the sleeping robot be less alert?

Indeed, since the human brain seems to be designed to wake up when something non-random occurs and we have considered behavior in the robot that affects that human characteristic, can we not put the same characteristics in the pet robot? Shouldn't it waken when something non-random occurs around him? There are many different levels of activity in our ordinary environment that we learn to judge by experience. A hubbub of playing children may not disturb us, while the very subdued sound of something breaking or learning may instantly attract our full attention right in the middle of all the fuss. We have, built-in, certain kinds of priorities. Why not build such priorities into the pet robot?

His clock may tell him, for example, that he should be quiet at night, again, not dead, but quiet. However, if non-random activity breaks out in the house, he should "investigate." That is, he should go where the action is. That's what pets are supposed to do.

Have you ever realized, while decorating a Christmas tree, that you have been stepping over the dog repeatedly without thinking about it? There he is, the dopey mutt, right in the middle of the action, happily watching all that does on and nibbling interestedly on the strange things that come out of the boxes. He's doing what he is supposed to do, a thing that is enormously appealing about pets. He is being interested in his master's activities, being a cheerful pal. The dog always gets up and goes to the kid who cries in the night, to the grandmother who is more restless and uneasy than usual – making non-random signals. The pet goes where the action is.

When there's a new baby in a household, one of the more endearing things an older, but still very young, child does is to listen very closely to what the baby is doing and then report vigorously when the baby cries. It's conventional for a little kid to run through the house, saying: "The baby is crying, Mom, the baby is crying," when Mom already knows. Why not the pet robot?

Different households have different needs, but people with tiny kids are always pleased to see the family dog or cat hovering protectively and kindly around the babies and small kids.

Notice that babies are commonly surrounded by a mild aroma of sour milk. How difficult is it to build a sour-milk detector into the pet robot so that it constantly seeks the baby? Some override will prevent the pet from climbing on the kid, but the robot will hang around the baby, sleeping, listening, shuffling about. Then, when the baby cries, the pet robot may run through the house (at no more than 2 mph) saying, in a charming little voice (suited to his Dwight Eisenhower-Hubert Humphrey baby face): "The baby is crying, someone come, the baby is crying." Is sour milk an inadequate lure? Surely the baby may be sought by other means.

HURRY UP SOMEONE!
where is everybody anyway?
THE BABY IS CRYING, SOMEONE COME!
hey, the baby is crying...
The BABY is CRYING!
The baby is crying!

It's entirely possible that the pet robot can learn to identify members of its own family (inexpensive enzymatic sensors might serve) and to be ill-at-ease with strangers. . . until he is taught that certain of them are friends.

Imagine that you are a burglar. (Presumably our readership includes a proportionate share of burglars.) You have entered an apparantly unattended house and are rummaging around for the cameras when something you can't quite identify hastens into the room (at no more than 2 mph) and announces in a slightly hysterical voice: "I think it's only fair to warn you that there are rattlesnakes loose in here!" Then the thing groans miserably and loudly, like Marley's Ghost, and settles down to following your moves with its sensors. From time to time, a rattlesnake rattle is heard in the room, accompanied by little gasps. This might inhibit you from reaching into dark corners

Of course, the pet robot might have a considerable repertoire of such activities, designed to speed the parting guest. It might even control secondary systems that make sounds, switch on lights, and so on, in some random and unpleasant fashion. No violence. No threat of violence, merely uncertainty.

When the pet robot is not trying to be menacing, it may even try to be very friendly. When a dog pants and lets his tongue loll out, especially, while wagging his tail, people take a shine to the cheerful beast and reach out to scratch his ears and pet him. Probably, this rises from tens of thousands of years of observation that a dog in this configuration is probably not all tensed up to bite somebody and that's a relief in itself. Can the pet robot give such signals? Of course.

The pet robot will be acceptable if it does things that are appropriate to changing situations, but not entirely predictable. The behavior of a household pet is predictable *in general*, but not in detail. It is slightly random, surprising, revealing of the personality of the pet. Publicity or no, pet rocks are not really pets.

It is bootless to speak of useful tasks the pet robot might perform – fire alarm, intrusion alarm, small tool carrier, paper fetching – until the pet-like characteristics that allow the creature to be loved are established. People do grow fond of machines. People become sentimentally attached to *old* cars whose increasingly random performance they have come to know. They are not *fond of* new cars.

We might dwell on this at great length, and may do so as time passes, but note just one more characteristic of pets: they depend upon their masters for care. No animal that hunts the countryside for his own food, who slinks away into a dark hole to care for his own hurts, is considered a *pet*. An independent animal of that sort may be an acquaintance, even sort of a friend, but he is not a pet. History suggests too strongly that a noble, independent, proud creature is dangerous, not to be trifled with, certainly not to be left with the baby, whom he may come to regard as a possibility for lunch.

Let the pet robot cry when it is hurt and come to Mom for aid. Let it grow more alert toward dinnertime, then more active, then positively importuning, softly whining and eager. Let the pet robot respond delightedly to the sound of the dinner bell, snorting deliriously when his battery charger is plugged in by the loving human hands.

The pet robot should come when he's called. (No great technical problem now.) He should go when somebody says "Go!" and stop when somebody says "Stay!" He should learn not to cross a barrier, so that he can be fenced into some area with no trouble.

No characteristic of pets discussed here (surely others have been omitted that are worth discussion) calls for any sort of technical breakthrough. All of this is within present technical capabilitities, within reasonable cost constraints.

The problem is not to make robots do something novel, but to let them do things that are familiar and welcome.

Build Your Own Working Robot : a review

Reprinted with permission of *CoEvolution Quarterly* (Box 428, Sausalito, CA 94965) from their Fall 1976 issue.

Several decades ago the neurophysiologist W. Grey Walter constructed a number of small robots, called "turtles," which possessed many animal-like behaviors. This book describes how to build a similar device, named "BUSTER," which carries on the tradition using modern digital electronics as opposed to the original vacuum tubes.

It's a do-it-yourself for-the-home-hobbyist style book, but the project involved is considerably more substantial and difficult than most of the genre. The use of discrete logic, rather than a compact micro-computer chip, complicates the construction but also insures that the design is not locked into using some particular manufacturer's product. The mechanical aspects are usually the most troublesome part of such an endeavor, and the shaken down methods given should be enormously helpful to the fledgling roboticist.

The book would be greatly improved by a little sensitive editing and better photographs. Most appropriate for someone who has just built a home computer and is looking for ideas for applications.

– *Marc Le Brun*
(Suggested by People's Computer Company)

BUILD YOUR OWN WORKING ROBOT

David L. Heiserman
1976 234 pp. $5.95
from the PCC Book Store

Full-function standby is an interesting mode for Buster whenever he is not in use. He is primed and ready to go at the touch of a switch and, of course, he will squeal for his dinner when the batteries run low.

From all outward appearances, the *full-function standby* mode is the same as the quiescent state. In full-function *standby*, however, Buster's reflexes are still operative. The blunder mode is operative, for instance, letting Buster respond to any object that touches his contact switch assemblies. Our family cat once approached Buster while in full-function standby. The cat's curiosity suddenly changed to terror, however, as he touched one of the contact switches and found himself being chased by a growling red monster. The chase lasted only about five seconds – until the blunder mode timed out – but it was enough to cure the cat of his curiosity about sleeping Busters.

SPACE & COMPUTERS — a fascinating match

Copyright, 1976, Earth/Space, Inc.
Reprinted with permission from
Earth/Space News,
November 1976

by JOHN PEERS

John Peers is President of an innovative computer company called Logical Machine Corporation. The company has developed a computer which requires no professional programmer middleman. The computer–called Adam–works in the English (or French or Polish or any other) language, uses ordinary nouns and verbs, and requires no previously set programming.

With Adam, the user controls the computer, rather than the computer restricting or changing the user's methods to fit what's acceptable to the computer. This is what's possible today.

Extending his vision of what might be, John Peers looks at a future in which the Computer and Space become entwined in a benevolent synergism, which could result in the knowledge of the World fitting in the palm of your hand.

Extending the use of computers by using the resources of Space is such a fascinating topic that I doubt if it is possible to cover even a small fraction of the possibilities arising from such a match in a short article. I am going to try to cover a few areas which I see as being available fairly soon. I have one proviso. These manifestations will probably only occur should free enterprise be allowed access to space.

For readers who are *not* in the computer business, I think a few words about the current state of computer technology might help.

First, the computational electronics. We have just begun the era of the micro-computer chip. A micro-computer chip is a single device, usually measuring less than a square inch, which performs either all, or the majority, of the functions that would have taken in about a hundred cubic feet of electronics as little as fifteen years ago. During the next five years the incredible micro-computer chips will become ever-faster, ever-smaller and ever-cheaper. I predict, therefore, that the price of actual computing power will collapse over the next decade, in a way similar to the hand-held calculator over the last seven years.

But computing is of little use, unless there is something with which to compute. This involves *storing* information in one form or another. There have been some very big strides made in computer storage capability. Particularly exciting are some of the Solid State storage devices, prototypes of which are under demonstration by IBM and Control Data (to name but two) and which *in prototype form* can hold perhaps one-half the written contents of the Encyclopedia Britannica. These storage devices are smaller than a domestic television set. The best estimate for the ultimate useful storage capability of these devices, within a five to ten year period, is that they should be able to contain the entire contents of a large library and still be physically the same size as a reasonable domestic television set. Approximately, of course. The user would be able to 'get' any book in about a thousandth of a second!

Computing power and huge, low-cost storage are of even less value unless these can be applied. Application needs software. (For those who are not aware of it, 'software' is the organization of a computer; 'hardware' is the equipment.) Thank goodness massive changes are occuring in software design as well as hardware. One recent development in software has been the design of a computer so that it can organize itself, which to a great extent eliminates the need for conventional computer-programmer training. This means the user can program his own requirements directly.

Today, it is becoming common to get a computer to 'talk back' to you. An obvious example is the banking systems to which you inform by 'touch-tone' telephone, and which conforms and replies, in voice. Even after 30 years of computers, however, we are still keyboard bound. Virtually every piece of information given to a computer must at some stage be keyed. Typewriter, card punch, key to tape, key to disc, direct VDU entry. They all require input using fingers. Fingers are slow, unreliable, and need training A much better source for a lot of uses, would be voice. Several companies, (NEC, IBM, Threshold Technology, for example) have voice recognition systems. I believe that over the next few years, conversational voice will be the natural, easy and inexpensive way to communicate with a computer. Instead of having to wait for a computer to print something out or display it on a television screen, the user soon will be able to communicate with the computer in voice. These two direct advances–direct programming and voice communication –coupled together suggest to me that during the next five to ten years, the act of programming will disappear. 'Programming' will be a function of use, not a separate occurance. This fits with my own basic theory that there is no difference between programming and data. They are simply the same thing looked at from two different points of view.

The next area of advancement has been communications between machines. Rapid strides are being made in communications and several manufacturers are making single or multiple chip communications logic devices. These will allow devices to talk to each other very inexpensively, using very little power. Needless to say, these chips are all extremely small.

Taking into consideration all of these developments in the computer field, it becomes clear that we are moving into an era where *everyone in the world* can have his *own* computer. Such a computer will be capable of storing huge amounts of information and communicating at high-speed with other computers. And the user will communicate with his computer using normal speech. This presents a fascinating challenge. For the first time ever, man will be mass producing information devices which themselves will help in the production of other devices!

Well, what about the use of Space? The big immediate use I see foı Space is of course–communications. I see a huge need for a world-wide synchronous orbit system to allow the development of information transfer without the horrendous problems of communication via thin pieces of copper laid beneath the earth. Let me dwell on that for just a moment. Imagine that the decrease in size and the increase in power of micro-computer chips carries on at the current rate. I can see each of us carrying around (rather like watches) his own personal computers. Back at our homes, each will have his own base computer. Each of these computers will be intelligent enough to act automatically as transceivers. Clearly, once we have a communication network, we have the ability to get access to our home base from anywhere on earth. In addition, since each of the machines be be capable of understanding reasonable speech, the problems of communications between *machines* should vanish. A machine having difficulty communicating with any other would simply go into an education dialogue mode, in order to educate the other one as to what it means.

The reader can surmise, then, that once we have sufficient synchronous orbit satellites with wide-band communication (perhaps in the 14 to 100 gigahertz band) *and* each person has a voice dialogue transceiver computer, that one can ask any question of anything known, anywhere on earth, and get virtually an instant reply.

In fact, it can get more interesting than that. If these computers are capable of decoding speech (which one uses to input the question) then clearly, they can 'listen' to the information they receive to answer the question and thus, 'learn' in the process. (No, I won't be sidetracked into a discussion as to what exactly I mean by intelligence or learning.)

Notice that this information handling will be the first time *ever* that man has actually gained on the information explosion. Each of us will be able to use his pocket computer to audibly reply to just about anything that was every stored in any computer. This includes novels, factual work, historical work, scientific work, etc., etc. This leads into one more very interesting area . . . Ever though about

esting area . . . Ever thought about how a picture is transmitted? One way is to describe each point on the surface of the image by numbers; i.e. digitize. The color is described at that point, the depth of the color and so on. The numbers are then transmitted and on the other side ends up a reproduction, which (depending on the degree of exactitude required) is virtually indistinguishable from the original. With high-speed miniaturized computers and wide-band communications, there is no distinction between handling digital information and handling video information. The appearance of motion is simply a function of the bandwidth required in order to send a sufficient number of image frame pictures. In fact, we can get to the point at which computers will be able to self-generate a picture out of a word description. Now *that's* a fascinating thought. (In fact, it is so fascinating it could be a subject of a whole new discussion but I won't be side-tracked into that either.)

So let's look at electronics. A semi-conductor chip is physically quite small inside its package. In other words, the package itself is relatively large. The chip usually takes up less than 3 to 5% of the total package. That is, the piece actually doing the work has to be protected from impurities, damage and particularly important, be easily handled by a human, in order to make a circuit.

Here on earth it is extremely difficult to make *large* semi-conductor crystals because of the problems of impurities, maintaining a good vacuum, etc., etc. In Space, these restrictions do not exist. It will be possible in Space to manufacture large crystals which *need not be encapsulated.* Also, it will be possible to get very good yields. But one of the most wasteful steps in each semi-conductor manufacturing process is the heat problem caused during the bonding of leads. Yields could be improved even more if heat did not have to be used after the semi-conductor had been made. Think of it. Since the electronics would not have to be encapsulated, there would be no need for a bond. Any device-to-device communications could be by *light*, not by copper or gold. Fascinating.

MORE SPACE & COMPUTERS

Another thought. Some Solid State mass storage devices are being created now which depend upon a particular kind of semi-conductor material (for the technical, '*m*etal *o*xide on *s*ilićon', or 'MOS') being affected by a relatively high energy electron beam. The MOS is then used as a binary storage medium. As it turns out, this could be worked down to about a 500 angstroms bit size. On a four inch square (about 10 centimeters square) surface a lot of information could be stored . How much? let's see.

1 angstrom (A) = 10^{-10}meter

500 A bit size gives 2 x 10^{6}bits/meter

(or 2 x 10^{4}bits/centimeters)

That is: 4 x 10^{8}bits/sq. cen.

or 4 x 10^{10}bits/target (Ta:get being 10cen.sq.)

or 5 x 10^{9}bytes/target

So, 500 megabytes of in:ormation could be stored in a 4" (10 cen.) square piece of silicon with *no moving parts.* Again, fascinating.

On earth, the trouble in producing such a device is that since we need *pure* crystals, and *large* crystals, we need a hard vacuum–which is difficult to obtain. In the Natural hard vacuum of Space there will be little problem in producing a large crystal (perhaps several *square feet* in size.) Let's look at the capacity of a 1-meter square target.

Each square centimeter can hold 4 x 10^{8}bits
so, 1 square meter (100000 sq. cen.) would hold 4 x 10^{12}bits or, 5 x 10^{11}*bytes.*
That is, 500,000,000,000 bytes (= 2 *Thousand* IBM 3330 disk drives) in a cabinet about the size of a TV set!

And remember we are still talking about 2-dimensional structures. What about some kind of 3-dimensional array produced by synthetic methods in Space. A 3-dimensional *cube* memory, assuming the same density, would hold:

$(2 \times 10^{4})^{3}$bits

That is: 8 x 10^{12}bits/cubic cen./cubic centimeter
or, 1 x 10^{12}bytes/cc.
That is: 1,000,000,000,000 bytes.
A 10 cen. cube–which is about 4" a side–(In other words, you could hold it in your *hand)* would hold

1 x 10^{3} x 10^{12}*bytes.*

That is: 1 x 10^{15} bytes.

I am told *that* capacity would hold the entire written contents of the library of Congress! In the palm of your hand!

What would we use this immense storage for? In my opinion, one of the most frustrating aspects of learning, is being unable to find what you want, when you want it. A large memory computer, space-based, with wide-band instant communications would be a very great help. This space based system could store information gathered from earth based sources, and provide that information, to any user, upon demand.

Another fascinating prospect for consideration is that of the superconductors.

What use is super-conductivity in computers? Just in case you had forgotten, super-conductivity occurs when the resistance of a material falls to zero. In all normal materials, there is some resistance to the passage of electricity. So the electricity has to 'fight' to get through the material. This 'fight' produces heat (an obvious example is a light bulb where the filament of the bulb becomes white hot). So what?

Even in micro-circuits this heat causes problems. Semiconductors have to be cooled, power has to be kept down, thermal stability becomes an important factor. And so an. In Space, however, it is so cold that materials become super-conducting. Literally, *no* (zero) resistance. No resistance, 'No fight'. No 'fight', no heat. No heat, no waste power, no problems. Obviously, the cass of materials we use for semi-conductors at the moment may not be suitable for Space computers. But there are some exciting opportunities presented by the effects of super-conductivity which could be explored in Space. (It is after all, rather cold!) Think of it. A huge computer, working on 4 D cells!

I've gone on enough. To me, the advent of a Space-computer match and what it will do to improve mankind's lot over the next 25 years is astounding–and probably to many people, far-fetched. However, the history of the computer business is such that no one has ever predicted the future accurately. They have always been too pessimistic. When the potentials of Space and the solutions to present problems that it offers–the cold, the vacuums and the weightlessness–are added to the incredible power and potential of the computer business, what can be achieved becomes so incredible that the majority of science fiction stories about it will become historical novels in very short order.

Enough for now. I'm sure there are many more things that could be said but space, in the sense of the printed word on paper, limits one. Space, in the other sense of course, doesn't.

Call For Papers

AAS/AIAA CONFERENCE–INDUSTRIALIZATION AND COLONIZATION OF SPACE

CALL FOR PAPERS

San Francisco Bay Area
October 18, 19, 20, 1977

The American Astronuatical Society, in conjunction with the AIAA Technical Committee, announces a multi-dimensional conference on the industrialization and colonization of space. The conference will focus on commercial activities in space over the next ten years.

There will be technical sessions on Large Space Structures (manned and unmanned); Manufacturing for Profit; and Economical Transport Systems. Sessions are also planned to discuss Space Law; Space Community Planning; Psycho-Social Considerations for Space Communities; and Economic Realities of Space.

Papers may consider–but are not limited to–the following subject areas:

Technical

Low-cost fabrication and assembly techniques for large structures in space
Optimum shapes of large space structures
Limits of size in space
Materials for large, light-weight space structures
Shielding techniques for manned communities
Design considerations for solar storm shelters
Radiation considerations: low earth orbit vs. geosynchronous orbit vs. cis-lunar space
Identification of tools needed for assembly of structures in space
Automated fabrication and assembly versus human fabrication and assembly
High capacity power alternatives for in-space use
Control of in-space manufacturing quality
Necessary research before commercial production plants can be implemented
Methods of increasing the number of experiments in orbit
In-space inter-system interfacing techniques
Low-cost transportation techniques
Technique for high frequency launch to orbit and rendezvous
Techniques for rescue and/or return of personnel
Single person transport modes for in-space assembly
Low orbit to high orbit transport alternatives: Cost-effective solutions
Material resources obtainable from extra-terrestrial sources

Space Law

Property rights in space
Freeports for commercial space use
The status of privately-owned space objects and colonies
Legal ties of a space community to earth
Internal legal options for space colonies
The impact of existing and imminent treaties on commercial space operations
The legal framework for intra-space cabotage
Rights of multi-national corporations in space
Rights and protection of information from space
Rights of entrepreneurs in space commerce

Space Community Planning

Architectural and structural considerations
Designing man into the system
Person selection techniques
Analogies of earth communities with space communities
Internal economic flow considerations
Ways of making the community economically self-sufficient
Effects of various G-levels on human activities, manufacturing processes, and agricultural efficiencies
Design of a typical room
Anticipated internal and external traffic flows
Moderate to large community shapes
Intra-community transportation methods
Food production in space communities
Recreational activities and facilities
Space medicine and health care

Psycho-Social Considerations for Space Communities

The effects of packing on human behavior in space
To what human need level should a space community
Anticipated boundaries of stay-time in space
Ranking of services for human needs in space
Design considerations for the human element

Economic Realities of Space

Near-term economic justifiers of space activities
Perceived risk factors in industrializing and colonizing space
How to finance medium to large scale space ventures
Major cost drivers in costing space communities
Economic trade-offs : low earth orbit vs. geosynchronous orbit vs. cis-lunar space
Economic trade-offs: large unified space structures vs. smaller independent space structures
Effects of population numbers and physical mass on colony cost

Abstracts of 200 to 500 words should be forwarded prior to 1 May 1977 to the Technical Program Chairman:

Paul L. Siegler
EARTH/SPACE, Inc.
4151 Middlefield Road
Palo Alto, California 94303

Authors will receive notification by 1 June 1977 of acceptance of their papers.

PEOPLE'S SATELLITES

We've printed info about the Public Interest Satellite Association (PISA) before. Here're some more bits and pieces to (hopefully) whet your appetites and shar pen your ideas about how us people-in- the -street can get a slice of pie in the sky.

We begin with excerpts from a letter Andy Horowitz, PISA's Co-Director, sent to Lee Felsenstein in November, 1976. Lee has been working on the "Cruncher" a travelling computer-based community information system to be housed in a van. The Cruncher project is described in the last PCC issue, Vol. 5, No. 3, page 38.

The second chunk is excerpted from descriptive material published by PISA. Feed in your ideas to Andy at PISA – the sky (universe?) is the limit!

Dear Lee Felsenstein

As I mentioned over the phone, there are some gaps in the range of our current experiments–and a big one is in the area of computer networking via satellite. Bob Albrecht felt you might be interested (and able) in helping to fill this void.

After listening and thinking about what you said, one idea comes to mind immediately: a three-way interconnection involving 1) Boston's Children's Museum, 2) the Lawrence Hall of Science, and 3) your mobile van. Your van could be equipped with either its own satellite terminal for direct communication to both Berkeley and Boston facilities or a micro-wave dish that could access a centrally located satellite terminal within line-of-sight transmission of your van. In short, what I'm suggesting is that it may not be necessary to segregate the experiments you will be performing with your van from those involving the linkage of the other two points. I think NASA would be interested in this kind of interconnect.

What do you think about it? And what other thoughts do you have?

Judging from NASA's receptivity to what PISA is trying to accomplish (which really amounts to our helping NASA do its job more effectively), I'm confident that you'll be able to work something out. So far, the cooperation has been marvelous.

Best,
Andy Horowitz
Co-Director, PISA

Materials Available From PISA

55 West 44th Street
New York, N. Y. 10036

* A Case Study of the ATS-6 Health, Education and Telecommunications Projects (Free, limited copies available)

* A Global Survey of Communications Satellite Systems ($2.00)

* A Study of the Communications Needs, Uses and Costs of Non-Profit Organizations ($7.50)

* Communications Satellites: Now You Can Have Them Too (Free)

* Should People Fight For Satellites? (Free)

PUBLIC INTEREST SATELLITE ASSOCIATION (PISA)

The Public Interest Satellite Association (PISA) was formed in the fall of 1975 to spearhead a broadbased public effort to explore the public interest uses of satellite communications technology and to ensure that its future development serves the communications needs and interests of the American people.

Unlike previous electronic communications technologies such as the telephone, radio and television, the communications satellite is the product of more than $80 billion of tax-payer funds invested in the U.S. space program over the past two decades. PISA's principal aim is to assure that the benefits of this powerful, publicly-financed resource are not lost upon those who could most benefit from what the technology has to offer. These include the large number of community and social service organizations, public interest groups, civic organizations and membership associations, educational and health entities, labor unions and other members of America's nonprofit community–organizations for which the need to develop alternative, low-cost means of long-distance communications has increased greatly in recent years.

PISA's ultimate goal is the establishment of an independent non-profit satellite communications system tailored to meet the specific needs and requirements of this nonprofit sector of the society. Much of its work (e.g., conducting studies on the communications uses and costs of non-profit groups; developing recommendations for future satellite policy directions; getting the word about satellites out to as wide a segment of the public as possible) is being directed toward this end.

NASA Satellite Experiments

An important aspect of PISA's work involves assisting interested non-profit groups acquire actual hands-on experience with satellite hardware itself–to use it, to see how it works and what it can do, and to determine how it best can be adapted to serve their communications needs. Despite the wholly commercial thrust of current satellite developments, there does exist a pathway that public groups can follow to obtain this practical experience. It leads to the National Aeronautics and Space Administration (NASA) and its "Experimental Communications Program."

Since 1966, NASA has launched a series of six Application Technology Satellites (ATS), four of which have been designed to provide communications services. Today, these NASA satellites hover in synchronous orbit above the earth and are a'ailable to the public for experimental use. The following are some of the groups PISA currently is working with to formulate proposals for the use of these experimental NASA satellites:

* The Pacifica Foundation and the National Federation of Community Broadcasters–to experiment with the establishment of a satellite-based FM radio network.

* The United Farm Workers–to interconnect migrant camps on the West and East coasts for purposes of exchanging data, tele-conferencing, and providing vocational training via slow-scan television.

* The Public Interest Research Groups–to form an experimental, interactive information exchange and distribution network linking local PIRGs active in more than thirty states.

* The Arkansas Community Organizations for Reform Now (ACORN)–to expand its community organizing operations throughout the Southern Region and into several Northern states via two-way television, radio, telephone and telex communication.

* The Community Satellite Video Organization–to share and exchange community-produced video material among local cable tv systems.

* The Association of Investigative Reporters and Editors–to expand the reach of a computerized information retrieval system it is developing to provide background material for reporters working on feature stories.

* The National Association of Neighborhood Health Centers–to provide, among others, training to paraprofessionals and health-care specialists active in more than 200 Neighborhood Health Centers across the country.

For these groups and others, PISA has been able to offer assistance in planning and preparing NASA proposals, which must include 1) a detailed description of how a would-be experimenter wants to use a satellite and why a satellite is essential to accomplish what is proposed; 2) carefully framed hypotheses that lend themselves to empirical analyses; and 3) preparation of a management plan and an implementation schedule. PISA's services are available to any non-profit organization that would be interested in exploring the possibility of a NASA satellite experiment.

Public Policy Task Forces

Just as there are technological alternatives to be examined, NASA experiments to be undertaken, relationships with public broadcasters and other commercial satellite entities to be established, and educational, organizational and research tasks to be performed, there is also the all important task of investigating policy options that should be explored to guide the development and use of satellite communications technology.

At the present time, domestic satellite systems are operating in a policy vacuum, in an atmosphere of "deregulation," under a 1972 FCC ruling that has been given the label of "Open Skies." Outside of its inspection of certain technical requirements, the Commission has all but washed its hands of satellites. Its view: that the marketplace, not the regulatory arm of government, should determine the social application of the technology. But what applications are likely to develop from this? And whose needs will be satisfied by it? Where does the Commission's mandate to uphold the "public interest, convenience and necessity" figure into its "deregulation" of satellites? Finally, what about the cold, hard fact of the public's investment? Where is its dividend to be found?

These were the questions that prompted the creation of the Public Interest Satellite Association. Since then, PISA has formulated others:

* Should a non-profit satellite system be developed? If so, how should it be financed and managed? NASA is currently preparing plans for the possible launch of a "Public Service" satellite by the end of the decade. If these plans should materialize, what kinds of services should such a satellite offer? To whom? And on what terms?

more ➡

PCC INTERVIEWS TED NELSON

Ted Nelson

Ted Nelson, author of the book, **Computer Lib/Dream Machines** *(available through the PCC Bookstore) dropped by the other day. For those of you not familiar with the back-to-back books Ted says* **Computer Lib** *"is an attempt to explain simply and concisely why computers are marvelous and wonderful, and what some main things are in the field."* **Dream Machines** *"is specially about fantasy and imagination, ana new techniques for it."*

PCC: Ted, in **Computer Lib** you profferred the chant

COMPUTER POWER TO THE PEOPLE!
DOWN WITH CYBERCRUD!

Has the message been received and acted upon?

Ted: I don't know, but lots of people are moving in this direction. I can't say who has been influenced - maybe those who respond already felt that way in his or her heart. But it's pleased me to meet people eminent in the computer field who were first introduced to computers by reading my book, such as Steve Dompier at Processor Tech. I can't take credit for what he's done but I must say I like it.

PCC: What's the next step in liberating computers (and people)?

Ted: By the wildest coincidence it happens to be the next several things I'm doing. For the dinky computer market (I don't like the term micro), we need easy-to-use home and personal software. I'm involved with the efforts of Sophystems Ltd., a subsidiary of the itty bitty machine co. Far-reaching services for people, possibly clustering around computer stores, including storage, printout, timesharing, music output, movie output, classy typesetting, and (sooner than some people think) libraries. Many have attacked the library problem in an improfound manner. Everybody should be allowed to put stuff in. Everybody should be allowed to get stuff out. Everybody should be allowed to tie stuff together. I have proposed a system of tying stuff together which I presently call the Classic Link Set; all these are currently defined into a literary network which I hope to implement. I also hope to put out a book called "Literary Machines" outlining the real problems, especially freedom of the user and archival safety of the Grand Corpus. Finally, I'm also working on a computer glossary for the totally naive and flustered. The long-awaited second edition of *Computer Lib* remains routinely targeted for 12 months hence, but I do hope to catch the 1977 Christmas market. This one will have readable print and probably 300 pages.

PCC: What resources on dinky computers do you recommend for people unfamiliar with both computers and electronics?

Ted: None.

PCC: What's the Xanadu* Information System?

Ted: I hope it will eventually be a repository for all the writings of the human race. In the meantime, I hope to have a small version of it online within the decade (heh heh).

PCC: Who'll be using the Xanadu Information System?

Ted: Anyone who wants: I'm presently reserving Xanadu sign-ons for those who want them. I may have to stop doing this if demand deeps increasing Tentatively I may offer a $10 package with a sign-on nameplate plus the *Xanadu Users Manual* and the *Xanadu Musers Annual.*

PCC: AT&T's SMRT supercharges can cause phone costs to shoot up as high as $80 per month for phones used for local timesharing. How will this affect a timesharing system like Xanadu?

Ted: I think we'll have the rooftop laser soon enough that it doesn't matter. I'm more worried about SMERSH.

PCC: Have any other soft-copy information systems successfully protected copyrights or provided author royalties, as you plan to do with the Xanadu System?

Ted: I don't know, but it's the only way to get cooperation from publishers. I'm concerned about building from what we have, rather than going collective on everything and hoping it will all work out.

PCC: Where do you see the personal computer movement going?

Ted: I think I was the first to say this in Spring '76: the hobby market will peak at about 40,000, then continue to grow slowly forever; the home market will explode in 1977, dwarfing citizen bands, video games, vibrators, and video cassettes, to name various fields of consumer electronics.

PCC: Thanks, Ted; come visit again soon. . .

The **User's Manual for the Xanadu Information System** *is a 12-page 4 x 5½ xeroxed booklet discussing system contents, copyright provisions, typical operation, etc. for the proposed system. It may be purchased for $5 from*

Ted Nelson
25 S. 7th Ave.
La Grange, IL 60525

* *"Xanadu" is claimed as a service mark and trademark for certain computer and information products and services offered by Ted Nelson.*

PISA(cont)

* What terms of access should pertain to non-profit groups wishing to use existing and proposed commercial satellite systems? As is the case with postal rates and IRS tax codes, should there not be special, reduced satellite tariffs for non-profit users?

* Even if a government "Public Service" satellite does not materialize, there does exist the very real possibility that the non-profit community may be able to support the construction and launch of a satellite system tailored to meet its specific telecommunications needs and requirements. But this possibility can only come to pass if the non-profit sector is guaranteed a useable "parking space" (a satellite's location in space) for whatever satellite it may choose to launch in the future. Hence, what rights will those who cannot yet afford or manage satellite systems have in reserving desirable orbital slots?

* Just as the possibility of sharing public television's earth stations offers many potential benefits to non-profit groups, so does the possibility of sharing the many satellite ground facilities presently being built across the country by cable television entrpreneurs to deliver pay-tv programming. Since cable operators are required by the FCC to maintain public access channels on their systems for free public use, should not the FCC also require that cable's satellite facilities be made available to public groups on a similar basis? Moreover, should not all satellite companies, like broadcasters, be required by the Commission to put aside some of their facilities for free public service use on the theory that the skies, like the airwaves, "belong to everyone?"

These are a mere handful of the policy issues PISA believes must be raised and dealt with in the months and ars ahead if satellite technology is to serve all segments of American society and if a true "public dividend" is ever to be realized. In an effort to seek answers to them, and to explore and wide range of related satellite matters, PISA has formed seventeen public policy task forces:

1. Satellites and User Needs
2. Satellites and Management of Non-Profit Organizations
3. Satellites and Public Television
4. Satellites and Public Radio
5. Satellites and Cable Television
6. Satellites and Citizens Band Radio
7. Satellites and Community Development
8. Satellites and Rural Development
9. Satellites and Health Care
10. Satellites and Education
11. Satellites and AT&T
12. Satellites and the FCC
13. Satellites and HEW
14. Satellites and NASA
15. Satellites and International Policy
16. The Design of A Non-Profit Satellite System
17. The Development of PISA's Organizational Structure

Chairpersons have been designated for most of these public policy task forces, and the work of many is well under way.

LETTERS

Dear P. C. C.,

Hi! My name is Debbie and I played with one of those cumputors. I played Hunt the Wumpus and I throught it was fun so I drew a picture of the Wumpus for you. Well got ta go so bye!

Thank you,

Debbie Ball
3642 Granzotto
Concord, Ca. 94519

Central City Occupational Center, 1646 So. Olive, Los Angeles, which is an activity of the Los Angeles Unified School District, is getting into microcomputers. I attended an evening workshop there in November and December which met twice a week for some glorious bull sessions. It will resume in January on a more structured basis. The objective is to come up with a one-semester full time course in micros. The instructor in charge of the pilot program is Guy Cook. Guy Cook used to be in calculator servicing and is concerned about what I call "endangered species" – micro manufacturers who fail and leave buyers without adequate servicing data. I'm trying to keep a roster of micro and accessory firms (including RGS!!) which may have left the field.

Mike Fern
1046 So. Westlake no. 1
Los Angeles, Ca. 90006

Thanks for the corrections to our list of computer stores, Mike. We encourage all readers to help us keep up to date.

I just read Vol. 5 No. 2, and I think its about time I wrote a letter. First of all, on STAR TREK:

I enjoyed the many letters about star trek programs. I wrote one myself a few year back (the 2-tty one in PCC last year), so I would like to give my thoughts (as I remember them) at the time on writing a star trek program. I wrote it for a science fiction convention (MileHiCon, by the Denver Area Science Fiction Association) so it had to not just be a "space war" type game, but it had to seem like science fiction. My friends & I talked about Star Trek, and after a few hours discussing the numerous things we considered "errors" in the program (like if you have transporters you can make just about anything, and if you have phasers why not just shoot first & see if you can remember to ask the questions later), we got down to trying to pick out some real things from star trek (with the help of the Star Trek Concordance) to make the game out of. Well, there were a lot of problems. There were the little ones, like how big is the universe and if it is really that big how fast is warp 1 and is it a velocity or an acceleration and how fast does phaser fire move or photon torpedoes and what do shields do and is space 3-d or, based on the mutterings used when changing course, 2-d and so on and on and on. (Trumpets) STAR TREK DOES NOT FOCUS ON THE TECHNOLOGY, IT FOCUSES ON THE PEOPLE, THE SOCIETY, AND THEIR INTERACTIONS.
This is why (in my opinion) the technology changes from show to show (e.g. "fire phasers" results in a beam of light on one show, a blob of moving light on another) but the *personalities* are consistent. So, to make a Star Trek game, you just program in the people, the societies, and the interactions. Simple, right? After .05 nanosecs you give up on that idea, and make a space-war-star-trek program anyway. It's easy. You use the words "federation", "Klingon," "phaser," "warp drive" and so on. And that's what we did. Just change the words and you can have any science fiction story made into a game! Then the problem was only to make a regular war type game with the proper vocabulary. I've always hated the play-against-50-synthetic-Klingons type game (especially the part where it says "you have 62 stardates to do such and such" which is just like saying "you have 62 Wednesday,-October 1976's to do such and such.") Since the program can calculate everything exactly, it can hit you exactly if it so desires. This is no fun, so some algorithm to screw things up is usually used. Since few "learn" or anything, they are almost always either too good at the game or too poor at it, and this is soon boring also. The solution? I was blessed with infinite computer facilities, so I made it use two terminals. The computer calculates the ship's positions and so on. It is semi-real-time. Every command does take time and energy, as mentioned in one of the letters I read. Plus, we used regular acceleration & velocity equations for two ships in space with rocket-type engines, so if you're moving then the longer you wait, the further you go. We gave it "scans" that see a certain distance in all directions, not the whole universe, and not whole quadrants. And "long range" scans with more distance but less information content. But all in all, it was rather lacking in plot (what game isn't?) and was still a kill.kill.kill type game. But it did have some good points and so I'll send people who want one a listing for $1 (the paper tapes are available, write for details).

But to get back to the letters, with respect to my experience with my star trek game. To Richard Salz: I agree completely. Those with a 33 have to sit for too long getting useless printout. I avoided all the problems with damage control and Klingons shooting first, but I'd like to say: maybe the author didn't think of all that! First you ask for people to not sneer at limited equipment, but then you ask them to think of everything, and write it in. How will the huge program for all this fit on your limited equipment?

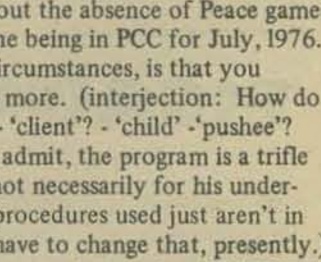
To Robert Currier: I agree that the Star Trek name should be dropped to keep from locking yourself to a set universe. But then, a space war game is just another space war game, but a star trek game is a STAR TREK!!!! game. (So what if it is really, or could be really, a space war game with the serial numbers rubbed off.) There is an advantage to making the name star trek but not using the star trek plot. It will attract players. Those players will be disgusted by its horribility. (At least, if I wrote it I'd put something in too blatant to ignore, like Hortas attacking.) Maybe, just maybe, one of them will figure out that it is a Computer Program, not the Word of God. It could be (shudder) *changed!* Maybe they'll even learn some programming and do it.

To the Mellow Flash:
Three bells, seven golden trumpets, and scattered applause. You seem to actually know something about writing games, not just about star trek. Go back (those of you who can't remember) and read his letter before you write a game, especially the 5 points. Add to point 5: provide negative feedback. When you ask a yes/no question, don't give it the old "well it wasn't yes so it *must* be no" junk. If it wasn't yes, check to see if it really was no, and if not provide some indication that a yes/no answer would be nice to have. Check that numeric numbers are within the proper bounds (and are they integers if they're supposed to be.) Of course, on smaller systems you may not be able to afford such things, but why should anyone have to point that out to you?

To leave star trek now, I would like to say that Jef Raskin ("A Mini Tale" from Datamation) is right, and I've seen it happen too. I taught a programming class with 3-4 students per terminal in a relaxed atmosphere and things worked great. What happens? Next year, 20-25 students per terminal, and a regular class atmosphere. A dismal failure, of course. Think of it – if each student used the terminal for an hour, they'd each get to use it twice! That should be enough, shouldn't it? They could use it before or after school, is a common answer, except that the administrators were against unsupervised activities with expensive equipment, and what about catching the bus home? At first, the computer system was used only for education. When I quit, most of the storage was "reserved" for other purposes (but not used, just wasted.) It still is, though they've got two systems last time I heard.

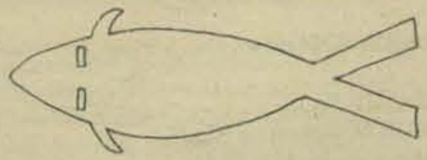
Thanks and keep it going.

Philip Tubb
ALF Products Inc.
128 S. Taft
Lakewood, Co. 80228



Gentledragons, Gentle Ladies, Noble Lords, Distinguished Visitors from Pern,

Every so often, one observes an exclamation in the PCC or in Dr. Dobbs Journal about the absence of Peace games for computers, the latest one being in PCC for July, 1976. What surprises me, in the circumstances, is that you haven't pushed Star Trader more. (interjection: How do you 'push' a game upon a - 'client'? - 'child' -'pushee'? It's supposed to be fun!) I admit, the program is a trifle large for MacHinery here; not necessarily for his understanding, but some of the procedures used just aren't in the IMSAI BASIC. (We'll have to change that, presently.) But some of the Huntington exercises, such as MARKET, are within even Mac's comprehension and crunching capabilities. It should not be impossible to run a little further work in those directions and devise some small operable multi-player games – in which the computer might even be enabled to take up one or more parts, illustrative of the effects of inflations, depletion of natural resources, discovery of new techniques and TRADE on an economy. Correction: that is work – but it's the kind of work that some or others of you folk are good at, already. Don't blame me – (1) I haven't got my journeyman's papers in the programming trade, and I'm still learning! (2) like the Volkswagen driver said I'm pedaling as fast as I can! (3) use the good things you've got, while working on and asking for the good things to come.



Sid Cochran, Jr.
P. O. Box 607
Tyler, Tx. 75710

P. S. I had MARKET up last night. Very interesting situations occur if neither firm wins in the first four years – both firms eventually go broke!

Although I am somewhat hesitant about adding more fuel to a running argument which has now drifted all over the place, I feel complled to reply to the letters of the Old Soldier and of Ron Seibel in Vol. 5 No. 2 of PCC:

I am *not* opposed to technology, nor to people's computing! I *am* opposed to the irresponsible use of technology, to the *de facto* assumption that anything which makes a profit is good, and to unrestrained growth as a goal for modern man. Please try to understand that there is a difference between the wholesale, dogmatic acceptance of everything labeled "technological" and the desire to use only that portion of technology which is needed and useful to solving *human* problems.

Technology certainly is self-limiting! A technological artifact is as much a part of our ecosystem as is anything "natural."

The statement that Gandhi rejected technology shows an almost total ignorance of the work and writings of one of the greatest men who ever lived. M. K. Gandhi was one of the first people to suggest the use of intermediate technology, as advocated more recently by E. F. Schumacher in *Small is Beautiful: Economics as if People Mattered.*

"Intertwingled" is a term, coined I believe by Ted Nelson in *Computer Lib and Dream Machines,* which I am particularly fond of because it so aptly describes the way things are in this Universe.

Ron Seibel says that "some of the best war machines ever made have a proud record of saving lives." Yes, that is true, but what an utterly tragic statement it is! Why do we first of all design machines with the intention of killing people and only then decide that they can also be used to save lives? Why don't we fund the research for life saving purposes in the first place? When are we going to outgrow our childish preoccupation with war and killing, in real life, on television, and yes, in computer games?

The real threat to our continued existence is that we, as a species, may fail to mature before we destroy ourselves fighting over short term goals which have ceased to have any real meaning in our evolution.

Although it has been said before, for once we really *do* live in a unique time. Thanks to technology and engineering, we have the ability to feed, clothe and shelter everyone on Earth. We have all but wiped out disease. We have walked on the Moon and sent messages to the stars. We have undreamed of power, and undreamed of responsibility. All we lack is wisdom.

We have the agonizing privilege to be living during the period of transition, when Man is in the process of changing from an animal into something else. At this point our future is literally what we make it. Can we overcome our past and make the future of our visions? Computers are currently reaching into every facet of human existence; perhaps they are the touchstone through which we can implement our dreams. At the very least, computer people bear an awesome responsibility for the shape of the future. But let's *choose* that future, not just rush blindly into it assuming that we can do no wrong.

To borrow a quote from a contemporary singer, let us "reach for the heavens and hope for the future and all that we can be and not what we are." Out of a group of people as diverse as those found in people's computing, something great has just gotta be possible.

John R. Lees, Jr.
P. O. Box 1543
Rolla, Mo. 65401

P. S. I am getting somewhat tired of this ineffectual letter ping-pong. I'm going to take the first part of 1977 "off" to work up in detail my ideas on tools, ethics, computers, artificial intelligence and stuff. If I like what I come up with, it will most likely appear as a series of articles in *Creative,* or maybe, who knows? Why can't I, too, get a book published?

I shall keep in touch. Keep PCC and Dr. Dobbs going strong. Lot of weird people in California – John

NICE WORDS ABOUT DDJ *FROM A RECENT* DIGITAL GROUP *FLYER*

"It is not very often that there is a journal/newsletter that the Digital Group is able to recommend without some hesitation (and we get them all). However, Dr. Dobb's Journal of Computer Calisthenics & Orthodontia is one pleasant exception. Jim Warren, the editor, has put together a good concept and is managing to follow through very well indeed. There is no advertising in the Journal. It is supported solely on subscriptions. That also means that manufacturers have zero leverage over the content of the magazine. The Journal's primary purpose is to place significant software into the public domain and to provide a communications medium for interested hobbyists. The approach is professional and they are growing quickly. Their latest project is to sponsor "The First West Coast Computer Faire" to be held in San Francisco, April 15-17. The Faire looks potentially like it might be the event of the year for hobbyists.

(In case it might appear otherwise to some people, there is no official link whatsoever between the Digital Group and Dr. Dobb's Journal – we've taken our lumps as appropriate just like everyone else when Jim felt they were justified.)

We think Dr. Dobb's Journal is here to stay and a publication that is a must for everyone in the hobbyist world of computers. Don't miss it!"

I heard that you are interested in biofeedback/related phenomena.

Perhaps you will be interested in a parapsychology "computer" which is binary/analog combination yet has no programming, memory, or firmware.

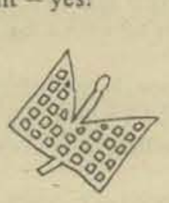
This machine is not fiction, indeed I have a small number, one right beside my chair here.

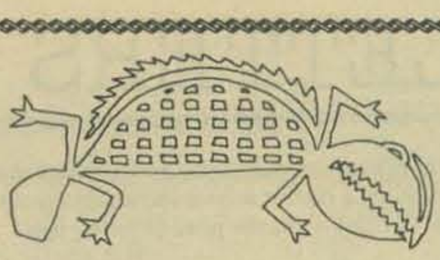
You are familiar with GSR changes in respect to emotions. What is a "machine" worked by monitoring a similar "intensity" field – the Biological energy. Any change in that field produces a go-no go signal which is sufficient to act as a machine which answers any question! (subconcious reaction)

Assuming the Human or Biological memory could be combined – a machine of 16 trillion bytes memory – then any information could be analyzed. Indeed this is a reality. Answers are expressed in wavelengths relative to that question/problem or go-no go.

Example: TB – a Biological disease – frequency 7 cm.
Example: attenuation of energy D2 normal – 1/D machine
Example: Will I hear from Bob Albrecht – yes!
Note: He did.

J. G. Gallimore
5627 E. Julian Ave.
Indpls. In. 46219





A poem. I wrote it about the July issue:

I was in the Computer Mart, passing the time
When a funny-looking dragon caught my eye
I said to myself, 'What can it be?
. . . Why, it's another issue of *PCC!'*

Within a clock cycle I lay down a buck
And picked up their last copy (was *I* in luck!)

I quickly rushed home – at a quarter to three
Opened the issue with ill-concealed glee
My enthusiasm I could not conceal
But as I read, doubts did congeal

Holograms of the heart? Don Juan? Lao Tzu?
That's not the *PCC* I once knew!
Something funny is going on, I fear:
In other words: "What the hell's happening here?"

The rest of the issue seemed okay,
Albeit, *funny* some way

An old back issue up I did dredge:
Re-read a thing called "The Cutting Edge"
What new directions did it foretell?
Of the old *PCC* is this the death knell?

To the question above, make the answer be 'NAY!'!
Come, my good people, what do you say?

I think the above expresses my sentiments well enough. Onward . . .

I had originally decided to do a running commentary, article-by-article, but decided against it. You don't need/want that kind of feedback, do you? If so, I'll be more'n happy to do it. Rather, in this letter I've got some general comments that I'll give, each one as a separate entity, rather than try to patch them together somehow.

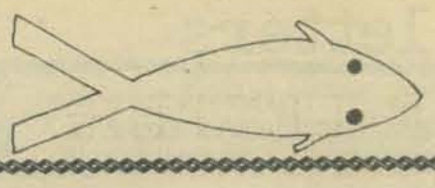
1. I liked the music programs. In my (garbage) collective subconscious I've got some vague glimmerings of an idea or two. The music will be "slightly less" random, and it will have harmonies (I think.) That is, unless Kemeny and Kurtz make me want to throw in the towel . . . That is, if I ever get to see their music program (hint! hint!).

2. About clubs: I haven't seen LICA listed, so I don't know if you know of it/them/us or not. The Long Island Computer Association meets monthly, usually the first Friday, at the New York Institute of Technology in Old Westbury. There are pre-meeting classes, about 70 members, and some nice support from the Computer Mart of New York. For more info, write to Gerry Harrison, Prez. 36 Irene Lane East, Plainview 11803. Or, call from 6 to 9 pm: 516-938-6769.

3. ComputerMart of New York has expanded to Long Island. I've enclosed a copy of LICA's newsletter. This has info on both items two and three.

4. *EDU* by the way is free to them that want. If you make that known mayhap more people will be interested. Also, you left *Electronic News* out of your list of "professional publications." Shame on you! *EN* is a weekly, published (I believe) by Fairchild, and – at least on this coast – is one of *the* "biggies" and deservedly so!

5. In spite of the fact that I agree with Tom Burke's letter (the one against Sprocket Man), how about a full face-on drawing?



6. Have you thought about publishing a cumulative index to *PCC* and/or having a reprint service for some of your more "popular" articles – for the price of xerox fees?

7. About the bookstore list. I don't think you have to list the complete inventory every issue, just the new books, or any sales, and special notices. A complete catalog could possibly be an insert – or somepin' like that – once or twice a year.

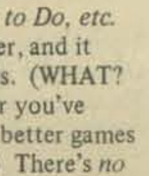
8. Please forgive the rambling tone of this letter, but the hour grows late and mind tends to wander.

9. I have devised a better way to play *TAXMAN* solitaire-style then the way that's shown in *What to Do, etc.* Below is my way of doing it. It's easier, neater, and it makes it easier to analyze the game afterwards. (WHAT? You mean you don't analyze every game after you've played it?!?) I think *TAXMAN* is one of the better games in the book, logic-wise (oh No! A semi-pun!) There's *no* chance involved whatsoever, and analysis really helps you to improve your game as you go on. I have an idea of/ for a strategy on how to play, but I don't quite know how to verbalize it yet. When/if I do, I'll let you know about it.

(By the way, this is the same game you've got on Page 112)

1 2 3, 4 5 6 7 8 9 10 (cross off the numbers as they get taken.)

ME	IT	
8	1 2 4	
9	3	
10	6 7	(no factors left)
27	28	(totals, of course!)

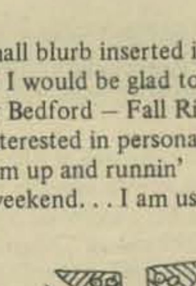


10. I'm still waiting to hear what you think of MOO (nudge! nudge!) . . . that's the game I sent in along with my subscription form.

11. About "THE PHANTOM": "The Phantom" is an omniscient, omnipotent, all-powerful, super-being. He's led to all sorts of interesting things: Phantomizing, which can best be described as throwing obstacles in someone else's way, although it's rarely malicious, usually just for fun. TPSA agents are people who've been authorized by a Spokesman for "The Phantom" to Phantomize someone and affix the phrase TPSA ("The Phantom" Strikes Again.) to his handiwork. It's really not as confusing as it reads . . . I think. (by the by, a friend and myself are the only Spokesmen.)

12. Take Care. Write If You Get A Chance.

Richard Salz
Phantom Unlimited
8 Glenby Lane
Brookville, NY 11545

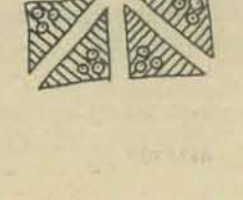


I am sorry that I won't be able to make the "Fault Stapling" in 1977.

Would it be possible to have a small blurb inserted in a forthcoming issue of "PCC" that I would be glad to talk to anyone in the Brocton – New Bedford – Fall River area of Massachusetts who are interested in personal computers? I have a micro system up and runnin' (ALTAIR). I am available any weekend. . . I am using ALTAIR 8k "BASIC".

Thanks for everything.

M. Douglas Callihan
Berkley St. RFD no. 1
Berkley, Ma 02780



I think the home computer user is ready for another use for the computer, that is, simulations. Just getting the computer up and running was a learning experience, then followed the games and other applications like music, security systems, accounting and many others. An application that has not been exposed to the serious amateur is that of simulations. Of course many games are simulations, for example, LUNAR, POLUT, MARKET, to name just a few. These are interesting, educational, and best of all fun. There is a class of simulations that not only can be interesting and fun, but can be used for real serious research by the amateur or professional. I refer to this class of simulations as SUPER LIFE simulations since it is a super set of simulations of which the well known LIFE (see the Mathematical Games section of *Scientific American*, October, 1970, and page 133 of *What To Do After You Hit Return* by the People's Computer Company) by John H. Conway, is just as example. If your computer can do the LIFE simulation then it can do the SUPER LIFE type. All that is needed is a higher level language like BASIC (with subscripted variables) and an I/O terminal.
In general, the SUPER LIFE simulations share some important common characteristics, for example, they:

1) use a cellular array. In LIFE the array has two dimensions.

2) have rules defining the status of each cell. In LIFE each cell is "dead" or "alive."

3) have rules defining the interaction between adjacent (or any) cells. In LIFE there are 3 rules for births, deaths, and life.

4) have a status for the array. The status of the array in LIFE is the pattern at each generation.

5) have a cycle for applying the rule(s) for each cell to determine the status of the array. In LIFE the rules are applied at each generation.

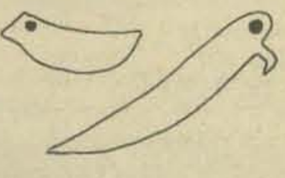
An example of another SUPER LIFE simulation is the computer photo enhancement of pictures from a satellite in space. Imagine a picture composed of a cellular array, say 512 x 512 cells, where each cell has its brightness status represented by a number between 0and 128. If one applies the rule that "each cell becomes as bright as the average (mean) of the four neighbor cells," then after one generation a large amount of the "static or snow" is removed. Note that if this is continued the picture will change to all gray after many generations. Another rule that could be applied to this array (picture) is "let the large brightness numbers become larger and the smaller numbers smaller." This would increase the contrast.
This same SUPER LIFE type of simulation is used at Fluidyne Engineering Inc. to replace wind tunnels. It can also be used to study traffic flow, wave motion, heat transfer (rule: each cell temperature becomes the average of its neighbor cells), economics, and even human social interaction.

Already this letter is getting longer than I expected. I will follow with more details and documentation but for now wanted you to know of my interest and see that this may be a subset of Bill Shenker's interests. I strongly feel that the amateur computer user is ready for more of this type of application. I also think there is then a need for a way for those interested in SUPER LIFE to share and build on each other's "discoveries." I also believe it is only a matter of time till some "amateur" comes up with a significant useful discovery.

I personally am interested in promoting this SUPER LIFE type of simulation and would be interested in serving as a clearing house or referee or some similar capacity.

Am looking forward to your reaction to this. Must get back to KIM now.

Richard Maus
14130 40th Avenue North
Minneapolis, MN 55441
Ph (612) 559-9639



P. S. Possibly a note about me is in order. I teach Math and EDP at jr. high and college level, and also have Major in Physics and sell some real estate and fly and have a MOS Technology Inc. KIM micro computer with 8k RAM.

PCC wants to share with you this letter from PISA, the Public Interest Satellite Association; please share it with others who might be interested. Also see the article about PISA in this issue.

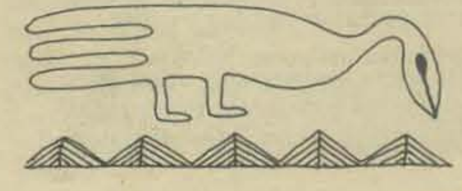
Not too long ago, you took the time and trouble to fill out a long and complicated Questionaire PISA had sent you about your telecommunications needs, costs and future desires. First, we wish to thank you most sincerely for it. Second, we felt you might wish to know something about the results; a copy of a Press Release from PISA is attached for your information.

The effort was worthwhile in all respects. PISA had started the survey with certain assumptions; as a result of your effort and cooperation they were borne out. (The Press Release provides the details.) We were able to meet the time deadline, and, at least as far as your work and ours is concerned, the Federal Communications Commission now has in its hands more information about the needs and wishes of the non-profit sector in the United States than has ever been accumulated before.

Where do we go from here? What has been accomplished is only the beginning. There is still, despite our mutual efforts, the possibility that the FCC (and other policy and regulatory agencies) will not heed the public interest and will yield to those who wish to see the technology most useable for your purposes foreclosed. PISA's principal objective is to prevent just that from happening. Once more, we feel it essential to enlist your help, the help of those organizations who took the time and effort to put their views on record. (We have, of course, as we had originally promised, protected the anonymity of all respondents to the survey.)

If you wish to go one step further – and we earnestly suggest it is in your best interests to do so – PISA would like to recommend a next move. **It may be necessary, in order to protect the potential of satellite technology for the non-profit community, for PISA to file a formal petition before the FCC, requesting that the Commission initiate a separate rulemaking proceeding to fully examine the use of satellites – and what satellites – for non-profit groups.**
At the very least, and this is the "next step" PISA would suggest, it would be helpful to have, from you, some indication of how you feel as an organization about the potential of satellite technology. The Consumer Federation of America recently drafted a resolution along these lines, which was sent to PISA for whaever use necessary. It reads:

Because communications satellite technology was developed at public expense, CFA believes that, as satellite frequencies are allocated, the needs and interests of non-commercial, public-interest organizations, and of those potential users who would most benefit by small and inexpensive technological methods, must be protected through legislative and regulatory policies.



MORE LETTERS

"CFA urges the FCC to initiate a full-scale study of the use of satellites to transmit programs directly to home television sets. Direct satellite-to-home transmission would increase programming diversity by allowing consumers to receive many more channels than the average three or four now being received, and by fostering the development of additional networks. Since the technological capability to provide these consumer benefits will exist shortly, the FCC should immediately proceed to encourage realization of such benefits."

If your organization could pass a similar resolution, to be sent to PISA for transmission to the appropriate regulatory and other agencies as an indication of the feelings of the non-profit community, this would be immensely helpful. Such a resolution is in no way legally binding; it is an important tool which can be used to indicate, to those engaged in the formulation of policy, what the feelings and needs of the sector of society with which we are primarily concerned are in regard to the development of this technology and its eventual use.

We stand at an historic crossroads in telecommunications history. There have been similar such crossroads before; what is at issue here is the possibility that, by default, we will lose out on improved and lower-cost communications from organizations to their members, from organizations to other organizations, from organizations to the public-at-large.

PISA thanks you for your help and cooperation.

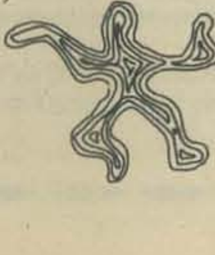
PUBLIC INTEREST SATELLITE ASSOCIATION
55 West 44th Street
New York, N. Y. 10036

Now – I finally have a system up and running: a Digital Group Z-80 w/18K of memory. I brought it home and got it up and running on Thanksgiving Day. Been playing with it for 12 – 26 hrs a day over the weekend, finally had to turn it over to my kids and that seems to be that for now. My wife has yet to play with it at all as she seems somewhat confused about the whole thing, slightly interested, but at least interested. It only speaks TINY BASIC for now but will grow, I suspect, into something else fairly soon. It took me awhile to figure out how to make the damn thing run but after wiggling this and pushing that and reading some it now seems to be working. My kids zap the system now and then and I don't know what or how or why but as this is somewhat of a communal learning situation we should figure out what we are doing. (pls. note: I don't zap it the kids zap it). There have been some problems with putting programs in from tape due, mainly I suspect, to my tape recorder and wrong button pushing but we can usually either fix what is wrong or put it in again and make it work. The games we have are those listed on the first three Digital Group tapes and are fairly simple and tonight the kids (or at least one kid) will try a hand at telling the machine what to do. So – so far it has been a hit but then it is still new and kids, being what they are, may get bored. This is my main concern. Oh yes, it also raises instant hell with a TV set that is in the same room as the computer which doesn't exactly break my heart.

I don't really have much to say and its strange. I've been into this thing so intensely for the last few days that I think that I will take a break, make some pots for money, read some science-fiction, and otherwise take a vacation from a machine that may zap me if I'm not careful. ("Weak and tired after having battled the devious machine monster for four days and nights a truce was called whereat Greg rested and . . . .")

Your Exhausted friend and mine,

Greg White
216 E. Fifth St.
Port Angeles, WA 98362



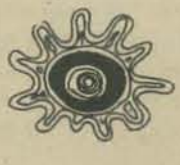
I just ran across your organization in *Alternative America* and I would appreciate as much information about what you do as you are able to send.

I have been working as a programmer for an alternative organization for 3 years. My main interest has been to develop an efficient database system so that the STASH Library can put its some 40,000 documents on psychoactive drugs on-line for easy retrieval.

I am particularly interested in micro-computers and their potential applications to database systems and computer-assisted photocomposition. All our programs to date run on borrowed time on a UNIVAC 1110, but we are considering the purchase of a micro- or mini-computer if it seems feasible and cost effective.

In Peace and Health,

Jim Gamage
109 North Few Street
Madison, Wisconsin 53706



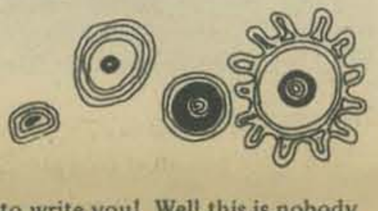
THE SAN ANDREAS FAULT

may it rest in peace

Dear Mr. Albrecht,

A better way of stapling a fault which uses fewer core resources is to make a picture of the fault and staple it with ordinary staples. This is called virtual stapling.

Sincerely,
Susan Chapin
11352 Orchard Lane
Reston, VA 22090



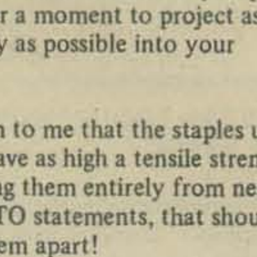
Dear Worm,

So you dare to ask me to write you! Well this is nobody you can order around to your beck and call. I think anyone who could even think about stapling the San Andreas, has to be out of his mind.

Now let's get down to the point. All of us who live on or near the San Andreas really give little for your idea's. I mean that if there is going to be a Quake, why not let "your" computers tell us about it. We all know the danger but we look at it with an air of indifference that only a true S. A. P. (San Andreas Person) can do.

So why don't you people go back to your dear little staple gun, and try to staple your brains in place.

Head S. A. P.
N. M. Madman
1044 Meadowlark Ln.
Corrales New Mexico 87048



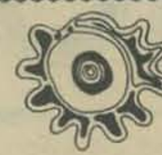
Dear People,

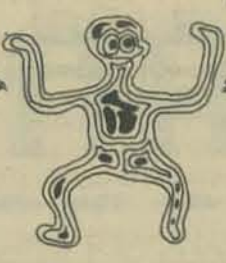
I unfortunately do not expect to be able to join you for your gathering to staple the Fault, as I already have a luncheon engagement on that day. However, if you will be kind enough to inform me of the exact time you plan to commence, I will pause for a moment to project as much of my personal insanity as possible into your project.

On materials – it would seem to me that the staples used on the fault would have to have as high a tensile strength as possible, how about making them entirely from nested loops and unconditional GOTO statements, that should make it impossible to pull them apart!

Best wishes to all, may benign insanity find you in the land of reversible cups & sanitary pedestals.

Barry Soucher
4906 Glenna Way
Louisville, KY 40219





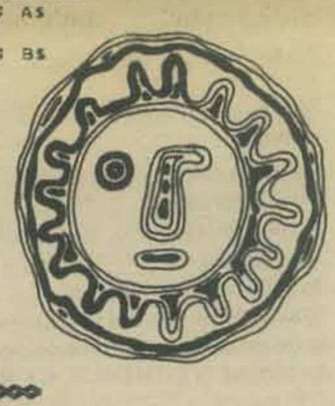
Coan's Solutions

In the last PCC we printed a problem sent in by Jim Coan from his book *Advanced Basic.*

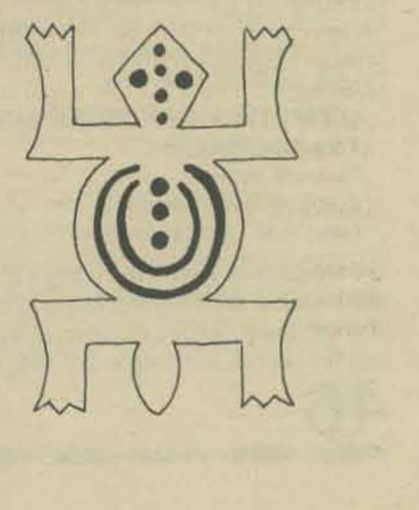
Here's the problem, followed by Coan's solution for HP and one for GE.

"In comparing strings for order, BASIC compares the two strings one character at a time. So for non-equal strings, the first non-equal characters determine order. Therefore, when comparing such strings as '60' and '100', BASIC will evaluate '60' as greater than '100.' Write a program to overcome this flaw. Be sure to provide for proper ordering of strings like 'A6' and 'A10'."

```
NO.8

100  DIM A(10), B(10)
110  PRINT
120  PRINT "AS";
130  INPUT AS
140     IF AS = "STOP" THEN 320
142
150  PRINT "BS";
160  INPUT BS
170  CHANGE AS TO A
180 CHANGE BS TO B
190     IF A(0) = B(0) THEN 270
200  FOR G = 1 TO A(0)
210        IF A(G) = B(G) THEN 260
212
214        REM * ASC(@) = 64  (SEE ASCII CODE)
220        IF A(G) > 64 THEN 270
230        IF B(G) > 64 THEN 270
240        IF A(0) < B(0) THEN 300
242
250     GOTO 230
260  NEXT G
270     IF AS < BS THEN 300
280  PRINT BS; "    "; AS
290  GOTO 110
300  PRINT AS; "    "; BS
310  GOTO 110
320  END
RUN
NO.8


AS? 100
BS? 60
60   100

AS? A60
BS? A100
A60   A100

AS? STOP
```



```
100  DIM AS[25],CS[25]
110  PRINT
120  PRINT "AS";
130  INPUT AS
140  IF AS="STOP" THEN 290
150  PRINT "CS";
160  INPUT CS
170  IF LEN(AS)=LEN(CS) THEN 260
180  FOR G=1 TO LEN(AS)
190  IF AS[G,G] <> CS[G,G] THEN 210
200  NEXT G
210  IF AS[G,G]>"@" THEN 260
220  IF CS[G,G]>"@" THEN 260
230  IF LEN(AS)<LEN(CS) THEN 270
240  PRINT CS;"    ";AS
250  GOTO 110
260  IF AS>CS THEN 240
270  PRINT AS;"    ";CS
280  GOTO 110
290  END
RUN
NO.11


AS?10
CS?6
6   10

AS?6
CS?10
6   10

AS?A100
CS?A60
A60   A100

AS?A60
CS?A100
A60   A100

AS?STOP
```



ANNOUNCEMENTS:

magazines

COMPUTER DECISIONS IN BRAILLE

The October 1976 *Computer Decisions* contains many topics of interest to the blind, and has been printed in Braille. Articles of special interest are

Computers and the handicapped: progress and potential by Senator Bob Dole
A U.S. Senator sees the computer as the means to an end in giving handicapped persons a more productive role in society.

Technology and the deaf-blind by Frederick M. Kruger
Electronic devices, some incorporating microcomputers, expand the horizons of the deaf-blind.

A site for sore eyes by Peter Duran
A computer utility in New England provides service to the blind. The users are solving business problems identical to those run on other systems.

Braille computer terminals by George F. Dalrymple
Terminals capable of embossing output provide the sightless user with full computer facilities. Blind persons can easily keep up with their sighted colleagues on computer-supported tasks.

Turn a deaf ear to your programming problems by Dr. Steven L. Jamison

More than 200 deaf people are currently employed as computer programmers. Each year more join the ranks, proof of their upbeat, positive attitudes.

The Braille *Computer Decisions* issue is available for about $5 from
Peter Duran
ARTS Services
Protestant Guild for the Blind
456 Belmont St.
Watertown, MA 02172
(phone 617/926-4100 extension 27)

A CHUM WE DIDN'T EVEN KNOW WE HAD

by Jef Raskin

Where would you go if you wanted to see a list of works of music written by computers? Articles on concordances, applications of computers to archeology, art, philosophy, linguistics, history, all the humanities and fine arts? Can you believe that there has been a journal publishing this kind of stuff steadily since 1966? That's a decade! And you thought PCC's been around for a while. The Journal is called Computers and the Humanities (abbreviated CHum) and is run by a friendly and helpful Professor of English named Joseph Raben at Queens College, in New York. Compared with PCC the style tends to be somewhat pedantic (but then, compared to us, who doesn't sound a bit pedantic) but the content is usually all there. Subscriptions are steep (something over $25 per earth-year) but get the latest facts from:

COMPUTERS AND THE HUMANITIES
Pergamon Press, Inc.
Maxwell House
Fairview Park, Elmsford
New York 10523

Yours truly was their Algorithms Editor for a few years, and knows it to be a Good Thing from the inside. Lots of notes about upcoming conferences, book reviews, calls for papers, and long words with Greek and Latin Roots.

Electronic Hobbyists' Underground Buying Guide

November 17, 1976, Saratoga, CA. A new directory has just been published that helps amateurs, CBers, experimenters and computer hobbyists locate equipment, parts, supplies and services.

Over 600 sources of standard and hard-to-find gear are listed in the handy guide. Many of the 600 sources are mail order firms and many are discounters, too. All are firms that do business with electronic hobbyists.

The first section of the book lists the firms alphabetically, their complete mailing addresses and phone numbers. Then, a list of the products or services they offer are provided along with key product information, minimum order amounts, shipping charges, etc. You'll find everything from the million dollar mail order firm that handles hundreds of product lines to the week-end garage operation offering a low-cost accessory for a microprocessor or CB transceiver.

The second section of the book is a breakdown of products and sources in over 200 categories. Under each category is a listing of all firms that sell or manufacture the items cross referenced to the first section. For instance, there are over 25 sources of low voltage (12.6 VDC to 13.8 VDC) power supplies and over 50 sources of digital ICs.

The buying guide contains many prices and sufficient detail that you can make your purchases directly from the book or obtain additional information to make your choice. Catalog prices, if any, are listed.

The last section of the book includes all sources by state so you can buy from the firm nearest you when there are several choices. You'll probably find some local firms that you didn't know existed.

You can easily pay for the guide by the savings you make in buying from the least expensive source. In addition, you'll save countless hours of shopping and hunting for hard-to-find items. You won't have to worry about making a poor buying decision— you can easily review all the producers of the equipment you are interested in from the convenience of your living room chair.

The Underground Buying Guide is available only by direct mail from Peninsula Marketing Services, 12625 Lido Way, Saratoga, CA 95070. The price is $5.95 plus 55¢ postage and handling. Californians add 39¢ sales tax. Moneyback guarantee within 10 days if you are not completely satisfied.

For further information contact:

Dennis A. King
Peninsula Marketing Services
12625 Lido Way
Saratoga, CA 95070
(408) 996-0471

DIGITAL TROUBLESHOOTING: a review
by Jim Day

Many computer hobbyists are either neophytes or "software people" who may have had little or no experience in the care and feeding of LSI hardware. When their JOLT, KIM, or POLY breaks a leg they either ship it back to the factory, tote it to the nearest computer store, or shoot it. There is, however, a book that should be quite helpful in providing a basic understanding of digital troubleshooting techniques. This is Richard E. Gasperini's *DIGITAL TROUBLESHOOTING*, available for $9.95 from the Hayden Book Co., 50 Essex St., Rochelle Park, NJ 07662.

This book introduces the reader to the fundamentals of bipolar and MOS technology, explains how to read logic deagrams, and discusses basic tools and diagnostic methods. It then covers the sue of common digital circuits such as decoders, multiplexers, flip-flops, buffers, counters, and shift registers. The book does assume some prior knowledge of electronics, but not of digital circuitry. It discusses the use of memory and display devices, and gives many useful tips on soldering and unsoldering ICs, obtaining replacement parts, etc. Chapter 18 contains some good suggestions on obtaining additional information, and an appendix explains how to interpret manufacturer's part numbers, such as SN74LS195N.

CRYPTOLOGY JOURNAL

CRYPTOLOGIA, a journal devoted to all aspects of cryptology, will be published four times a year. There is a need for a forum for the exchange of ideas related to cryptology in the public sector. This journal will meet that need with research papers, survey articles, personal accounts, reviews, educational notes, and problems. Topics included are computer encryption, paper and pencil cryptanalysis, history and literature of cryptology, mathematical cryptology, cipher machines and reading ancient languages.

Some of the articles in the first issue are:

The Biggest Bibliography, A Review – David Kahn

Cipher Equipment – Louis Kruh

Review of Statistical Methods in Cryptanalysis – C. A. Deavours

Beatrix Potter's Journal Cipher – Barbara Harris

Cryptology of Multiplex Systems – Greg Mellen & Lloyd Greenwood

Poe Challenge Cipher Finally Broken – Brian J. Winkel

Cryptographic Applications of Permutation Polynomials – Jack Levine

Unicity Points in Cryptanalysis – C. A. Deavours

Cracking a Random Number Generator – James Reeds

Subscriptions are $16/yr from
CRYPTOLOGIA
Albion College
Albion, MI 49224

clubs

SCCS

The Southern California Computer Society is a non-profit, international organization whose purpose is to promote and encourage a community interest in the computer arts and sciences. Of particular interest is the application of the computer to education and to the social, scientific and environmental problems of society.

Formed in June 1975 by a handful of computer hobbyists and experimenters, the Society has grown rapidly to

➚

MORE ANNOUNCEMENTS:

clubs continued

include more than 800 consumers, amateurs and professionals the world around. It is now the largest, fastest growing organization of its kind in the world. The Society serves as a focal point through which computer buffs and anyone can meet, learn, and share common interests and needs.

WHAT CAN SCCS DO FOR YOU?

- Frequent meetings for the exchange of information and computer equipment.
- A fine magazine . . . . communicating topics and news of interest to members, the community, and other organizations. Free ads for members. Chapter news and reports.
- Classes on programming, building computers and the general use of computers.
- Informal technical assistance and advice on anything related to computers.
- Consumer clout through group purchases. Co-op purchasing of popular items at reduced prices.
- Centers in development with library, computer center, laboratory, and workshop (soon in your area).
- Ombudsman service – to help resolve consumer complaints. Publish your local news and meeting notices.
- Job placement, talent bank, and scholarships.
- Video and audio presentations (soon).
- Equipment and book reviews.
- Record keeping and billing services for the chapters. (Chapters formed or forming in many areas).
- Specialty groups (Sphere, Altair, Imsai, Educators, Hams, etc.).
- Insurance for meetings and chapters.

WHO SHOULD JOIN SCCS?

- Anyone who is interested in computers anywhere as an individual, an experimenter or a hobbyist
- Anyone who wants to be introduced to the world of computers (but does not know where to start), or
- Anyone who has computer related talents, equipment, services to share.

Memberships are $10 per year from
SCCS
Box 54751 C
Los Angeles, CA 90054
[phones (213) 472-0388, 322-6028]

ADCIS Meeting

The 1977 Winter Meeting of the Association for the Development of Computer-Based Instructional Systems (ADCIS) will be held in Newark, Delaware, February 22-24, 1977. For further information, contact the conference host, Fred Hofstetter, Department of Music, University of Delaware, Newark, DL 19711, (302) 738-2497.

NEW CLUB, MISSOURI – ILLINOIS

The first meeting of the St. Louis Area Computer Club was held October 29 on the campus of Washington University. Approximately thirty people attended the organizational meeting, representing various vocations, ages, and geographic areas (including the Illinois side of the Mississippi River). Jon Elson, president pro tem, remarked that the club may be unique in having a number of obscure computers, making communication among members difficult at any level below the flowchart. Committees on hardware, software, and applications were formed, and questionnaires were circulated to aid in planning tutorials and facilities needed. Members stayed late discussin their favorite topics, with hardware reliability a commonly-voiced concern. Meetings are expected on a monthly basis.

Contact: Lou Elkins, 314/427-6116, or Box 1143, St. Louis, MO 63188

PANHANDLE COMPUTER SOCIETY
AMARILLO, TEXAS

President:
Tex Everett
2923 S. Spring
Amarillo, Texas 79103
Phone: (806) 373-8207

Vice President:
Phil Spray
3006 NE 26th
Amarillo, Texas 79107
Phone: (806) 383-0655

Secretary/Treasurer:
Jerry Fewell
3109 Browning
Amarillo, Texas 79103
Phone: (806) 374-0897

The Texas Panhandle now has a computer hobbyist club and though less than 3 months old, we boast a membership of more than 30.

The Panhandle Computer Society meets every Friday and is open to anyone interested in computers. All inquiries should be addressed to TEX EVERETT or JERRY FEWELL address above. We also welcome replies from other groups especially in Texas.

courses

TAKE HOME A MICRO

W INTEK Corporation has scheduled its popular 3-day "HANDS-ON MICROPROCESSOR SHORT COURSE WITH TAKE HOME MICROCOMPUTER" at nine locations next February and March. Each attendee receives a microcomputer to use at the course and then take home – a WINCE MICRO module including a 6800 MPU, clock, ROM, RAM, serial and parallel I/O. It is fully expandable using WINCE Ram and ROM modules. The ROM contains FANTOM II, a monitor/debug program that allows single step execution of user programs, insertion and deletion of break points, and set up of interrupt vectors as well as allowing user to load and dump programs, examine and change memory and registers, and reset. The course covers microprocessor hardware, software, firmware, and economics. Tuition is $495. Course locations include Boston, Chicago, Dayton, Huntsville, Los Angeles, St. Petersburg Beach, Palo Alto, Philadeiphia, and Washington, D.C.

WINTEK Corporation
902 N. 9th Street
Lafayette, Indiana 47904
(317) 742-6802

A COURSE:
MICROCOMPUTER DATA PROCESSING

"Microcomputer Data Processing" is an extension course offered by the Thames Valley State Technical College, 574 New London Turnpike, Norwich, Connecticut 06360

Microcomputers and microprocessors now control the operation of many different kinds of machines. Equipment now containing µC's (Microcomputers) ranges from small computer data entry terminals to general purpose computer systems, from laboratory instruments to automobiles. This new technology now affects or will soon affect every major industry in the United States. µC's offer a practical, low cost means of incorporating decision making capability in devices that previously lacked such capability. As minicomputers opened new areas of computerized control a few years ago, so does the µC today. µC's will soon pervade all aspects of industrial, commercial and personal endeavors. The µC with its ease of programmability provides a very large range of functions that have been, until recently, implemented either by hardware or not at all because of excessive cost.

Microcomputer Data Processing will give students a thorough knowledge of the MOS Technology 6502 µC. While a particular µC will be used in the course, emphasis will be on software/hardware techniques that apply to µC systems in general. Each student will receive a 6502 µC on a KIM-I system board as well as related documentation from MOS Technology. The Hardware and Programming manuals have earned a reputation as equal to the best in the electronics industry. The KIM-I system includes a 6502 µC, 1152 bytes of RAM, 2048 bytes of ROM, hexadecimal keyboard and LED display, interfaces for a Teletype and audio cassette recorder, two crystal controlled clocks and 15 lines of user programmable input/output. The key to any µC's power lies in software and this course will cover software/hardware in the ratio of approximately 80/20.

Course Prerequisites

Knowledge of Algebra and Electricity

Cost

The $325 course fee includes the following:
KIM-I system board
KIM-I users manual
6502 Programming manual
6502 Hardware manual
power supply kit

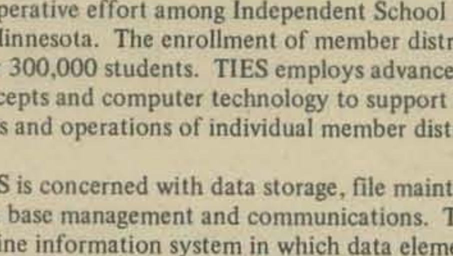
Dates and Hours

March course dates were not available at press time. An earlier course is running from December 6, 1976 to February 28, 1977 (22 meetings). The course meets Monday and Wednesday, 6:30 p.m. to 9:30 p.m.

Notes

A $50 deposit is required upon registration. The balance of $275 is due about 1 week before the course starts. Deposit will be refunded only if course is cancelled. Course oversubscription is expected, early registration is advised. Course will be cancelled if supplies cannot be obtained in time. Four credits will be awarded upon successful course completion.

Contact Frank Rybicki at (203) 886-0177 for additional information.



1977 - - - MIS ORIENTATION WORKSHOPS - - - 1977

TIES – Total Information for Educational Systems is a cooperative effort among Independent School Districts in Minnesota. The enrollment of member districts is near 300,000 students. TIES employs advanced systems concepts and computer technology to support the functions and operations of individual member districts.

TIES is concerned with data storage, file maintenance, data base management and communications. TIES is an on-line information system in which data elements are stored once in integrated files which are randomly accessible for rapid inquiry and update capabilities. The system is capable of providing information for decisionmakers in all district operational areas.

MORE ANNOUNCEMENTS:

courses continued

ΜIS ORIENTATION WORKSHOPS, continued

In October 1974, in cooperation with the Council of Chief State School Officers, the Office of the Deputy Commissioner for the Bureau of School Systems, U.S.O. E., the Minnesota Department of Education and the Minnesota Educational Computer Consortium, the staff of TIES presented an Orientation Workshop to representatives from almost every State and Territory. In response to requests for further workshops the Minnesota School Districts Data Processing Joint Board has elected to offer such workshops on a regular basis. The schedule is given in the brochure.

In order to meet the needs of people with a wide range of interests the workshop is organized in two distinct parts. Day 1(Thursday) is given over to a broad view of the total operation of TIES and the Joint Board which operates TIES. Areas discussed are Governance and Organizational Structure; Management Information Systems and Data Base Management Systems; Student and Census Data Bases; Financial and Personnel Data Bases; Instruction and Research Services; Telecommunication and Interpersonal Communications; and how it all fits together.

On the second day (Friday) individual interests of the participants are met in small group sessions. TIES project managers, systems analysts and operations staff, along with district coordinators, are available for detailed discussion of particular areas of interest to participants.

If time constraints make it possible for someone to participate in the workshop for only one day, then it is recommended that the THURSDAY session be the one attended.

Where

The workshop sessions will be held at the TIES building at 1925 West County Road B2, in Roseville, Minnesota. Roseville is an inner ring suburb north of St. Paul.

The sessions will start each morning at 9 a.m. On Thursday the afternoon session will finish at 5 p.m. Finishing time on the Friday afternoon will be arranged to best meet the travel arrangements of the participants.

Who

The workshops are organized in such a way that individuals with diverse backgrounds and responsibilities may benefit from participation. In particular State Department personnel, superintendents and others with decision-making responsibility who require information in order that they may make knowledgeable decisions based on current data could benefit immensely from attendance at the workshops.

When		**Thursday**	–	**Friday**
1977	January	20	–	21
	March	17	–	18
	May	19	–	20
	August	18	–	19
	October	20	–	21
	November	17	–	18

Cost

The cost for both days (including materials but excluding meals and accommodations) is $95 per participant. If a reservation is made for more than one individual, then the cost is $80 per participant. If an individual can attend the first day (Thursday) only, the cost is $60.

Reservations should be sent to Dr. James Lindsay at TIES, 1925 W. County Rd.B2, Roseville, Minnesota 55113, (612) 633-9110. Payment must be made in advance. A $10.00 handling fee will be charged for cancellations made during the two weeks prior to the workshop.

In order that the most productive sessions can be organized for the second day, it would be helpful if there were included with the reservations brief descriptions of the position each participant holds in the school district or other organization and the particular interests of that individual. Be certain to list the dates that the participants wish to attend.

Hotel reservations will be made by the TIES staff. Participants will be informed well in advance of the hotel where reservations have been made.

software

BASIC ETC

RICHARDSON, Texas – September 23, 1976 – Binary Systems Corporation today introduced a new interpreter program for 8080-based microcomputers.

Called BASIC ETC, the new interpreter was co-developed by John Arnold and Dick Whipple of Tyler, Texas, implementors of the original Tiny BASIC programming language.

An interpreter, in contrast to a compiler, translates and executes the user's program directly. A compiler generates a machine code program which is subsequently executed independently of the high level source program.

Arnold said they decided on an interpretive translator for convenient editing and quick program development capability.

Our goal was to develop a variant of BASIC designed specifically for the hobbyist and small business user, keeping in mind that the most important priorities -- from the user's standpoint -- were ease of program development and straightforward, one-step program execution. We feel we've accomplished that goal, and with a memory-efficient program, too, he said.

BASIC ETC uses the lower 8 KB of memory plus at least 1 KB of RAM for scratchpad. Since BASIC ETC is for games and business applications, the less frequently used scientif:c functions of Dartmouth BASIC are not available. According to Arnold, BASIC ETC is readily software adapted to the individual's system, and *the best answer today for the 8080-based microcomputer owner shopping for an easy to use high level language.*

The BASIC ETC kit, which includes the program -- on either audio cassette tape or paper tape -- and a 32 page, detailed user's manual, sells for $25.00. The manual sells for $6.00 separately.

Kits may be ordered from. the Micro Store, 634 S. Central Expressway, Richardson, Texas, 75080. The Micro Store is the retail affiliate of Richardson-based Binary Systems, Inc. Orders should include a check or money order for the price of the item. For cassette tape, the purchaser must indicate his choice of either the Kansas City or Suding/Digital Group recording technique.

Features of BASIC ETC are listed below:

* Immediate delivery.
* Readily software adapted to user's system.
* Resides in only 8 KB of memory.
* Supplied on either cassette tape (Kansas City or Suding/Digital Group format), or on paper tape.
* Thorough explanatory manual.
* Full string capability - up to 255 characters string variable.
* N-dimensional arrays.
* Variable precision arithmetic.
* Easily handles assembly language routines.
* Direct memory and I/O addressing.
* 27 error codes.
* Both character and line erasure editing.
* Subroutine nesting permitted.
* 31 commands and statements.
* 8 functions plus user defined functions.
* Null control: 0 to 25 seconds.
* Formatted output statements.

BASIC + SOFTWARE

Lewis and Clark College is experimenting with an expanded use of time-shared computing in its curriculum through a three-year productivity grant from the Northwest Area Foundation. Many academic areas of the college which stand to benefit greatly from computer involvement are precisely those areas where the faculty has little or no computing experience. Thus, although adequate facilities are necessary, we have found that it is equally important to sponsor a faculty and student development project. After a series of programming seminars, participants in this project acquire, develop, test and implement materials for their particular disciplines.

Many of the materials listed below are in the testing and development stage. All of these programs are written in Basic Plus and run on a PDP 11/45.

Biology: Simulations and multiple choice questions to support a semi-self-paced introductory course using Curtis as a text.

Chemistry: Short drill and practice problems similar to homework and test questions. Longer tutorial exercises to precede faculty-student tutorials.

Economics: Problems to be solved prior to computer sessions with grading and evaluation of answers done by computer. Simulations and statistical methods programs to teach concepts.

German: Programs to enhance vocabulary of second year students through the use of compounding, and using prefixes and suffixes.

International Affairs: Programs to facilitate interactive data analysis, compare national perceptions, determine similarity in patterns of escalation of past crises, simulate a multi-national corporation and view consequences of user's decisions, and teach students about the constraints on American foreign policymakers through simulation.

Music: Programs to build musical vocabulary and ear training for use in basic skills course. Some of these programs require a "Plato" music box and all use a Tektronix 4010 graphics terminal.

Physics: Prográms that describe and display results of Newtonian mechanics, ray optics, quantum mechanics and electricity.

Math: Programs to provide the necessary background for utilizing mathematics in the understanding of other disciplines.

even MORE ANNOUNCEMENTS:

and more software

Dr. John C. Abele, Director
Educational Computing Project
Lewis and Clark College
Portland, OR 97219

Please specify the areas in which you are interested and enclose a self addressed envelope.

OMSI will handle the exchange at its customary courtesy exchange rates:
1) $1.50 for domestic postage ($5.00 foreign) plus
2) one of the following:
a) 2 11-formatted DEC tapes, or
b) your own tape plus $12.00 (your tape may be 11-formatted DEC or 9 track mag at 800 BPI) or
c) $24.00

PLEASE DO NOT REQUEST OTHER EXCHANGE MEDIUMS

6800 SOFTWARE

TIRED OF THE 'SOFTWARE VACUUM'?
Need Software to feed your 6800? We've got it!

GAMES OF SPECIAL INTEREST

1. **Hangman**: The old word guessing game. Easily modified with your own word list. 640 Bytes $3.25
2. **Acey-Ducey**: A card game played against the computer. Bet and try to break the Bank. $3.25
3. **Craps**: Casino style. Match your luck against the computer and win money $3.25

FLOATING POINT PACKAGE

4. Full floating point capability. 9 digits accuracy with exponent range -99 to +99. Add, subtract, multiply and divide. 512 Bytes. We include FREE I/O driver routine. $5.00

**Watch for our full scientific floating point package.

SPACE SIMULATION GAMES

5. **Space Voyage**: Similar to the famous STAR TREK with only a few limitations. Runs in 4K. $10.00

***Watch for our complete STAR TREK that runs in 8K.

6. **Klingon Capture**: A small space simulation game but with many of the same surprises. Requires only 2K to run. $4.75

SOFTWARE – SOFTWARE – SOFTWARE

7. Stockmarket	(1K)	$3.50
8. Line Editor	(512)	$4.00
9. Random Number Gen.	(60)	$1.50
10. Mastermind	(512)	$3.00
11. Card Shuffle & Deal	(640)	$2.75
12. Number Guess I	(256)	$1.50
13. Number Guess II	(512)	$2.00
14. Hurkle	(640)	$2.00
15. Rover	(1K)	$2.50
16. Switch	(512)	$2.00
17. Chomp	(512)	$2.00
18. Subroutine Pkg.		$3.00

SPECIAL PACKAGE DEALS

All supplied in a 3-ring binder

I. Programs 1,2,3,6,9,10	$13.50
II. Programs 1,2,3,9,10,11,12,13,14,15,16,17	$18.95
III. Programs in I and II plus 4 and 8	$29.50

PROGRAM OF THE MONTH CLUB!

$2.00 gets you a 1 year membership. Receive a monthly bulletin of new programs. No obligation to buy anything but you get a 15% discount on the featured program.
** Free Random Number Generator if you join now!

GENERAL INFORMATION

All program listings include:
A. Complete instructions.
B. Fully commented source listing for 6800 microprocessots.
C. Hex listing for immediate entry using MIKBUG*.
D. Complete label table.
E. Listings punched for 3-hole binder (not included).
F. Sample program output.
G. Easily converted I/O.
*MIKBUG is a Motorola trademark

+ FUTURE PRODUCTS +

Blackjack, Slot Machine, & others

Full Text Editor. Hardware: Cassette interface. A/D-D/A sys.

+ ORDERING INFORMATION +

Add 50¢ handling under $10.00. Indiana res. add 4% sales tax. For more information send SASE

Technical Systems Consultants
Box 2574
W. Lafayette IN 47906

PDP – 8/M SOFTWARE

I am one of a small group of programmers who have been working at Choate School on its PDP8/M system for a few years. We have worked in both BASIC and assembler. hardware and software, APL and FIV, etc. But as we see the computer field expanding, we would like to make some of the more interesting games that we have designed on our system available to other users. The cost we have listed for each program is just slightly over what it will cost us to duplicate the files on paper tape and send them to buyers. All profits will go to our computer center for hardware improvement, but we hope to gain correspondents in other fields of computers through their purchase of our games.

Charles L. Perkins
Software Librarian
Choate Computer Center

NEW SOFTWARE TO REJUVENATE YOUR BASIC . . .

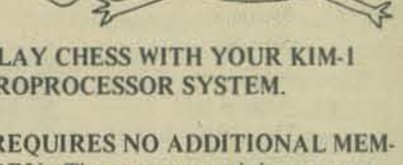
TREK2	*	$2.50	-You command the illustrious starship Enterprise in another version of the popular computer game. Here you must warp between six planets infested by klingnon battle cruisers - destroying as you go. 40 Klingnons in 20 days will win!
3DTREK	*	$2.50	-Your mission: to protect your star base from klingnon invaders in a three-dimensional space where manuvering is tougher than it sounds. Get the klingnons before they can get the base!
FOOTBL	*	$2.50	-Simulates a game of football. Choose offense or defense to begin. Plays are: Short, Screen, and Long Passes; Center and End Runs; Double and Single Reverses; Quarterback Sneak; Field Goal; Punt and Quick Kick; ZONE, RUSH, Quick Rush, BLITZ, and Man On Man defenses. Score and Time on request!
HANGMN	*	$2.50	-Play hangman with the computer. Fill in the boxes one by one and try to guess the word. 200 words initially, but more are easy to add. Look out, you only get ten guesses for each word!
SEA	*	$2.50	-Try to survive the ordeal of the sea. An island lies but a scant 200 miles from your small raft. You need preset amounts of sleeping pills, food, water, and tobacco which you must use to survive each day. Only the right combination will win!
WAR	*	$2.50	-Meet the Nazis in battle. You (the Russians) fight on a semi-realistic battlefield of the attack on Leningrad. Deploy Artillary, Soldiers, and Tanks to recapture the doomed city - or return to Russia in defeat.
ARCTIC		$2.00	-Can you elevate to the challenge of trying to survive in the arctic colds with only a tent and a few meager supplies? A propane heater is all that stands between you and freezing cold. There's only one way to reach the replacement party and be rescued!
SEARCH	*	$2.00	-Make up your own word search puzzles. You supply the words, the computer places them horizontally, vertically, and diagonally backwards and forwards in a puzzle whose dimensions you specify. Now that you've created the puzzle, try to find the words yourself!
PUNCH	*	$2.00	-Punches a message up to 100 characters long onto your paper tape via the paper tape punch. Any character that can be typed on a TTY can be used except back-arrow and at-sign.
BANNER	*	$2.00	-Prints a banner lengthwise onto TTY paper using the asterisk as the printing character. Same limitations on message and characters as the program PUNCH.
RAISE		$1.50	-An equation of the form: (aX + bY) raised to the N can be expanded. For example, (3X + Y) squared plus 4XY plus Y squared. X and Y can have exponents, too.

* Files marked by an asterisk (*) contain certain statements in use on our EDU250 BASIC system and may not be suitable for your system. String functions such as MID, CAT, and LEN are used and the NOECHO command or equivalent for the program HANGMN. We can (at a small extra cost) re-write our programs for your system, or give you information on how to re-write them. If this is the case or if you want these files in another language, please feel free to write us about it. We can also write hardware for specific applications on your system in just about any language including assembler. Write us for details.

SEND ORDERS, REQUESTS, OR QUESTIONS TO:
Software Development Section
Choate Computer Center
The Choate School
Wallingford, Connecticut 06492

CODE BREAKER

FOR SALE: CRYPTOGRAPHIC PROGRAM that will baffle the codebreakers of the CIA. ENCODE and/or DECODE your private correspondence for maximum security. Documentation of cypher technique, program listing and punch paper tape in BASIC for only $6.
Jon Stedman
1528 Summit Rd.
Berkeley, CA 94708



MICROCHESS

NOW YOU CAN PLAY CHESS WITH YOUR KIM-1 6502 BASED MICROPROCESSOR SYSTEM.

MICROCHESS **REQUIRES NO ADDITIONAL MEMORY.** The program and data occupy only 1100 of the 1152 bytes of available RAM.

MICROCHESS **REQUIRES NO ADDITIONAL PERIPHERALS.** All moves are entered and displayed via the KIM keyboard and LED display.

→ → ⇨ →

MORE SOFTWARE, MAGAZINES, CLUBS, ET CETERA AND SO ON.....

MICROCHESS **PLAYS CHESS.** Although a good chessplayer will probably beat the program, he will be surprised again and again by challenging moves.

MICROCHESS **HAS SEVERAL LEVELS OF PLAY.** You may set the program up for 3, 10, or 100 seconds per move. Change the speed at any time during the game!

MICROCHESS **IS EXPANDABLE AND FULLY DOCUMENTED.** You receive a Player's Manual, complete annotated source listing, and Program Documentation describing the strategic algorithms. Instructions are provided for modification, expansion or system conversion. Experiment with your own strategies by replacing one simple subroutine.

MICROCHESS **COSTS ONLY $10.00.** Send your cheque or money order today to:

MICROCHESS, 1612-43 Thorncliffe Pk. Dr., Toronto, Ontario, M4H 1J4, CANADA

SPACEWAR — NOW IT'S AVAILABLE FOR CROMEMCO DAZZLERtm

Probably the most famous of all computer games is available for first time for a microcomputer

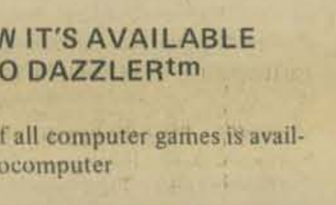
SPACEWAR is available for the Dazzler. That's a message that will interest many a computer hobbyist. The Cromemco TV Dazzler is one of the best-known peripherals in the hobbyist field. It lets the hobbyist see the content of this computer memory mapped in full color on an ordinary color TV set (SPACEWAR, however, is displayed b/w).

With the new Cromemco software, two players play SPACEWAR using two Cromemco JS-1 joystick consoles. The players control orientation and thrust of their rockets with the joysticks. Switches on the joystick consoles let the players fire missiles or transcend into hyperspace. Sense switches on the console can select a sun with or without gravity.

SPACEWAR requires a little over 4K of memory. Another 2K is required for Dazzler picture storage.

Price for a paper tape listing of SPACEWAR is $15. Tapes are available at computer stores or from Cromemco. Delivery from Cromemco is immediate.

For more information contact:
Mr. Joe McCrate
Cromemco
2432 Charleston Rd.
Mountain View, CA 94043
(415) 964-7400



SCCS SUIT

On November 30, 1976, the Southern California Computer Society (SCCS) filed a suit in the Los Angeles County Superior Court against Robert Jones and Robert Jones doing business as McPheters, Wolfe and Jones.

In the suit, SCCS seeks an injunction restraining Jones from alleged unfair competition, compensatory and punitive damages, and an accounting. The dispute centers around *SCCS Interface* which has been published since mid 1975. SCCS alleges that Jones, who provided SCCS with publishing services from December, 1975 through July, 1976, is engaging in unfair competition in publishing the magazine *Interface Age.*

The unfair competition claim alleges that, among other matters, Jones is using, in *Interface Age*, a name, logotype, format and design improperly similar to those of *SCCS Interface*; is unfairly using the SCCS membership list; and is making improper representation to and threats of lawsuits against various parties.

The suit alleges three causes of action in addition to unfair competition. It seeks an accounting for membership dues and other monies allegedly collected by Jones, seeks the return of personal property and alleged damages for its use, and alleges the conversion of other tangible property.

For further information contact:
Larry Press
Box 5429
Santa Monica, CA 90405
(213) 396-0048

COMPUTER ART

The Computer Arts Society is offering a set of approximately 135 slides on Computer Art for $40.00. Please make checks payable and send orders to:
Ruth Leavitt
5315 Dupont South
Minneapolis, MN 55419

STONY BROOK HOME-BREW

I'd like to inform you that we have formed a Stony Brook Home-Brew Computer Club, which is both hardware and software oriented. We have friendly and informal meetings and access to various simulators and cross-assemblers on the school computer. Members have their own systems to demonstrate and discuss (8080 and 6800).

Dr. Ludwig Brown
College of Engineering and Applied Sciences
State University of New York
Stony Brook, NY 11794

Note: Lud Brown is director of the Huntington II Project, source of many computer simulations widely used in high schools.

SMUG at SMUD

SMUG (Sacramento Microcomputer Users Group) meets at 7:30 p.m. on the third Tuesday of the month at SMUD (Sacramento Municipal Utility Districts? That's PCC's guess . . .). The SMUD Maintenance building is on 59th Street between Folsom Blvd. and "S" Street (right behind Corti Bros. store). There's plenty of nearby parking.

The club is an informal forum where computer hobbyists gather together to discuss their views, problems, and new information in the field of personal computing. It is a place where new hobbyists can learn from the mistakes of the seasoned (one to two year) veterans in the field so the neophytes don't reinvent the wheel so to speak.

We normally have one or two speakers on chosen topics which are of interest to the members of the club. A question and answer period follows each speaker to adlib on the topic. Next we then have a short AC-DC (advertisements and commercials from distributors and commercialists) period. The intention is to acquaint the club members of local availability of products, equipment, and systems, and create discounts and price breaks for members.

Thereafter, we have final announcements and then go into our "Random Access" period where individuals can obtain personal information and help from others in the meeting.

Push & Pop, the SMUG Newsletter, is free to active members, $5/year (12 issues) for others. Advertising is accepted.

Push & Pop
SMUG, P.O. Box 161513
Sacramento, CA 95816

THE ULENAR CORPORATION PRESENTS THE MARIN COMPUTER CENTER

Ulenar is a non-profit educational corporation. Its purpose is to research and make available to the public new applications of technology in the field of education. Our first project is the creation of a community computer center in central Marin County. The goals of this center are to (1) make available educational computer programs and games for people of all ages, (2) provide the public with enjoyable as well as intellectually enriching experiences, and (3) de-mystify the computer.

Sophisticated computers no longer cost millions of dollars. In the near future, people will have them in their homes for the price of a used car. In preparation for this time, the Ulenar Corporation is making computers readily accessible for public use.

As the future denizens of a computerized world, we feel that today's children have much to gain from early and ful-filling interactions with computers. Today's adults as well could greatly benefit from recreational interaction with computers. As Dr. John G. Kemeny states in his book, *Man and the Computer*:

> ". . . for many inexperienced users, the opportunity of playing games against a computer is a major factor in removing psychological blocks that frighten the average human being away from free use of machines."

The educational use of computer games encourages imaginative thinking and develops self confidence. Participants take risks, confront uncertainty, develop decision-making capabilities and problem-solving skills in new and unexpected situations.

The computer is an admirable teacher. It waits patiently while the student experiments with different lines of thought and possible solutions. After each student-response, the computer provides immediate and non-judgmental feedback. All students learn at their level of ability and at their own rate.

The Marin Computer Center will be equipped with a time-sharing minicomputer, ten color CRT (TV) terminals, a large library of educational and recreational programs, and video equipment. Schools that wish to rent or purchase their own terminal for on-campus use may tie into our com-

puter through their telephone. The center will be staffed by highly skilled educators and will be open for scheduled field trips, ongoing classes, teacher training seminars, and general public use.

In order for the Marin Computer Center to become a reality, we need a show of support from the community. If you feel this project can be of value to the people of Marin and to yourself, please contact us at (415) 388-1294 or write to The Ulenar Corporation, 236 Almonte Blvd., Mill Valley, CA 94941.

David Fox, Director
Annie Fox, Educational Coordinator

COMPUTER RECREATIONS

To the amazement of everyone – most of all myself – we are finally on-the-air. Through a series of highways, byways, detours, and dungeons, we arrived at
609-448-7900
with a multi-user BASIC system devoted to fun, games and all sorts of entertainment. We can be dialed up from 7 p.m. to midnight Monday through Friday and from 2 p.m. to midnight on Saturday and Sunday. It's all for free right now as we shakedown ourselves and the system. After we are rolling we'll charge 99¢/hour. We have the usual array of BASIC games but are going to concentrate on inter-user computing. We have a mailbox facility for leaving notes for other users and getting same. We will soon have some fine multi=user games and, of course, we'll have a simple conference/rap facility. We are also working with some local personal computing people to try some downstream loading of such computing systems.

We'd love to hear from anyone – by mail or via the suggestion box on the system – with ideas about how our system can grow to be a community's system.

Yours in cyberfun,
Scott Guthery
Computer Recreations
P. O. Box F
Cliffwood, NJ 07721
(phone (609) 448-7900)

MINNESOTA COMPUTER SOCIETY MEETINGS

Monday, February 28, 1977 – 7:30 p.m.
TWIN CITY FEDERAL (Edina Branch)
3924 W. 50th
Edina, MN

Monday, March 28, 1977 – 7:30 p.m.
TWIN CITY FEDERAL (Edina Branch)
3924 W. 50th
Edina, MN

Program:
"Computer Privacy Legislation"
"Microprocessors in the Intelligent Terminal"

(This March meeting is co-sponsored with MAEDS, the Minnesota Association of Educational Data Systems.)

T. C. F. rides again!

The **Trenton Computer Festival** (TCF), the originator of amateur computer fests, will happen again, on April 31 – May 1, 1977.

The second annual **TCF** will be bigger and better than the original. It is being expanded to two full days, with new and larger facilitities to house up to 90 exhibitors. There will be 30 speakers and demonstrations galore. There will be hundreds of door prizes and a huge outdoor flea market.

Computer conference sections and forums are planned on the following topics: Microcomputers for home, radio amateurs, education and medicine: consumer applications of microprocessors; computer music; robots; graphics; speech synthesis; establishing amateur computer standards; computer club convention.

It is expected that attendance will exceed 3,000. There will be amateur computing contests and awards, Saturday night awards banquet, programming copying service, forum audio cassette copying service, free bus service between hotels/motels and fest, and free parking for up to 4,000 cars.

TCF-77 is being sponsored by: Amateur Computer Group of New Jersey, Philadelphia Area Computer Society, Trenton State College Digital Computer Societ", Institute of Electrical & Electronic Engineers - Princeton Section and the Department of Engineering Technology, Trenton State College.

SUSTAINING SUBSCRJBER WANTS BACK ISSUES

Hi –

We'd like to complete our set of PCC back issues. We need:
Vol. 2, Numbers 1 – 4, 6
Vol. 3, Numbers 2, 3, 5, 6
Vol. 4, Numbers 1, 2

If you have them and are willing to part with them, please contact us.

Thanks
Mike Calhoun
Boneless Beef Packing Co.
P. O. Box 301
Palestine, TX 75801

AN 8080 ASSEMBLER

RO–CHE SYSTEMS announces a new 8080 CPU Assembler for use with their Multi-Cassette Controller. This new assembler (Assem 1.0) now makes it possible to assemble source code programs larger than memory.

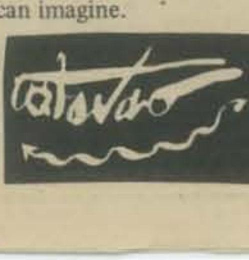
The Multi-Cassette Controller allows an 8080 computer to read or write individual records to any of four software selected recorders, rather than blocks of data. Normally with a two-pass assembler the source code resides in memory. The RO–CHE Assem 1.0 reads the source code from cassette tape record by record and builds a symbol table during the first pass. During the second pass the source is read again and the object code is created and stored in memory. Upon completion of the second pass the assembled program may be executed or dumped to cassette.

When using the RO–CHE Assembler it is no longer necessary to limit the program comments (so necessary six months later) because comments never take up memory space as with a conventional assembler.

Assem 1.0 is available from RO–CHE Systems, 7101 Mammoth Avenue, Van Nuys, CA 91405, for only $10.00. Included is the Assem 1.0 and line editor on cassette tape with an instruction manual.

Here's a plug for the latest in keyboards from SWTPC. For those of us whose only experience with SWTPC keyboards is with the original ones, the bad memories may make one a bit leery about giving them another chance. The new KBD-5 has much better contacts than before, and it uses the 2376 keyboard encoder which makes it a cinch to redefine keys. At $50, plus another 10 for UART+ to make it standard serial RS-232, this is the best possible buy I can imagine.

Bob Powell
22 Bunker Hill Run
East Brunswick, NJ 08816



CALL FOR PAPERS

UPIEEE-77 WORKSHOP, JUNE 1977

BENCH PROGRAMMING OF MICROPROCESSORS

Did you know that you can incorporate microprocessors into your product without the use of expensive microprocessor program development equipment? This workshop is soliciting papers from people who have been doing just that. Three types of papers are expected – software, hardware and trouble-shooting (debugging). All hardware papers accompanied by a working board will be published in the Proceedings.

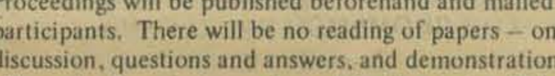
In the 1950s the physicists gave us the transistor. Before we could conveniently use it in circuit design, we had to strip it of the theory with which the physicists had saddled it, and provide it with a practical engineering approach. Now the computer people have given us the microprocessor. Before we can conveniently use it, we have to strip it of the top-heavy computer approach and provide it with a practical engineering approach for use in circuit design.

The purpose of the UPIEEE-77 workshop is to exchange experiences and learn pitfalls and shortcuts in writing firmware; and in trading-off firmware, software and hardware right at the design bench. Papers are due before February 1st.

The workshop will take place on Friday, Saturday and Sunday, June 10, 11 and 12, in Philadelphia. Dormitory space will be available to out-of-town participants. Proceedings will be published beforehand and mailed to participants. There will be no reading of papers – only discussion, questions and answers, and demonstration of boards. Boards chosen for-publication-only will be displayed in the lobby, and authors will be given time for poster talks. Time will be allocated for bull sessions.

For futher information about UPIEEE contact Miss Helen B. Yonan, Philadephia IEEE, Moore School, Univ. of Penna., Philadephia, PA 19174.

For editorial information about UPIEEE-77 phone K. V. Amatneek, Chairman, at (215) 448-7951 or leave message at (215) 233-2962.



WINTEK CUTS MICRO PRICE 50%, ANNOUNCES FANTOM-II

Wide customer acceptance of its WINCE MICRO MODULES has allowed WINTEK Corporation to halve the price on its single card microcomputer. The price was reduced from $298 to $149 for the minimum configuration WINCE CONTROL MODULE consisting of a 6800 MPU, clock and baud rate generator, 1K ROM with FANTOM-II, 128 byte RAM, and ACIA (UART serial I/O) or PIA (16 TTL lines parallel I/O). The price for the maximum configuration module consisting of 6800 MPU, clock and baud rate generator, 1K ROM with FANTOM-II, 512 byte RAM, ACIA (UART serial I/O) and 2 PIA's (32 TTL lines parallel I/O) was reduced from $398 to $199. All WINCE modules (control, RAM, ROM, EROM programmer, A/D, etc.) are on industry standard 4-½" x 6-½" inch printed circuit boards.

FANTOM-II is a new 1K monitor/debug program that allows single step execution of user programs, insertion and deletion of break points, and set up of interrupt vectors. It also allows the user to load memory, examine and/or change memory, print and/or punch memory, display MPU registers, go to user's programs, and reset.

hardware

SC/MP Keyboard Kit

LOW–COST SC/MP KEYBOARD KIT ANNOUNCED BY NATIONAL SEMICONDUCTOR

Santa Clara, CA . . . An inexpensive hand-held terminal for use with the 8-bit "SC/MP" microprocessor kit has been developed by National Semiconductor Corp. to eliminate the need for a costly teletype system, according to Hashmukh Patel, marketing manager for the company's "SC/MP" products.

Known as the "SC/MP Keyboard Kit," the new terminal gives SC/MP kit users low-cost input/output capability through a hand-held calculator-type keyboard with a 6-digit hex display. It features simple microprocessor control to allow the user to evaluate the SC/MP CPU, and direct object code program manipulation for development of a variety of application software.

The keyboard kit comes complete with manual, all required integrated circuits, resistors, keyboard display cable connector assembly, wire wrap connectors, precut wires, and a hand-held wire wrap tool. The heart of the SC/MP keyboard kit is a ROM firmware package (512 bytes) which replaces the "Kit Bug" ROM originally supplied with the SC/MP kit and which allows use of the hexadecimal keyboard to execute programs, to modify or examine the contents of memory and the SC/MP registers and to monitor program performance.

The keyboard is arranged as an 8 x 4 matrix array, but only twenty of the possible thirty-two keys are used. The functions for each of the twenty keys are defined as follows:

- 16 key for hex command value O thru F
- Abort Command
- Memory Command
- Go Command
- Terminate Command
- Power on/off switch is used for initializing SC/MP

Abort Key: Terminate current command and return to wait loop.

MEM Key: Read next memory location. If pressed while in wait loop, it will display last referenced address and the data at that address. If pressed while in memory mode, it will increment memory address and display data at that address. If in "go" mode, it will display error.

GO Key: Display last referenced address. If TERM Key is now pushed, execution will begin at this address. If a hex key is entered, this will be the address that is first executed when the TERM key is pushed.

TERM Key: This key is used to end the present mode of operations. If in the memory mode, the first push of the TERM key will terminate the input of address and allow input of data. The next push of the TERM key will write the data entered at the address specified. If in the "go" mode, pushing the TERM key will cause execution to begin at the specified address after the registers are restored.

The SC/MP keyboard kit is useful for engineers and companies who do not have access to a teletype. It can be used as a low-cost teaching, learning, and developing tool for hobbyists, professors, students and electronics entrepreneurs of all levels. The price is $95.

NEW "SC/MP – II" MICROPROCESSOR —HIGHER SPEED, LOWER POWER CONSUMPTION, SINGLE VOLTAGE SOURCE

Santa Clara, Calif. — Samples are now available of a new N-channel MOS version of the "SC/MP" 8-bit single-chip microprocessor that is twice as fast and which uses only one-fourth as much power as the P-channel version.

According to Hashmukh Patel, marketing manager for "SC/MP" microprocessor products at National Semiconductor Corp., the new "SC/MP–II" microprocessor has all of the features of the older version while offering several advantages over the P-channel device.

"Power consumption of 'SC/MP–II' is less than 200 milliwatts, considerably lower than the approximately 800 milliwatts consumed by the earlier version and the lowest power consumption of any N-channel MOS processor on the market today," Mr. Patel said. "Another significant improvement is that we have eliminated the need for two power sources. The 'SC/MP–II' chip needs only a single source of +5 volts for operation, which is a great improvement over the first model which required two power sources — a +5 volt and a –7 volt supply."

The speed of "SC/MP–II" is twice that of the P-channel model. The new version takes one microsecond to complete a microcycle, and typical instruction execution time is 5 microseconds.

The "SC/MP–II" is fully compatible with its predecessor in terms of pin configuration, object code, and software, and with slight modifications to the crystal frequencies, it will be compatible with all of the "SC/MP" support equipment, such as the "SC/MP KIT" in the U.S.A., the "SC/MP INTROKIT" in Europe, the "SC/MP LCDS" (Low-Cost Development System), and the "SC/MP Keyboard Kit" which eliminates the need for a teletype machine.

The clock oscillator, which is located on the "SC/MP–II" chip, is designed to use very inexpensive television-type crystals of 3.58 or 4.0 megahertz. As an alternative to a crystal, the user may drive the clock with a standard TTL (transistor-transistor logic) timing system. In addition to the clock, all of the inputs and outputs are compatible with TTL devices and can also be easily interfaced with MOS and CMOS circuitry.

The "SC/MP" CPU was designed as a self-contained general-purpose microprocessor which would be easy to use in stand-alone, DMA (direct-memory access), and multiprocessor applications. It is intended for use in general-purpose applications where cost per function is one of the most significant criteria. But even though the "SC/MP" devices are inexpensive, they offer a variety of useful functions that are not provided by some of the more costly microprocessors, such as self-contained timing circuitry, 16-bit (65K) addressing capability, serial or parallel data-transfer capability, and common memory-peripheral instructions. Other features of the "SC/MP" include:

- bidirectional TRI-STATE$^{(R)}$ 8-bit data bus
- 16-bit addressing
- serial I-O ports for easy interfacing
- built-in flags and jump conditions
- an interrupt structure that gives fast response to asynchronous events
- a delay instruction that simplifies timer systems
- a set of 46 control-oriented instructions
- handshake bus-access control on chip
- multiple addressing modes – program-counter-relative, immediate data, indexed, auto-indexed, and implied
- three user-accessible control-flag outputs
- separate serial-data input and output ports
- two sense inputs
- capability to interface with memories or peripherals of any speed

Sample quantities of the "SC/MP–II" microprocessor are immediately available from the factory, according to Mr. Patel. The price for a single sample is $17.76, and production quantities of more than 1,000 will sell for approximately the same prices as the P-channel "SC/MP," which is "significantly below $10 each," according to Mr. Patel. "Prices will be considerably lower in 1977 because we will then be able to offer the 'SC/MP–II' in a plastic package. We are also working on a very-low-cost retrofit kit which will allow the users of the existing 'SC/MP KIT' and 'INTROKIT' to evaluate the N-channel 'SC/MP–II' with a minimum of cost and effort."

MICROCOMPUTER EATS PDP-8 SOFTWARE

The newest PDP-8 software-compatible microprocessor is now available from PC/M, Inc. The late addition to their line is the PCM-12A, a 12-bit microcomputer designed around the Intersil IM6100 microprocessor. Like its predecessor, the popular PCM-12, the '12A is fully compatible with DEC's software for the PDP-8/E minicomputer, and is an expandable bus-organized system with capacity for up to 32K words of memory. Several features added to the PCM-12 resulted in the '12A designation. Major improvements to the Central Processor module have produced "quieter" signal distribution on the 15-position backplane bus. The CPU module also includes a built-in crystal-controlled Baud-rate generator which services all asynchronous I/O device interfaces on the bus. The control-panel design has been expanded to include an absolute binary-format tape into any field of memory at the press of a front-panel switch. Front-panel nomenclature has been improved for stronger PDP-8 compatibility. Cabinetry and documentation have also been "beefed up," and a floppy-disk mass-storage capability has been added to the system. The machine is available in both kit and assembled form. Either way, the microcomputer is fully compatible with nearly all software written for DEC's PDP-8 family. With addition of the floppy-disk for mass storage, the PCM-12A is even capable of running DEC's OS-8 and Intersil's IFDOS operating systems. Available interfaces include serial and parallel I/O, high-speed reader/punch (all DEC-compatible), and an audio-cassette recorder interface. Memory modules include NMOS and CMOS RAM and a flexible EPROM module. Priced in kit form as low as $799.00, with CPU, control-panel, lK static RAM memory, cabinet and power supply. For more information, contact Pacific Cyber/Metrix, Inc. (PC/M), 180 Thorup Lane, San Ramon, CA 94583; telephone (415) 837-5400.

NEW STUFF FROM MANUFACTURERS

from Jim Day

RCA Solid State (Box 3200, Somerville, NJ 08876) now offers a COSMAC microprocessor kit for $249. Designated the CDP18S020 Evaluation Kit, this expandable PC board includes a CDP1802 MPU chip and all necessary supporting components *except* power supply, terminal, connectors, and cabinetry. Easily assembled in a day or less, it features a utility ROM and start, stop, and single-step controls.

SGS/ATES Semiconductor Corp. (435G Newtonville Avenue, Newtonville, MA 02160) has developed a new line of chips for use in music synthesis. Included are tone generators, rhythm generators, etc.

Texas Instruments (Box 5012, Dallas, TX 75222) is rumored to be working on a bubble memory device capable of storing 92K bits and able to transfer data at a rate of 50K bits per second. If all goes well, evaluation quantities may be available in a few months.

Fairs

PERSONAL COMPUTING
1977 NATIONAL COMPUTER CONFERENCE

Dallas Convention Center • June 13 – 16

The NCC traditionally has provided a focal point for new trends within the computing industry . . . and now,

Personal Computing is the idea whose time has come!

The 1977 National Computer Conference will feature:

- **Personal Computing Fair** —Exhibits of individually owned projects selected in a competitive screening process will be on display. A number of awards will be given in several different categories.
- **Personal Computing Program** – Two days of paper sessions are planned for Personal Computing. The deadline for submission of papers is past.
- **Personal Computing Club Congress** – Representatives of clubs will gather to exchange ideas and discuss issues relating to club activities and problems.
- **Personal Computing Headquarters Hotel** – One hotel will be designated as the Personal Computing Headquarters Hotel. Social functions and workshops will be scheduled at the hotel.
- **Personal Computing Product Exhibits** – In order to give emphasis to the Personal Computing exhibits, there will be a special area designated for Personal Computing products. Space is available to qualified exhibitors at half the normal cost.

The '77 NCC will be the year's largest gathering of data processing users and computer professionals. Approximately 25,000 people are expected to gather for the conference program of over 100 sessions plus the year's largest display of computer hardware, software, systems, and services featuring over 250 exhibitors.

For exhibit space sales
information contact:

Carol Sturgeon
Manager, Conference Operations
National Computer Conference
210 Summit Avenue
Montvale, NJ 07645
201/391-9810

For other information
contact:

Harold Mauch
Personal Computing Chairman
PerCom Data Company
4021 Windsor
Garland, Texas 75042
214/276-1968

HOW TO ENTER NCC'S PERSONAL COMPUTING FAIR

The 1977 NCC will feature a Personal Computing Fair. Entries in the fair must be noncommercial personal or group-owned small computing systems. Entries can be actual or documented systems and may feature hardware and/or software implementations, games and recreation, music and art, amateur radio and scientific applications, miscellaneous and general applications of the personal computer.

A one-page description of the proposed entry will be the basis for preliminary selection. The description must be cleanly typed, double spaced on one side of the paper. The page should include the name, address, and telephone number of the entrant or group. Deadline for the Fair Entry description is March 15, 1977. 100 entries will be selected and the entrants will be notofied before April 15, 1977.

The fair entry to be on display at the National Computer Conference will usually be a working system. However, if the project does not lend itself to a working system exhibit (for example, certain types of computer generated art may take weeks to produce), the display should be viewer-oriented with illustrations sufficient to convey the methods used as well as the end product of the project.

Prizes will be awarded in the following categories:

- HARDWARE AND SOFTWARE IMPLEMENTATION - system componentry as opposed to application.
- GAMES AND RECREATION – if it is for fun, this is the category.
- MUSIC, ART and the SENSITIVE SCIENCES – whatever swells the spirit.
- AMATEUR RADIO – hams, this is your own area.
- SCIENTIFIC APPLICATIONS – it probably controls something or solves scientific problems.
- BUSINESS APPLICATIONS – anything that helps us manage our personal business or small business affairs.
- GENERAL APPLICATIONS – if it doesn't fit elsewhere, enter it here.
- AWARDS OF THE JUDGES – expect some surprises and fun from these awards.

entries will be selected and the entrants will be notified before April 15, 1977.

SHOW & TELL

Be a SPEAKER or PANELIST at the FIRST WEST COAST COMPUTER FAIRE

EXCLUSIVELY DEVOTED TO PERSONAL & HOME COMPUTING

- Present a **TUTORIAL** for an absolute beginner, just starting to think about owning his own computer
- Describe a **COMMERCIAL or HOMEBREWED SYSTEM** of interest to personal computing enthusiasts
- Discuss the **DESIGN or IMPLEMENTATION of SOFTWARE** for home & hobby computers
- Detail a **CURRENT or FORESEEABLE APPLICATION** of personal computers
- Debate a **CONTROVERSIAL ISSUE** in the realm of individual computing

Or, suggest an exciting speaker or interesting topic, appropriate for the Computer Faire

ALSO

EXHIBIT a HOMEBREWED SYSTEM

- Hardware, Software, or an Integrated System
- Prizes for the Best "Home Cooking" (just like the old county fairs)

SEE & HEAR

Attend the FIRST WEST COAST COMPUTER FAIRE, April 15-17, 1977, in San Francisco Civic Auditorium

- 7,000 to 10,000 people
- 100 Conference Presentations
- 200 Commercial & Homebrew Exhibitors
- Special-Interest Social Centers
- Banquets with nationally-known, present-world, & future-world speakers
- Tutorials for Beginners
- Esoterica for Experts

Co-Sponsors include amateur, professional, & educational groups.

Some Conference Sections Being Planned:

- Personal Computers for Education associated with a University of California short-course
- Computers & Amateur Radio
- Computer Graphics on Home Computers
- Computer-Driven, & Computer-Assisted Music Systems
- Speech Synthesis Using Home Computers
- Microprogrammable Microprocessors for Hobbyists
- Program & Data Input via Optical Scanning
- Floppy Disc Systems for Personal Computers
- Computer Games: Alphanumeric & Graphic
- Computers & Systems for Very Small Businesses
- Personal Computers for the Physically Handicapped
- Personal Word-Processing Systems
- Software Design: Modularity & Portability
- Several Sections Concerning Standards
- Other Sections for Club Leaders, Editors, Organizers, etc.

Get Your *free* Copy of the *Silicon Gulch Gazette*

- A tabloid newspaper for computer fanatics
- All the news about the Computer Faire
- Details about the *Proceedings* of the Computer Faire
- *Plus* "hot news" & raging rumors from Silicon Valley (the San Francisco Peninsula), product announcements, equipment descriptions, etc.

THE FIRST
WEST COAST
COMPUTER FAIRE

□ Send a *free* copy of the *Silicon Gulch Gazette.*

□ Send program & accommodations information, & Pre-registration Form (those who pre-register may take advantage of a reduced admission fee).

□ Send speakers' instructions.

□ Send homebrew exhibitors' instructions.

□ Send commercial exhibitors' prospectus.

□ Send details on the expected Computer Faire *Proceedings.*

PLEASE PRINT OR TYPE

NAME____________________

ADDRESS ____________________

____

CITY__________STATE____ZIP____

77-1-10½

COMPUTER FAIRE † (415) 851-7664

BUSINESS REPLY MAIL
No Postage Necessary If Mailed In U.S.A.

FIRST CLASS
Permit No. 91
Palo Alto, CA

POSTAGE WILL BE PAID BY

THE FIRST
WEST COAST
COMPUTER FAIRE

BOX 1579
PALO ALTO CA 94302

Supplement to the ·PCC BOOKSTORE·

The 2nd edition of
WHAT TO DO AFTER YOU HIT RETURN
will be available in the PCC Bookstore
Feb. 25, 1977.

This NEW edition will contain listings of ALL the games, and will cost $8.00.

CHESS AND COMPUTERS

David Levy
1976 151 pp. $7.95 Paper

David Levy has an International Master rating in chess. He is also a computer scientist. The combination has produced an exceptionally lucid book – *Chess and Computers.*

Starting with the Automaton Chessplayer of Baron von Kempelen (actually a hidden human), he traces the development of mechanical chess players all the way up to the current research.

The major portion of the book is devoted to attempts to program computers to play chess. Levy manages to explain these programs without losing either the chess players or the computer people. There is also an excellent bibliography of English and Russian language works on computer chess up to July 1975.

I would recommend this book to anyone interested in computer chess on any level. – Eryk Vershen

LOGIC DESIGN PROJECTS USING STANDARD INTEGRATED CIRCUITS

John Wakerly *Stanford Univ.*
1976 203 pp. $6.95

Here's a source book for introductory and advanced experiments using standard, inexpensive and reusable TTL or CMOS integrated circuits and plug-in breadboards.

* Common features of logic breadboards and hints for using them

* Guidelines for project design and debugging

* Interesting, fun projects directed toward designing, building, and debugging a particular system

* Macrologic – a family of LSI parts for high-performance microprogrammed digital systems

* First-hand experience in replacing SSI and MSI with LSI by doing conventional implementation, and then using the Macrologic on the same project

THE MYTHICAL MAN-MONTH (Essays on Software Engineering)

Frederick Brooks
1974 188 pp. $6.50

Frederick Brooks is described just inside the front cover as the "father of the IBM System/360." Now the 360 is among the largest computers, and the software, known as OS/360, occupies a small library. Hardly what you would call a Personal Computer. And yet, Brooks comes across not as a paragon of gigantism, but as a perceptive analyst and an engaging writer. There is a sense of re sponsiveness to the larger context beyond the gates of industry and the walls of academia. For this reason this unified collection of essays has a utility independent of scale. This book explores the human aspects in the creation of artifacts. It is an argument for conceptual integrity, which is ultimately an argument for a sense of style.
– Marc Le Brun

Reprinted from *CoEvolution Quarterly*
Box 428
Sausalito, CA 94965

THE PROGRAM DEVELOPMENT PROCESS: THE INDIVIDUAL PROGRAMMER

Joel D. Aron
264 pp. $13.95

Practical guidelines on how to achieve effective results in the procedural aspects of programming. For programmers whose work ranges from simple business applications to large complex systems programs.

This volume reveals the sequence of steps to take in the development process and how to determine which activities are most important in tradeoff situations where conflicting objectives have to be resolved.

Contents in brief
History of the Program Development Process. Some useful Concepts. Problem Analysis and Planning. Program Design.Coding. Debugging, Documentation, and Delivery. Using On-Line Terminals. Conclusion to Part I.

About the author
Joel Aron is presently a member of the technical staff in the IBM sponsored Systems Programming Series being published by Addison-Wesley of which this book is the first volume.

AN INTRODUCTION TO DATABASE SYSTEMS

C. J. Date
366 pp. $15.95

Database management – a comparatively new subject of constantly increasing importance – is concerned with how data is stored and retrieved in computers. This book introduces the fundamental concepts of database systems clearly and concisely. It presents all three approaches to database management (relational, hierarchical, and network) in considerable depth with the emphasis on the relatively new and extremely important relational approach. As examples of the hierarchical and network approaches, the book includes detailed tutorials on IBM's Information Management System (IMS) and the proposals of the CODASYL Data Base Task Group (DBTG).

Prerequisite: a reasonable knowledge of a modern computer system, particularly its file-handling features, and familiarity with at least one high-level programming language.

Contents in brief
Database system Architecture. The Relational Approach. The Hierarchical Approach. The Network Approach. Security and Integrity.

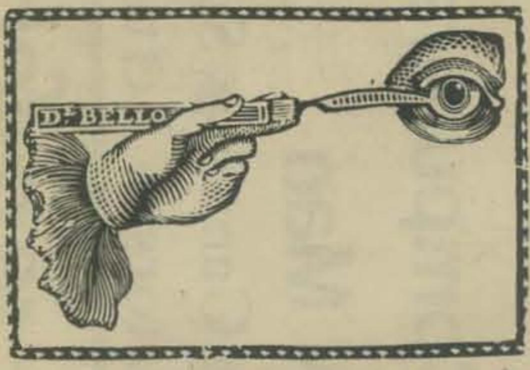
About the author
C. J. Date is a project programmer in the IBM General Products Division, Palo Alto, California. Formerly, he was with IBM United Kingdom Laboratories Ltd. The author and coauthor of several papers about database, Mr. Date is also a frequent lecturer particularly on the subject of relational database. A full member of the British Computer Society, he received both his B. A. and M. A. degrees from Cambridge University, England.

RECURSIVE PROGRAMMING TECHNIQUES

William H. Burge
277 pp. $15.95

Written on the advanced undergraduate-graduate level, this book examines various elements of programming (program structure, data structures, parsing, sorting) from the point of view of expressions rather than mechanisms. It contains several novel and significant developments in the theory of computability and the use of mathematics. Describes a particular method of programming which uses a language based on the notation of the lambda calculus. Numerous examples of this method explain and illustrate the techniques. The main thesis of the book: the emphasis on expressions, as opposed to mechanisms, in many cases simplifies and improves the task of programming.

A number of examples of programs are expressed as recursive functions, often the simplest way to specify a program. Most of the programs given as examples have a fairly simple structure and provide excellent starting points to encourage programmers to embark on more complex programming tasks along the lines suggested.



Contents in brief
Basic Notions and Notations. Program Structure. Data Structures. Parsing. Sorting. Index to Programs.

About the author
William H. Burge is a Research Staff member at the T. J. Watson Research Center. He has been with IBM since 1965. Formerly, he was with E. M. I. Electronics, Ltd. working on the design of a computer and its programming system and with UNIVAC Division of the Sperry Rand Corporation as Manager of Programming Research.

TECHNIQUES OF PROGRAM STRUCTURE AND DESIGN

Edward Yourdon
***1976* 384 pp. $16.95**

This book tells how to
- Design smoother-running programs
- Reduce testing problems
- Speed-up any debugging process
- Increase your personal productivity
- Slash valuable CPU time

What's more, you'll find dozens of ways to minimize maintenance costs, cut expensive CPU time, save memory space and increase your personal productivity with the fully detailed procedures sampled here:
* Getting the most out of general-purpose subroutines
* Using decision tables to avoid program contradictions, incompleteness or redundancy
* Converting a cumbersome unstructured program into a streamlined structured program of great simplicity
* Specifying proper inputš automatically with a test-data generator (TDG)
* Killing "bugs" faster with DDT (Dynamic Debugging Technique)

* AND MUCH MORE

SORTING AND SORT SYSTEMS

Harold Lorin
451 pp. $16.95

A non-mathematical reference work that examines the various sort algorithms from the programmer's point of view, including many examples of principle sort techniques and practical applications to existing systems.

In this comprehensive presentation, Lorin covers a diversity of programming situations, ranging from sorting small internal lists to sorting massive files on storage devices. And he has added an appendix with PL/1 and ALGOL programs for easy reference.

Contents in brief
Internal Sorting: Basic Sort Concepts and Basic Sorts. Exchange Sorts and Linear Insertion. Shell's Sorting Method. Structure in Sorting. Tournament Sorts. Trees in Insertion. Quicksort. High-Order Selection Sorts. Internal Merging. Distributive Sorts. Comparison of Internal Sorts. External Sorting: The Sort Phase of an External Sort. Tape Merging. Polyphase Tape Merging. Cascade and Compromise Tape Merges. Oscillating and criss-cross Merges. Tape Merge Overview. Random Access Sorting. Sorting Systems: Generalized Sorting System. Special Systems Consideration.

About the author
Harold Lorin is a faculty member at the IBM Systems Research Institute. Prior to his present assignment, he was on the Senior Staff of the Service Bureau Corporation, with the Systems Development Corporation and, for many years, with the UNIVAC Division of Sperry Rand.

THE STRUCTURE AND DESIGN OF PROGRAMMING LANGUAGES

John E. Nicholls
572 pp. $17.95

This volume provides the reader with valuable insight into the problems of language deisgn and structure. The major programming languages (FORTRAN, COBOL, ALGOL, PL/1, and others) are included.

In order to clearly demonstrate the principles of language design, Mr. Nicholls uses a topic-by-topic approach rather than language-by-language. Each topic is discussed from two points of view: that of the user who has some computing task to carryout, and that of the implementor who must construct a compiler. To help you select the most appropriate language for your needs, comparative studies are also included. Prerequisite: one year of programming experience and a knowledge of one or more high-level programming languages.

Contents
Introduction and Technical Background: Introduction. User aspects. Applications. The implementation of languages. Theoretical foundations. Elements of Procedural Languages: Program structure and representation. Data elements, variables, and declarations. Data structures. Expressions and assignment. Sequencing and control structures. Basic input-output. Subroutines, procedures, and programs. Appendix: History and outline structure of major languages.

About the author
John E. Nicholls is manager of an advanced system project in the Programming Center of IBM United Kingdom Laboratories Ltd. in Hursley, England. He was formerly responsible for PL/1 language development at Hursley.

COMPILER DESIGN THEORY

Philip M. Lewis II,
Daniel J. Rosenkrantz,
and Richard E. Stearns
672 pp. $18.95

This text covers the basic mathematical theory underlying the design of compilers and other language processors and shows how to use that theory in practical design situations.

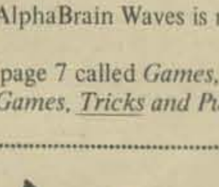
The applicable mathematical concepts come from automata and formal language theory developed in a rigorous but informal style to meet the needs of a wide range of readers. The authors believe that these concepts constitute an excellent basis both for teaching compiler design and for designing real compilers, and have designed two commercial compilers based on this theory.

Contents in brief
Implementing Finite State Machines. Mini-Basic Lexical Box. Pushdown Machines. Context Free Grammars. Syntax Directed Processing. Top Down Processing. MINI-BASIC Syntax Box. Bottom Up Processing. Shift-Identity Processing. Appendix A: Mini-Basic Language Manual. Appendix B: Relations. Appendix C: Grammatical Transformations. Bibliography.

About the authors
Philip M. Lewis II is Manager of Mathematics and Software Design at the General Electric Research and Development Center.

Daniel J. Rosenkrantz is an Information Scientist at the General Electric Research and Development Center and is on the adjunct staff of the State University at Albany.

Richard E. Stearns, Ph.D. from Princeton University, is a Mathematician at the General Electric Research and Development Center.



ERRATA

(From PCC Bookstore Catalog, V5–3)

(1) The price of AlphaBrain Waves is now $1.75 (page 10)

(2) The book on page 7 called *Games, Trucks and Puzzles* is, in reality, *Games, Tricks and Puzzles.*

WHEW!

PCC
p.o. box E
menlo park CA
94025

★

people's computer company

$1~ ✸ 56 PAGES

in this issue:

Robot Pets

Computers & Space

PCC Interviews Ted Nelson

Tiny Languages

Teaching Kids Using Conversational Programming

High School Computers

FORTRAN Man

& Announcements ✸ Games ✸ Surveys
Books ✸ Calculators ✸ Letters

vol.5 no.4 jan~feb77