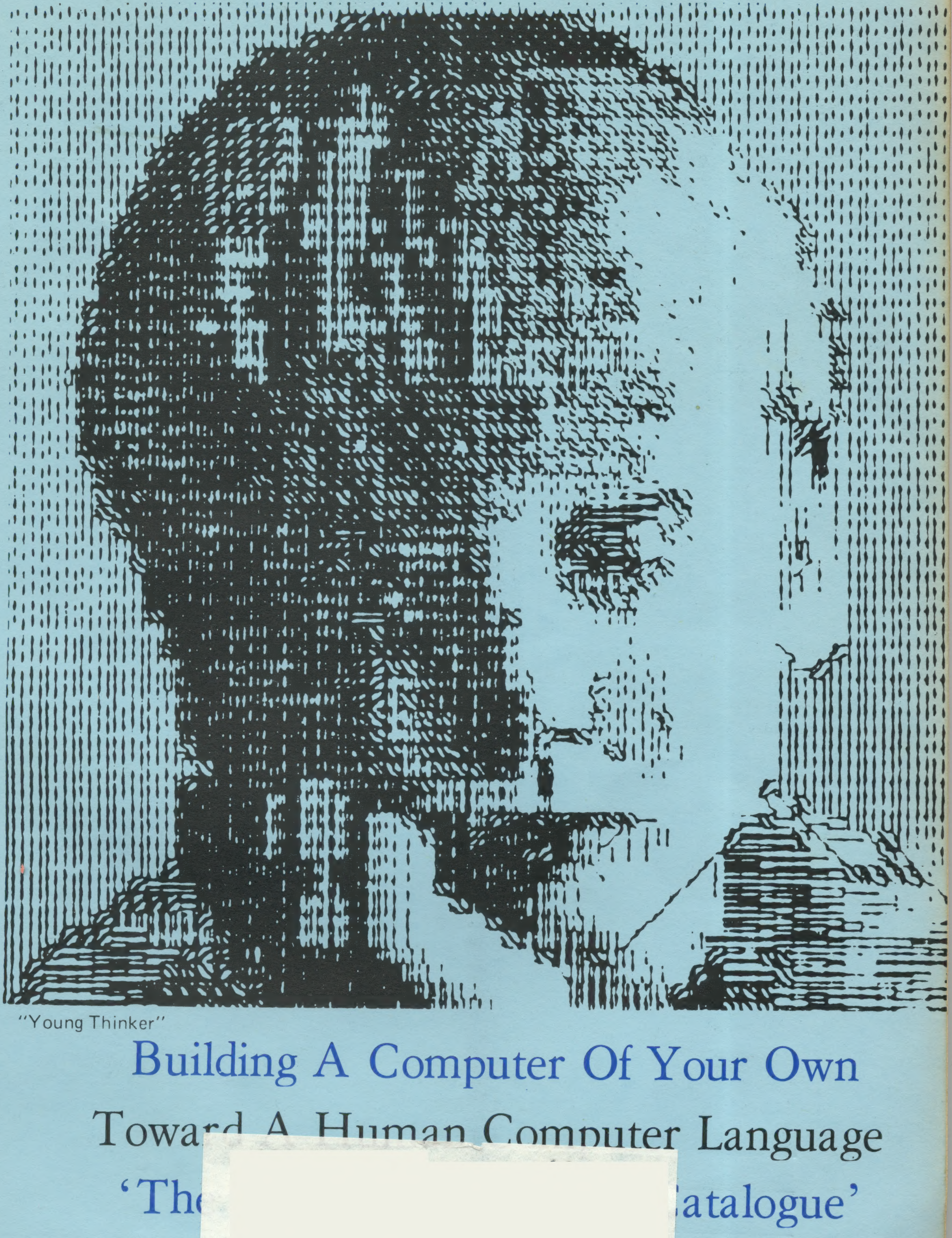
creative computing

the magazine of recreational and educational computing

Sep-Oct 1975

$1.50

COMPUTER LITERACY ISSUE



"Young Thinker"

Building A Computer Of Your Own

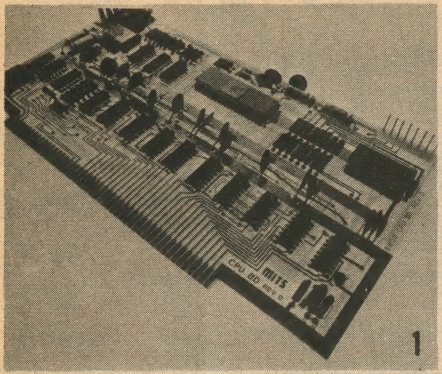
Toward A Human Computer Language

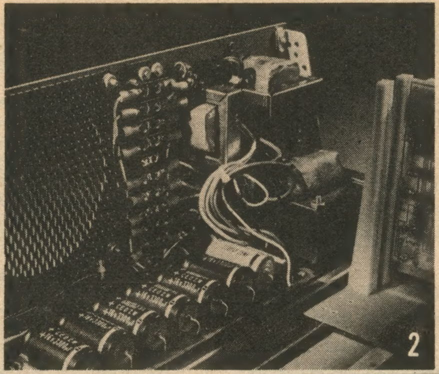
'The [illegible] Catalogue'

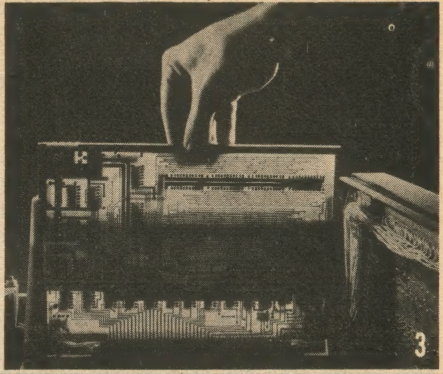
Puzzles & Problems Without A Computer

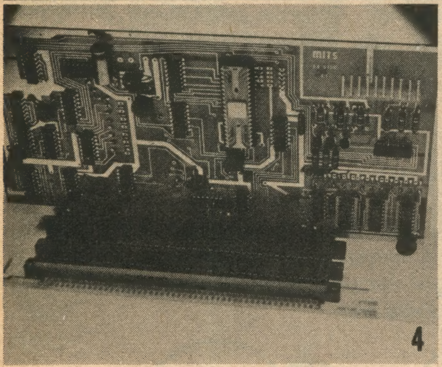
Games: Wumpus, Schmoo, Civil War

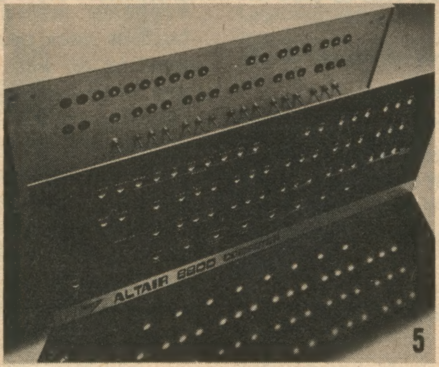
INSIDE the Altair Computer

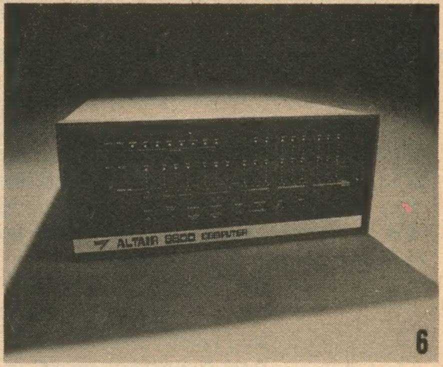

1


2


3


4


5


6

1. Central Processing Unit (CPU) Board. This double-sided board is the heart of the Altair. It was designed around the powerful Intel 8080 microprocessor—a complete central processing unit on a single LSI chip using n-channel silicon gate MOS technology. The CPU Board also contains the Altair System Clock—a standard TTL oscillator with a 2.000 MHz crystal as the feedback element.

2. Power Supply. The Altair Power Supply provides two +8, a +16 and a −16 volts. These voltages are unregulated until they reach the individual boards (CPU, Front Panel, Memory, I/O, etc.). Each board has all the necessary regulation for its own operation.

The Altair Power Supply allows you to expand your computer by adding up to 16 boards inside the main case. Provisions for the addition of a cooling fan are part of the Altair design.

3. Expandability and custom designing. The Altair has been designed to be easily expanded and easily adapted to thousands of applications. The basic Altair comes with one expander board capable of holding four vertical boards. Three additional expander boards can be added inside the main case.

4. Altair Options. Memory boards now available include a 256 word memory board (expandable to 1024 words), a complete 1024 word memory board, and a 4,096 word memory board. Interface boards include a parallel board and 3 serial boards (RS232, TTL and teletype). Interface boards allow you to connect the Altair Computer to computer terminals, teletypes, line printers, plotters, and other devices.

Other Altair Options include additional expander boards, computer terminals, audio-cassette interface board, line printers, ASCII keyboards, floppy disc system, alpha-numeric display and more.

5. All aluminum case and dress panel. The Altair Computer has been designed both for the hobbyist and for industrial use. It comes in an all aluminum case complete with sub-panel and dress panel.

6. It all adds up to one fantastic computer. The Altair is comparable to mini-computers costing 10-20 thousand dollars. It can be connected to 256 input/output devices and can directly address up to 65,000 words of memory. It has over 200 machine instructions and a cycle time of 2 microseconds.

You can order the Altair Computer by simply filling out the coupon in this ad or by calling us at 505/265-7553. Or you can ask for free technical consultation or for one of our free Altair System Catalogues.

PRICES:

Altair Computer kit with complete assembly instructions $439.00
Assembled and tested Altair Computer $621.00
1,024 word memory board ...... $97 kit and $139 assembled
4,096 word memory board .... $264.00 kit and $338.00 assembled.
Full Parallel Interface board .... $92.00 kit and $114.00 assembled.
Serial Interface board (RS232) .... $119.00 kit and $138.00 assembled.
Serial Interface board (TTL or teletype) ..... $124.00 kit and $146.00 assembled
Expander Boards $43.00 kit and $57.00 assembled.

NOTE: Altair Computers come with complete documentation and operating instructions. Altair customers receive software and general computer information through free membership to the Altair User's Club. Software now available includes a resident assembler, system monitor, text editor and BASIC language.

MITS/6328 Linn NE, Albuquerque, NM, 87108 505/265-7553

MITS
"Creative Electronics"

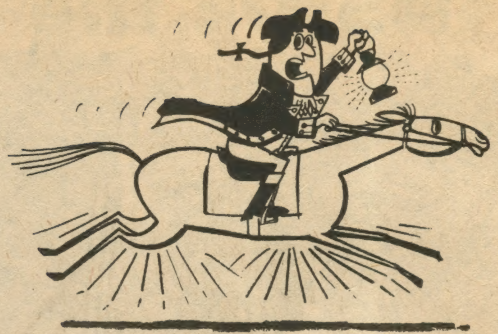
Prices and specifications subject to change without notice. Warranty: 90 days on parts for kits and 90 days on parts and labor for assembled units.

MAIL THIS COUPON TODAY!

☐ Enclosed is check for $______
☐ BankAmericard #______
☐ or Master Charge #______
☐ Credit Card Expiration Date ______
☐ Altair Computer ☐ Kit ☐ Assembled
☐ Options (list on separate sheet)
Include $8.00 for postage and handling.
☐ **PLEASE SEND FREE ALTAIR SYSTEM CATALOGUE**

NAME______
ADDRESS ______
CITY______ STATE & ZIP______
MITS/6328 Linn NE, Albuquerque, NM, 87108
505/265-7553

Notices, etc.



RENEW NOW!!

Some absolutely FANTASTIC stuff is coming up in Volume 2 of *Creative Computing.* One issue will be entirely devoted to games and puzzles, another to artificial intelligence and future computers, still another to the computer in language arts.

Also, all subscribers will receive in place of the May-Jun 76 issue, a fabulous, full color book "Computer Artists on Computer Art." The bookstore price will be at least $5.95, but subscribers will receive this at no extra cost as part of their subscription!

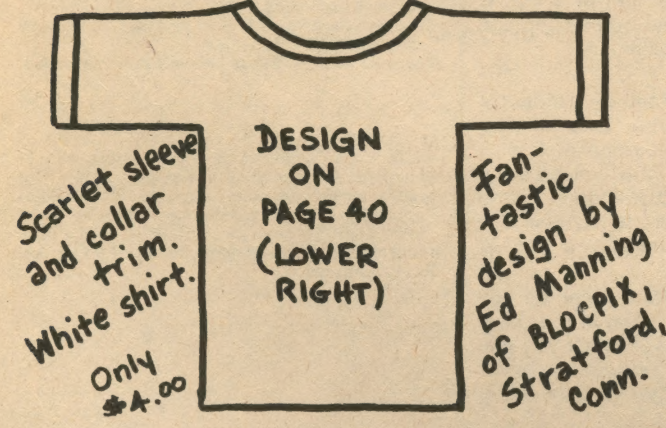
BACK ISSUES ON MICROFILM

If there is sufficient demand, Xerox University Microfilms will make *Creative Computing* available on microfilm. Volume 1, Numbers 1 and 2 are no longer available in hard copy and No. 3 is virtually sold out, so if you want these now or if you might want complete volumes in the future (6 issues), please write directly to:

Ms. Candace Gillen
Xerox University Microfilms
300 North Zeeb Road
Ann Arbor, MI 48106

STOP PRESS! CREATIVE IS NOW AVAILABLE FROM UNIV. MICROFILMS!

creative computing T-SHIRT !



T-SHIRT ORDER FORM

Please RUSH me a *Creative Computing* T-Shirt today. Circle size S M L XL. Enclose cash, check, or money order for $4.00 (includes postage in U. S. Add 50¢ postage to Canada, $1.00 to other countries).

Name ______________________

Street ______________________

City ______________ State ____ Zip ______

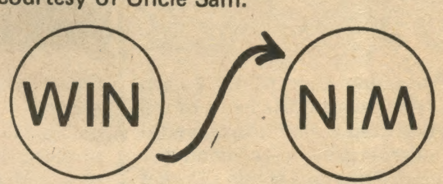
Return form to Creative Computing T-Shirt, P.O. Box 789-M, Morristown, NJ 07960 USA.

FREE BUTTONS!!

In conjunction with President Ford's "Improve the Economy with Singing" Program, millions of "WIN" buttons were printed. Yes, millions! Nobody quite knows what to do with these embarrassing quantities of buttons and in most large cities the Federal Office of Economic Opportunity or its equivalent will give hundreds of these buttons free to anyone who asks for them. (You may have to do some diligent searching by phone to find the right agency in your area).

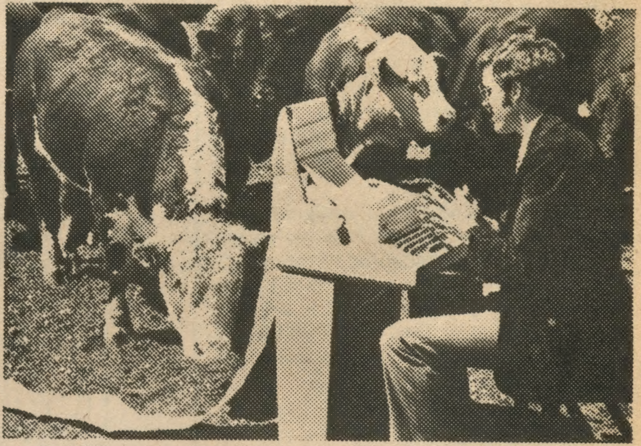
Well, fans, did it ever occur to you that WIN up-side-down spells NIM, a popular computer game. So get a hundred NIM buttons for your school or computer center today, courtesy of Uncle Sam.

—DHA



COMPUTER LITERACY HELD OVER!

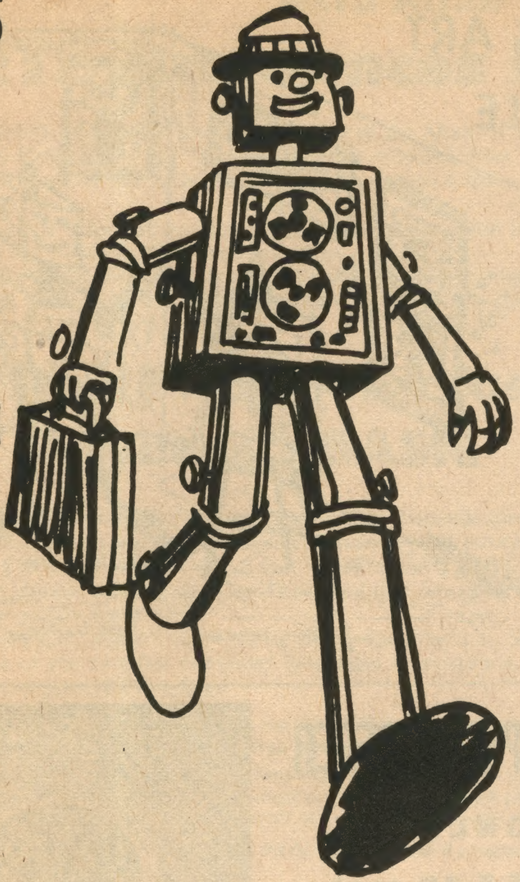
Due to the agency of an ill-natured fairy, the USPS, or who knows what, Walt Koetke's "Problems for Creative Computing" on computer literacy arrived a few days after this issue was typeset. Hence, it will appear next issue. Watch for it!



Satellite terminal connected to the University of Wyoming's Sigma 7 computer over an unusual line installed by Mountain Bell. Joe Wenger, an Industrial Management major from Evanston, Illinois is one of four Commerce and Industry students who manned the University of Wyoming's exhibit at the Wyoming State Fair at Douglas, August 26 - September 1, 1974.

The computer is programmed to answer questions in 18 different areas. Running is a program of interest to ranchers to provide nutrients and diet balance for five different types of cattle. (Photo by Rasmussen, University of Wyoming)

Creative Computing



In This Issue of CREATIVE COMPUTING

SPECIAL COMPUTER LITERACY ISSUE

ARTICLES

NEWS, NOTES, SOURCES

THINGS TO DO

COMPUTER GAMES

FOOLISHNESS

DEPARTMENTS

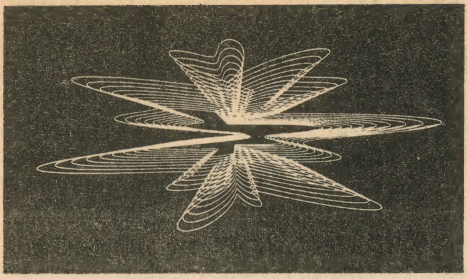
Volume 1 Number 5

CREATIVE COMPUTING is published bi-monthly by Ideametrics, P. O. Box 789-M, Morristown, NJ 07960. Second class postage paid at Morristown, NJ. Subscriptions: Institutional 1-year $15, 3-year $40; Individual 1-year $8, 3-year $21; Student 1-year $6, 3-year $16. Foreign postage $1.50 per year additional. Single copies $1.50.

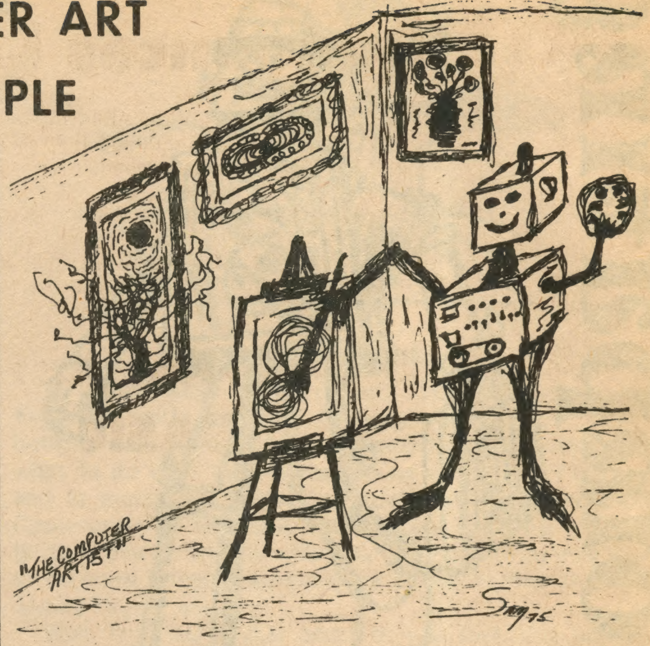
THE COVER

The cover design "Young Thinker" was produced by a programmer (sorry, don't have his or her name) at California Computer Products, Inc. It was plotted on a CalComp 502 plotter with a step size of .01 inches.

COMPUTER ART FOR PEOPLE

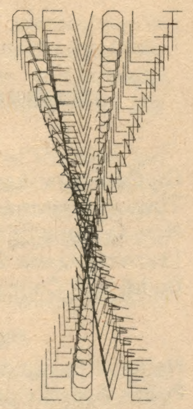


SONIC BOOM | 13 x 21 | blue on white



This offer is good only thru October 1975.

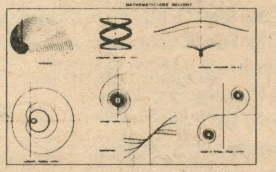
ORDER TODAY!



LOVE REVOLT | 18 x 24 | black and white on multi colors

FLORES en FORTRANES | 18 x 24 | red, green and blue

FREE!
MATHEMATICIANS' DELIGHT



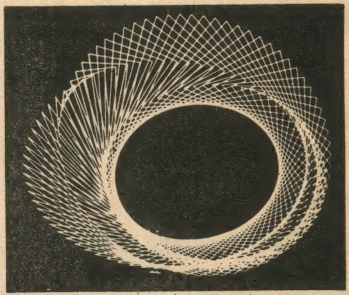
Mathematicians' Delight will be sent free with any order of 2 or more posters.

SPECIAL OFFER

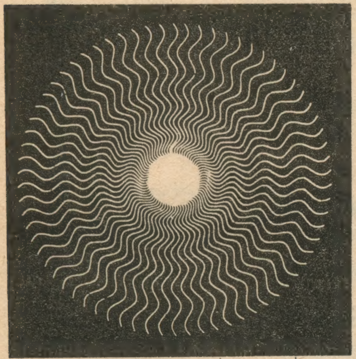
ONLY....

$2.50 each P.PD.
OR ANY 5 ITEMS
only $11.00 P.PD.

COMPUTRA
Box 608
Upland, Indiana
46989 U.S.A.



CRIMSON COSMOS | 16 x 20 | white on red



RA large Fluorescent orange | 21 x 21 |

O! TANNENBAUM — 3 5/8 x 7 1/4 — forest green on gold foil.

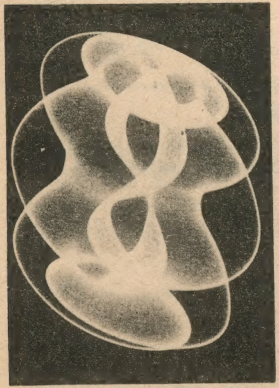
NUMERICAL NOEL — 3 5/8" x 7" — midnight blue on gold foil.

ALL CARDS ARE HAND SILK SCREENED ON QUALITY GOLD CARD STOCK AND HAVE THE FOLLOWING GREETING PRINTED INSIDE IN CHRISTMAS SCRIPT.

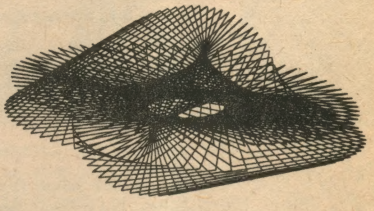
CHRISTMAS GREETINGS
and best wishes for the New Year

CARDS ARE PACKAGED TEN IN A BOX.

GRAFIKEN III | 24 x 30 | black and white



GRAFIKEN I | 24 x 30 | black and white



UNIVERSE | 9 x 21 | dark red on gold

MITS Altair Computer Report II

MITS Announces Lower Memory Prices!

On July 1, 1975, **MITS lowered the price** of the Altair 1K Static Memory Card (88-1MCS). The kit price was dropped from $176 to **just $97** while the assembled price was dropped from $209 to $139.

This price reduction was made possible by a reduction in the price of the Altair 1K 8101 memory chips.

Also affected was the price of 88-MM 256 byte (word) memory modules. The $53 kit price was lowered to just $14 and the $61 assembled price to $26.

Altair BASIC—Not Just Anybody's BASIC

Altair BASIC is an easy-to-use programming language that can solve applications problems in business, science and education.

You will find that with only a few hours of using BASIC that you can already write programs with an ease that few other computer languages can match.

Altair BASIC doesn't compromise power for simplicity. While it is one of the simplest computer languages in existence, it is also a very powerful language.

ALTAIR BASIC comes in three versions. The first of these is a 4K BASIC designed to run in an Altair with as little as 4,000 words of memory. This powerful BASIC language has **6 functions** (RND, SQR, SIN, ABS, INT, and SGN) in addition to **15 statements** (IF . . . THEN, GOSUB, RETURN, FOR, NEXT, READ, INPUT, END, DATA GOTO, LET, DIM, REM, RESTORE, PRINT, STOP) and **4 commands** (LIST, RUN, CLEAR, SCRATCH).

The second ALTAIR BASIC option is the 8K BASIC designed to run in an Altair with as little as 8,000 words of memory. This BASIC language is the same as the 4K BASIC only with **8 additional functions** (COS, LOG, EXP, TAN, ATN, INP, FRE, POS) and **4 additional statements** (ON . . . GOTO, ON . . . GOSUB, OUT, DEF) and **1 additional command** (CONT). This BASIC has a multitude of advanced STRING functions and it can be used to control low speed devices—**features not normally found in many BASIC languages.**

The third ALTAIR BASIC is the EXTENDED BASIC version designed to run on an Altair with as little as 12,000 words of memory. It is the same as the 8K BASIC with the addition of PRINT USING, DISK I/O, and double precision (13 digit accuracy) add, substract, multiply and divide.

Altair BASIC is only the beginning. MITS is currently engaged in an extensive software development program. Other software now available includes an Assembler, System Monitor, and Text Editor.

Altair software comes with complete documentation.

One Month Specials

The Altair Users Group is quite possibly the largest computer hobbyist organization in the World. It is both a means of communication among Altair Users and a method of building a comprehensive library of Altair programs. All Altair 8800 owners are entitled to a free, one year membership in this group.

For one month only, you can become an Associate Member for one year at a reduced rate of $10 (regularly $30). Among other benefits you will receive a subscription to the monthly publication, **Computer Notes,** which contains complete update information on Altair hardware and software developments, programming tips, general computer articles and other useful information.

Now available is the **Altair Software Documentation Book I** which contains technical data on the Altair Assembler, Text Editor, System Monitor and BASIC language software. This documentation is free to purchasers of Altair BASIC. For one month only, it is being offered for only $7.50 (regularly $10).

Offers good until September 30, 1975.

The 1K Static Memory Card contains 1024 bytes of memory with a maximum access time of 850 nanoseconds.

Now ready for production is the new Altair 2K Static Memory Card (88-2MCS) with 2048 bytes of memory. Like the 1K Static Memory this new card contains memory protect features and provisions for disabling the ready.

It has a maximum access time of 850 nanoseconds and is engineered with the finest components available. It is inexpensively priced at $145 kit and $195 assembled.

HARDWARE PRICES:

Altair Computer kit with complete assembly instructions	**$439**
Assembled and tested Altair Computer	**$621**
1,024 Byte Static Memory Card	**$97 kit and $139 assembled**
2,048 Byte Static Memory Card	**$145 kit and $195 assembled**
4,096 Byte Dynamic Memory Card	**$264 kit and $338 assembled**
Full Parallel Interface Card	**$92 kit and $114 assembled**
Serial Interface Card RS232)	**$119 kit and $138 assembled**
Serial Interface Card (TTL or Teletype)	**$124 kit and $146 assembled**
COMTER II*	**$780 kit and $920 assembled**

***The Comter II Computer Terminal has a full alpha-numeric keyboard and a highly readable 32-character display. It has its own internal memory of 256 characters and complete cursor control. Also has its own built-in audio cassette interface that allows you to connect the Comter II to any tape recorder for both storing data from the computer and feeding it into the computer. Requires an RS232 Interface Card.**

SOFTWARE PRICES:

Altair 4K BASIC	**$350**
Purchasers of an Altair 8800, 4K of Altair Memory, and Altair Serial I/O or Audio-Cassette I/O	**ONLY $60**
Altair 8K BASIC	**$500**
Purchasers of an Altair 8800, 8K of Altair Memory, and Altair Serial I/O or Audio-Cassette I/O	**ONLY $75**
Altair EXTENDED BASIC	**$750**
Purchasers of an Altair 8800, 12K of Altair Memory, and Altair Serial I/O or Audio-Cassette I/O	**ONLY $150**
Altair PACKAGE ONE (assembler, text editor, system monitor)	
Purchasers of an Altair 8800, 8K of Altair Memory, and Altair I/O	**ONLY $30**

NOTE: When ordering software, specify paper tape or cassette tape.

Warranty: 90 days on parts for kits and 90 days on parts and labor for assembled units. Prices, specifications, and delivery subject to change.

MAIL THIS COUPON TODAY!

☐ Enclosed is check for $__________

☐ BankAmericard #__________ ☐ or Master Charge #__________

☐ Altair 8800 ☐ Kit ☐ Assembled ☐ Options
Include $8 for postage & handling (list on separate sheet)

☐ Altair Users Group Associate ☐ Software Documentation

☐ Please send free literature

NAME__________

ADDRESS__________

CITY__________ STATE & ZIP__________

MITS/6328 Linn N.E., Albuquerque, NM 87108 505/265-7553 or 262-1951

MITS

"Creative Electronics"

MITS/6328 Linn N.E., Albuquerque, NM 87108 505/265-7553 or 262-1951

A National Computers in Education Conference?

reported by David Ahl

During the 1975 National Computer Conference in Anaheim, a meeting took place which had as its innocent purpose to discuss the overlapping activities of various societies and organizations that have an interest in computers in education. The 30 some odd attendees represented eight societies having a major interest in educational computing and approximately 15 other societies who are on the periphery, but none-the-less have a real interest in the subject. The eight major societies represented were: ADCIS, AEDS, COED (IEEE), NAUCAL, SIGCAI (AERA), SIGCAS (ACM), SIGCUE (ACM), and SIGCSE (ACM).

Following some preliminary descriptions of each major group represented, and some of the others too, the attendees got into discussion about some of their overlapping activities (and publications of which there are at least 20 that appear more-or-less regularly!). The dialogue was, to say the least, spirited, even heated and impassioned at times. It lasted well into the wee hours, was resumed at many breakfast counters the next day, and the day after that.

Two concrete proposals emerged. The first proposal appeared to be endorsed by the overwhelming majority, that is:

1. To publish in one place an informal annotated guide to societies, groups, and publications involved with computers in education. The word "informal" was put in because it seemed that people wanted a brief evaluation of each organization, Whole Earth Catalog style, in addition to or even in place of the organization objectives as stated in its constitution or bylaws. To my knowledge it was not decided who would publish this guide. I'd be glad to volunteer to publish the information in *Creative Computing* in the "Complete Computer Catalog" column or as a separate section or even a separate booklet. On the other hand I do not feel qualified to write capsule descriptions of each organization and will depend upon reader submittals for this.

The second proposal had a great deal of support, although hardly unanimous. It was:

2. To hold a National Computers in Education Conference. One large faction felt that it was certainly time that we got around to this. After all, if the IFIP can hold a World Computers in Education Conference (Marseilles, Sep 1975), the U. S. certainly should have a national one. Should this be done by AFIPS? Jointly with NEA, NCTM, AASA, etc.? Many questions -- few answers. Another faction felt strongly that small specialized conferences were more productive, promoted more meaningful dialogue, and were certainly more manageable. Organizations, understandably feel strongly about retaining their own individual identity, although one has to wonder whether this may be hampering the bringing about of broad changes and major innovation in the use of computers in education. Again, no answer.

Creative Computing solicits comments from readers on these subjects. In future issues we will present a continuing forum on them.

PUBLIC ACCESS QUESTIONNAIRE

by Larry Press

I am compiling and maintaining a survey of what people are doing in the way of public access computer projects. This would include projects for community service, education, and recreation. If readers are connected with or know of such projects, please answer the applicable questions below; augment with comments, brochures, or literature; and send to Larry Press, 128 Park Place, Venice, CA 90291. A later issue of *Creative Computing* will carry the results of this survey.

1. How many public terminals do you have? Are they owned, borrowed, leased, or donated?
2. What CPU do you use? Is your time owned, borrowed, leased, or donated?
3. Is your project supported by a grant, host institution, user payments, or no one?
4. How many hours per week do you support public access?
   Are resource people available during these hours?
5. Do you teach classes in terminal operation, programming, etc.?
6. How many programs in your public access library?
7. Are users permitted to write programs or are they restricted to using library routines?
8. Which, if any, of the following applications do you support?
   - A. Game playing programs for recreation and familiarization with terminal operation and functional characteristics of the system.
   - B. Local data bases with information on, e.g.
     1. where people work (car pools)
     2. what skills and interests people have
     3. consumer information on local stores-prices, complaints, etc.
   - C. How to do it dialogs with reference to local sources of information, people, materials, etc.
   - D. On-line opinion polls (with capability for questions and issues to be raised by citizens) on specific local issues such as a proposed zoning change.
   - E. On-line suggestion and complaints "box".
   - F. Dialogs to guide people in the preparation of forms required by various agencies.
   - G. Para-legal/ombudsman dialogs, such as:
     1. How to handle your divorce.
     2. How to get building permits.
     3. How to interpret warranties on retail goods and services.
     4. How to do your own escrow.
     5. What are tenants/landlords rights.
     6. Detecting discrimination in the granting of credit.
     7. Loan payment/interest/term computations.
     8. Help locating sources of credit.
     9. Income tax preparation.
     10. Help in obtaining favorable utility rates.
   - H. Consumer guidance dialogs, e.g., how to buy a used car or appliance, with information on local suppliers, consumer reports, etc.
   - I. Local want ads.
   - J. Dialog to route people to agencies, e.g., drug rehabilitation, family planning, welfare, etc.
   - K. Calendar of community events.
   - L. Field trips from local schools.
   - M. Data Processing for community service organizations.

Input/Output

May-Jun Issue – Pro and Con

Dear Editor:

We have been very impressed with *Creative Computing* as an interesting and enjoyable magazine for elementary and secondary school teachers and students – in fact, we are in the process of trying to get a bulk subscription for our TIES district schools. We were, however, most disturbed to see the "Future City Comics" in the May-June issue. Not only were they irrelevant to the topic of educational computing, but they contained material which we felt would be offensive to many teachers, students, and parents in our districts. It seems one of our districts also felt this was true as they cancelled the four subscriptions they had sent us. We hope that in future issues you will take this into consideration and will not let a few off-color pages damage the reputation of what is otherwise a terrific magazine.

Linda Barry, Marie Keljik
Donald Holznagel, Norman Thompson
TIES, St. Paul, MN

Dear Editor:

I am absolutely appalled at the material appearing in Creative Computing of the May-June issue concerning the article titled "City of the Future." Some of the material is crude, in poor taste, inappropriate, filthy, and has no place in an otherwise excellent magazine.

Jerry Hansen
Detroit Country Day School

Dear Editor:

I've just received a sample issue of your May-June 1975 "Creative Computing" magazine and I'm already dissatisfied – with that orange insert titled "Our Face Is Red". I couldn't find anything offensive about anything on any page. I like your magazine as it's put together now.

James A. Muysenberg
Blytheville, AR

Dear Editor:

My compliments to you and your staff on a highly informative and exciting magazine! I feel confident that the ideas presented in your articles will stimulate my computer science students and keep their minds active. Should your articles continue to be as exciting as those in volume No. 4, I would be more than willing to submit money for a lifetime subscription.

Keep up the good work!

B. K. Metcalfe
Winnipeg, Manitoba

Dear Editor:

I have been very busy establishing a small computer network here among our local high schools. I continue to enjoy *Creative Computing* and have actively encouraged all users of our network to subscribe. Each issue is better than the last and I will do whatever I can to support the magazine.

James S. Vinson
Univ. of North Carolina

Robert Crumb's cartoon feature "City of the Future" and the illustration on Page 19 of the May-Jun issue obviously offended a number of readers, although some other readers loved it. I intend to pretty much steer clear of this particular type of material in the future, however, I feel obligated to share with you my motivation for including it. I believe that we should all be aware of the fact that computers (and technology in general) have a very mixed image in the world today. While it would be delightful if everyone presented their views in a well written reasonable manner, many people don't. And I'm not just thinking of cartoon strips with offensive language. I'm thinking about fire bombing of computer centers, destruction of magnetic tape files, and the use of technological techniques to commit massive fraud. We just have to recognize that many people regard computers with HATE and FEAR. And if you want that image to change, you are the ones who can do something about it. Perhaps an offensive comic strip isn't the best vehicle to get the message across that many people see technology as BAD; I'll look for other ways such as the "Bugs" column in the Mar-Apr issue. But you can count on this: Creative Computing *is going to present all sides of computers and technology – good, bad, beneficial, wasteful, mind expanding, and dehumanizing. I firmly believe that knowledge of the undesirable possibilities will help us achieve the best overall results with our computers and technology. –DHA*

Dear Editor:

Thank you for the publication of my article on the Universal Word Game (Mar-Apr 75, p. 49). As usual, a small after-the-fact bug turned up some months back, and I had forgotten to note the fact to you. I would be most grateful if you might print a small correction box somewhere in a future issue – line 860 should read 'if c3 <= 20 then 1120.' It won't cause much trouble unless someone gets exactly 20 wrong, but sloppiness is not to be tolerated in any case.

I might also add that the program is now up on the ECN system (formerly EIS) in French, Spanish and Latin, running under Jeul, Juegol and Ludusl, respectively, in the three star library).

Barney M. Milstein
Stockton State College, NJ

more

BASIC Language – Pro and Con

Dear Editor:

It was with pleasure that I opened my first issue of *Creative Computing*, only to feel disappointment as I closed it. There is something more to computing than BASIC running on a mini with TTY. If that's as far as educational and fun computing is destined to go, forget it.

I realize that there is a current base of users heavily into mini-BASIC, but a magazine such as yours should be avant-garde. One would never know there was such a thing as a microcomputer from your March-April issue! Micros are the vehicle for fun and games computing of the future. And for heavier educational areas, micros are perfectly at home interfacing a floppy. Higher level languages for micros exist and are developing. What I really object to is the rule framework of your Call for Games. Please ask for flowcharts. Also admit languages other than BASIC and I/O other than TTY's.

To be creative requires some freedom. Let's not put everyone in a straitjacket and then ask for creativity.

A. D. Robbi
Hopewell, NJ

Dear Editor:

I have just received my third issue of *Creative Computing*. It is an interesting, informative and challenging magazine. But there is one problem, i.e. your exclusive interest in BASIC as the language for instructional computing. Granted BASIC is the language most prevalent in secondary schools and one in which the editors have a personal interest. But much creative computing is being done in other languages. Why exclude the submission of these programs simply because they are not done in BASIC? I realize the desire for standardization is strong. But many people would like to submit games, simulations or other programs in another language without going through the bother of writing, debugging and listing the programs in BASIC. In some cases this task may be impossible.

A common problem among computer people is the oft held opinion by many that "the language I use is the best" – be that language BASIC, APL, FORTRAN or whatever. Such narrowmindedness frustrates me because my concern as an educator is the communication of good ideas despite the vehicle of expression.

Therefore I urge you not to foster such parochialism in a magazine of national appeal. Many young people will grow up thinking BASIC is the only language, a situation which I find regretable.

Mary Lou Fox
Fairfield University

You'll be seeing more and more of other languages (see "On Languages" this issue), microprocessors and other avant-garde material in Creative Computing. *How about some articles from you readers into these things. –DHA*

Dear Editor:

Please keep up the great work with *Creative Computing*. I have been looking forward to a magazine of this type for a long time, and I am glad you came out with it.

I am planning to donate games and math programs in the near future. Meanwhile, may I offer you the results I got recently from a new program(?).

Iasked the computer about its self-awareness, and it answered:

I COMPUTE, THEREFORE I AM
I CAME, I SAW, I COMPUTED
TO COMPUTE OR NOT TO COMPUTE, THAT IS THE QUESTION

I got more, but that is all that is of interest(!).

Andrew S. Glassner
Highland Park, NJ

Dear Editor:

I am writing both to subscribe to *Creative Computing* and to reply to the letter you printed from George Grossman, the Director of Mathematics of the NYC Board of Education.

Unfortunately Mr. Grossman seems to view the computer only in the light of teaching mathematics, and finds 95% of what is covered in *Creative Computing* as being aimed elsewhere. What he is in fact saying is that out of the 100% of computer oriented material he can only find 5% germain to his narrow circumscribed and stultified conception of what computers are all about. This is very sad, and what is sadder is that this man controls, at least nominally, all computer education in the City school system.

Fortunately his ideas are carefully ignored in the more enlightened parts of the system. At my school our courses cover a wide range of computer science topics from assembler language programming and system architecture, to automata, recursion, string processing and compiler writing. Light does shine in New York.

Creative Computing shows potential. It seems to be floundering as to what its direction ought to be, but if anything it errs in the direction of too many topics, rather than, thank heaven, too few. I will read the issues that I get and should I figure out exactly what I think needs changing, I will let you know. Until then, good luck!

Name withheld by request

Dear Editor,

I noted with some interest your article on multiple precision arithmetic. For some time, I have had a personal mania for computing huge factorials exactly, and it is gratifying to see that I am not alone in my proclivities. I have calculated 10,000 factorial exactly (I think – how could you ever check it?) and think that this may well be the largest one yet computed exactly. In any event, I also learned that there is nothing quite so dull as 7 pages of digits.

A very good reference on multiple precision arithmetic is in Knuth's *Seminumerical Algorithms*. He says all that you would normally need to know. Although it might be tough going for computing neophytes, the book is well worth the effort. Especially interesting is the part on modular arithmetic in which it is revealed how to do multiple precision arithmetic without having to do any carries.

Keep up the good work.

John Levine, Student
Yale University

Dear Editor:

Thank you for sending *Creative Computing*. Especially enjoyed the book review, and the cartoons. At times the prose could be more succinct.

Suggest that you arrange to have *Creative Computing* put on board jet airliners.

In future issues I would suggest having a contest to write programs useful in civic work:

1. Construct a tree based on Zip codes for car pooling or passing news by phone.
2. Sociogram to place friends in same seat on a bus.
3. Scheduling for volunteers, or shifts.

Or, Sorting a record collection into segments the right length to fit on an 8-track cartridge.

Maybe there is a source of bibliographic information pertaining to omputers in education that would prepare a search for you to publish in the newsletter. It would be nice to have a list of addresses for university newsletters in the field of computer education.

I hope someday you can find someone to write a story on Educational Testing Services, Inc., a *very* lucrative business.

Chris Connors
Berkeley Hts., NJ

"Feature" Letter to the Editor

Dear Editor:

I've been meaning to write since your first issue, but finishing college and other things have gotten in the way. Your latest issue (May-Jun 75) contains so much thought provoking material that I simply can't wait any longer, so, here are my thoughts.

First, congratulations on a simply excellent publication! Your content, layout, artwork and direction are just great. To be commended above all esle, particularly this latest issue, is *Creative Computing*'s, DIVERSITY. By the way, I hope you *do* view *Creative Computing* as being a member of the Alternative Press, and not just because you're on newsprint. You're not going to be able to keep a stance of open-eyed diversity and attract the readership you need if you fal! into some worn-out rut. If newsprint's respectability causes trouble, then come out in microfiche also. That would be nice for permanence, anyway.

Re: David Ahl's editorial and Gregory Yob's esthetic and philosophical comments on GEOWAR, I couldn't agree more. My undergraduate major is philosophy and my "heros" are John Muir, Henry Thoreau, Mohandas Gandhi and Bertrand Russell – facts which always astonish people since I appear to live in our computer center. I can see why on the surface people are astounded to find that I am a philosophy major. As was pointed out in your third issue, computers have received a very bad press and very, very few people have any real conception of what a computer actually is, on the hardware or software level. [Which raises an interesting question: which *is* the computer, hardware or software? More on this later.]

I have always been deeply disturbed by the proliferation of war games and the tendency for every new technical (or intellectual) advance to be adapted to the purpose of killing. The conduct of our species is what must evolve now. Not physical evolution but intellectual.

Unfortunately, most large R & D budgets have been and are still tied to "Defense." That is what is so hugely disappointing about the demise of the space program. For a while a significant number of people were united behind a peaceful research project of significant scale. Now we are united behind nothing and I am afraid that the answer to the question "Where are we going?" is to hell and that very quickly if we don't regain some unifying objective and goal.

I hope your editorial moves some people to take time out from Star Trek at the CRT and spend some time looking at the real stars. Perhaps I spend too much time reading science fiction, but I can't believe that the future of the human race lies totally on Terra of Sol. We are just going to sit here and stagnate if we don't get off this planet – at least through intellectual contact with extraterrestial life. The current generation of computer fanatics is just the generation to decide that this is a worthwhile goal, and it may be the last generation capable of making such a choice. By the end of this century we're going to be too busy surviving to notice the stars.

Re: Ed note on page 18. I have doubts about reversals eventually becoming palindromes because of randomness. I am always suspicious of falling back on randomness, because it seems pretty clear to me that there is no such thing as true randomness. Anyway, since the number continually gets larger, doesn't the probability of "randomly" hitting a palindrome get smaller? If I get around to it I'm going to play around with 1675 on the IBM370/168 VS1.7 we're tied into. Might as well use up the money in some of our course accounts in the interest of curiosity.

[Ed note: As Fred Gruenberger and others have pointed out, my speculation that reversals of 196 could become palindromic due to randomness is dead wrong because the size of the number is increasing at every step – DHA]

*** SPROING. 18 HOUR INTERLUDE. ***

Enclosed is a copy of a program I just wrote which takes the number 196 through 12066 reversals to produce a 5000 digit number without ever being palindromic. I apologize for its being written in 370 assembly language, which has got to be the most exclusive "language" around, but we have to pay 22¢ a cpu second and I had to have the most efficient program possible. With slight modification this program would handle results up to 8,000,000 digits in length (the system has 16 meg) but there is this slight problem of paying for it. I may have a go at taking it out to 10,000 digits. There is always that nagging suspicion that the next reversal, or maybe the next, or surely the one after that . . .

Re: Things I'd like to see discussed in *Creative:* More about ways in which people have used computers to investigate the world and themselves, i.e. situations in which someone said "I wonder if . . ." and then used a computer to help find out. A large selection of "I wonder ifs . . ." without answers would be nice, too. How about a list of prodigious problems which might lend themselves to computer solution if only someone looks at them in the right way?

How about some input on the average user level on the proliferation of languages? If you're using a computer as a tool (or a friend) to solve problems, do you get more done if you *know* BASIC or ANS FORTRAN inside out or if you have an acquaintance with PL/I or GIBBERISH II? What do your readers like/dislike in their languages-systems? etc. *[Ed Note: "On Languages" will be a regular forum to discuss just these issues – DHA]*

How should computer use fit into a sane lifestyle; Does computer use overall create or solve problems? etc.

Having been myself a member of a programming team competing in a contest (University of MO at Rolla – March 29) I have wondered if such contests encourage "good programming." Should they?

How does someone not in an educational institution get time on a computer? Does anyone sell time (reasonably) to individual users?

Re: Your upcoming issue "The Computer Threat to Society": Change is always a threat to staying the same. I hope in this issue someone will follow up on the thoughts expressed in the next to last paragraph of David Ahl's editorial.

Also, looking over your staff listing, I notice that you are not too well represented in the midwest. Things do occasionally happen out here, even without the benefit of wall-to-wall people. You really should get someone out there – talking is still the best medium for information exchange.

In any case – good luck! I'll keep trying to talk people into subscribing.

Peace and Love,
John R. Lees, Jr.
Fulton, MO

[Ed note: John is now a regular Reviews Editor for us. -DHA]

Dear Editor:

Many mathematicians feel that computer programming can and should be taught to students in the upper elementary grades. The most difficult problem facing someone in this position is the generation of problems that are appropriate for a computer, conceptually easy enough for students to grasp, and engage their interest.

Included with this letter are some problems that my 6th-grade students find interesting and helpful in demonstrating how a computer can help them solve problems that they might meet elsewhere. Most of the problems require only a few storage units and maybe 1 loop, but this has been plenty for my students to handle.

Charles A. Reeves
6th-grade Math/Science Teacher
Developmental Research School
The Flordia State University
Tallahassee, Florida 32306

Many of Charles' problems can be found in the "Problems for Creative Computing" section of this and future issues. If other readers have favorite problems, please send them to us. – DHA

What's Wrong With the Little Red Schoolhouse?

I have a couple of observations on education and learning:

1. Kids learn best from other kids, probably outside of school. Some of the best learning probably occurs in the street!
2. Although textbooks are written by the best minds and are cram full of enlightened information of allegedly great worth and importance, they never get stolen.
3. Kids spend a fantastic amount of time in front of the boob tube. Indeed, at the time of entering first grade, the average child has spent more time with the TV than with both of his parents.
4. Motivation is much more important than teaching method or style of delivery.
5. Kids don't respect people who talk down to them, try to use their slang without knowing what it means or who are in any way artificial or self-important.
6. Kids' minds are unencumbered by constraints of what can and can't be done. Kids will try just about anything.
7. Learning by discovery, doing, or manipulating sticks with you far better than learning by reading about.
8. Learning to learn is infinitely more important than learning facts and data.
9. Isaac Asimov, Herbert Simon, Marshall McLuhan, and Herman Kahn will have more impact on the future than all the textbooks in print.
10. Computers are the most powerful tool man has ever invented and the most awesome responsibility he has ever faced.
11. Education has become relatively less efficient than practically any other aspect of our economic or social development.

What does all this mean? In short, it means that the little red schoolhouse which we hold so dear to our hearts is no longer satisfactory. Not only just unsatisfactory, but in urgent need of change. No longer can we rely upon the teaching approaches of yesterday; in fact, we can't even rely upon the ones of today. Half of the piecemeal, one-at-a-time changes we're making in the educational process are out of date before they're even implemented widely.

Compounding the slowness of educational innovation is the fact that many of the new systems introduced over the last 20 years using the latest technology have fallen flat on their faces. This is true of things like teaching machines, drill-and-practice CAI, language labs, closed-circuit TV, and perhaps now even the "new math." These shortcomings and failures have given the traditionalists all kinds of ammunition to shoot down other new things on the horizon and, indeed, have imbued nearly every educator with a hearty skepticism for educational technology.

Those educators who have revolted against drill-and-practice PI and CAI believe that getting a pile of data into a kid's head is not as important as teaching underlying structure and modes of self-learning. Challenging and stimulating students, and then providing them with an information-rich environment in which they can seek their own solutions, is seen as a better way to teach. On the other hand, opponents of these new ideas fear that free-form education will result in gaps in the basic knowledge considered essential in conventional education.

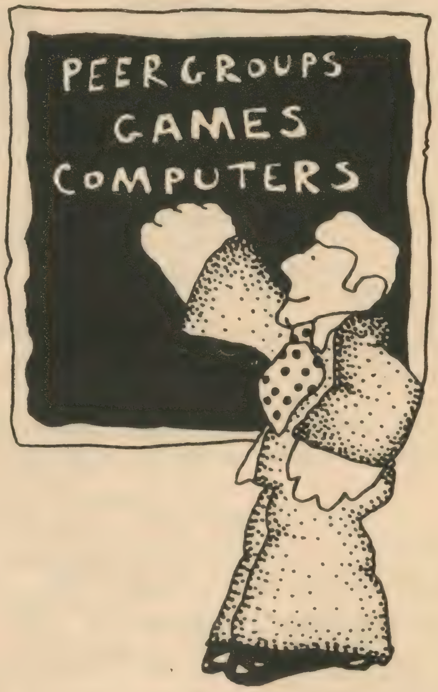
Determining who is right and who is wrong is probably not very productive, although many researchers are engaged in that happy pursuit. More to the point, when one looks more deeply, it's not clear that all the technological innovations did fail. Some of them undoubtedly did. However, what's much more evident is that we probably don't know how to measure the results. Or going one step deeper, it's not at all clear that we have even established meaningful objectives.

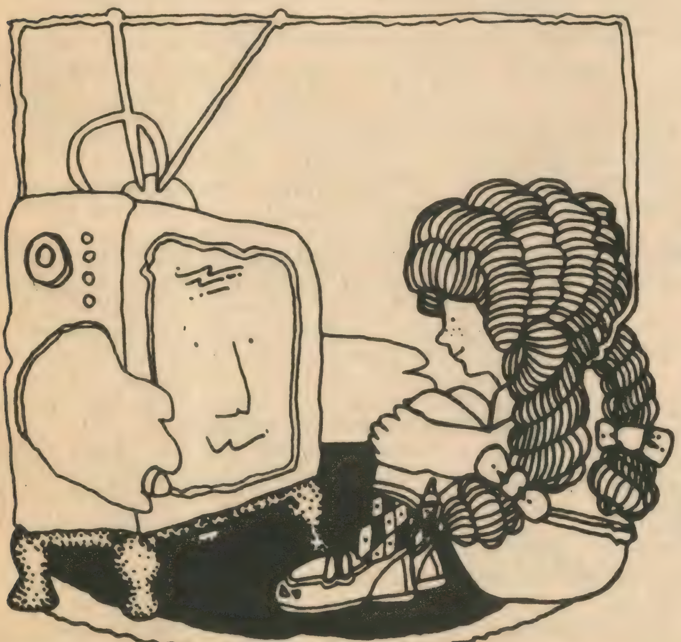
Objectives must be broadly stated and must not only be relevant to the world of today but to the world of the future. Knowledge is changing and advancing so rapidly that we must expect objectives to change, or conversely, be formulated in a broad enough fashion to keep up with technological advances. The Mager-style behavioral objectives don't begin to meet the need. Education today needs more than small behavioral steps. It needs objectives that are dynamic and can be expected to change over time; objectives that are stated in an entirely new way.

Objectives must be devised to lead young minds through an imaginative exploration of the jungle of political, social, psychological, and ethical issues that will confront them as adults. What is the objective which will foster decision-making ability under uncertainty? What are the objectives which elucidate the ways in which technology and values will interact in the world of tomorrow? How does one measure whether one has learned to learn? I'm not saying these objectives are impossible to devise. Indeed, these are the kinds of things that education must focus on, and these are the objectives that must be devised.

And while educational objectives must change and grow, so must its methods. One-third of the observations stated at the outset say, in one way or another, that the medium really is the message. Today, learning to read from a book can be substantially enhanced with the "Electric Company" on TV and with Moore's Talking Typewriter. For some aspects of education today the book is hardly an acceptable alternative at all; yet for other aspects, it's still the best approach.

For example, learning to fly an airplane is best done on a flight simulator. Learning to create a movie is best done by actually creating one. Learning about political action is best done by observing and participating. Learning about resource management and resource utilization is best done by manipulating computer simulation models. Learning a logical approach to problem solving is best done by breaking a real problem down into manageable pieces, flowcharting it, and programming it for a computer. Learning an appreciation for literature is best done by reading books.





So, in addition to broad, dynamic objectives, education also needs some new methods. Not one to the exclusion of others, but a whole potpourri of techniques. Things like peer teaching, computer games and simulations, free learning environments and piles of motivation. Motivation is strongest when it comes from real involvement and genuine accomplishment. We have to let kids work with real tools on real problems, not a bunch of contrived textbook situations. We have to give kids tools far more powerful than we think they can possibly use. The results will be unbelievable!

David H. Ahl

Reprinted with permission from *EDU* No. 10 published by Digital Equipment Corp.

COMPLEAT COMPUTER CATALOGUE

You've heard of *Big Rock Candy Mountain, Whole Earth Catalog, Whole Earth Epilog, Rain, Yellow Pages of Learning Resources,* and *Tom Swift and His Electric English Teacher.* Well here we go again with another catalog/access handbook/annotated bibliography. The idea of this, *The Complete Computer Catalog,* is to provide source information about resources for learning about and using computers effectively. The emphasis will be on applications, software, and learning aids rather than computer hardware and systems.

This catalog is for:

1. People who have never seen a computer and want to learn what all the fuss is about,
2. People who are just meeting their first computer either in school or on the job,
3. Experienced computer people who want to use their machine more effectively.

Between two and eight pages of *The Complete Computer Catalog* will appear in each issue of *Creative Computing.* Eventually all the entries may be collected together and published as a book (or catalog).

We would welcome entries from readers on any worthwhile item related, even distantly, to computers. Please include the name of the item, an evaluative description, price, and complete source data. If it is an item that you were acquainted with over one year ago, please check with the source to make sure it is still available at the quoted price.

Our catalog categories are not cast in concrete. However, at the moment they are:

- Books and Booklets
- Periodicals and Journals
- Games
- Learning Aids and Films
- Computer Software
- Computer Hardware
- Groups and Associations

Send contributions to The Complete Computer Catalog, c/o *Creative Computing,* P.O. Box 789-M, Morristown, NJ 07960.

BOOKS AND BOOKLETS

COMPUTERS AND THEIR IMPACT ON BUSINESS AND SOCIETY

Course outline and text for 40-hour course "to present a balanced view of what a computer is and how it can be used, and also set pupils thinking about what computing could mean to them as people.". This book was prepared by the National Computing Centre, Ltd., Manchester, England. U. S. price unknown, but it is available from:

Hayden Book Co., 50 Essex St., Rochelle Park, NJ 07662.

COMPUTERS IN THE ELEMENTARY SCHOOL: A COURSE FOR TEACHERS

This 32-page booklet consists of a fairly detailed course outline to give elementary teachers an introduction to computers and electronic calculators. It includes sections on problem-solving, calculators, computers, programming, future implications, and classroom strategies. $1.50.

Curriculum Group, Oregon Council for Computer Education, Dept. of Computer Science, Univ. of Oregon, Eugene, OR 97403.

SCIENCE FOR SOCIETY

An extensive 110-page bibliography for students, faculty, and lay groups. Most references 1971 and later – mostly newspapers and magazines. Price unknown.

Commission on Science Education, AAAS, 1776 Massachusetts Ave., NW, Washington, DC 20036.

TOM SWIFT AND HIS ELECTRIC ENGLISH TEACHER

A "Whole Earth Catalog" type book by G. Howard Poteet for English teachers. This oversize (11" x 15") handbook is chock full of teaching techniques, ideas on how to use new and old equipment, quotations, hardware, software, etc. Excellent sections on the use of radio, TV, filmstrips, and drama. Source and price info included. $4.95.

Pflaum/Standard, 8121 Hamilton Ave., Cincinnati, OH 45231.

THE WHOLE WORD CATALOG

A practical collection of assignments for stimulating student writing, designed for both elementary and secondary level. Activities included to foster personal writing, collective novels, diagram stories, fables, spoofs, and language games. Contains an annotated bibliography. Many computer people do not write easily or well; this will help. $3.00.

Teachers & Writers Collaborative, 186 West 4th St., New York, NY 10014.

LAYMAN'S GUIDE TO THE USE OF COMPUTERS

The emphasis of this guide is on instructional applications of computers. It tends to have a data processing slant and isn't as comprehensive as some other guides (OCCE, for example) but for $3.00 presents a reasonable overview (although the price should really be more like $1.50).

Association for Educational Data Systems, 1201 Sixteenth St., NW, Washington, DC 20036.

MATHEMATICS CURRICULUM REPORT

Report (October 1974) discusses changes in mathematics programs in junior high and middle schools. Discussion of pocket calculators, computers, statistical concepts, practical problem-solving, and various university and vendor curriculum projects. Ask for Vol. 4, No. 1, "Mathematics Programs are Changing" – 50¢ prepaid.

NASSP, 1904 Association Drive, Reston, VA 22091.

RANGER 'RITHMETIC

Series of booklets for first to eight grade with 20 some odd problems in each around the theme of forest conservation. Many problems suitable for calculator or computer solution. Ask for the one for your grade level. Single copy free. Also ask for list of teaching aids.

Forest Service, U. S. Dept. of Agriculture, Washington, DC 20250.

CURRICULUM MATERIAL PRODUCT CATALOG

This booklet lists virtually all of the materials for using computers in the curriculum published by DEC including the popular Huntington II simulation materials. Pictures of each item, capsule commentaries, price, and ordering information are included. Free.

Education Products Group, Digital Equipment Corp., Maynard, MA 01754.

MAGAZINES, JOURNALS, NEWSLETTERS

LEARNING

An excellent, colorful, contemporary magazine for elementary and middle school teachers. The focus of the March 1975 issue is on math – teaching strategies, new methods and materials, activities for hand-held calculators, opinion, etc. Single copy $1.50.

Education Today Co., 530 University Ave., Palo Alto, CA 94301.

MINICOMPUTER NEWS

This bi-weekly tabloid newspaper reports on developments in minicomputer hardware, software, and applications. A recent issue covered a real potpouri of topics – Decnet, timesharing user group news, oceanographic research with minis, and Centronics re-entry into the electronic games field. $6.00 per year, sample issue free.

Benwill Publishing Co., 167 Corey Road, Brookline, MA 02146.

COMPUTER PROGRAM ABSTRACTS

The National Computer Program Abstract Service is a clearinghouse for computer program abstracts – simulation models, application programs, MIS, etc. – from business, government, military, and universities. As of Spring 1975, 17,000 abstracts were indexed in 167 specific subjects. A quarterly Program Index Newsletter is published for $10.00 per year. Sample copy free.

NCPAS, P.O. Box 3783, Washington, DC 20007.

GAMES & PUZZLES

The only monthly magazine devoted to nothing but games and puzzles. Word games, crosswords (watch out for British spelling), number games, Go, Backgammon, Chess, Scrabble, various card games, mazes, reader puzzles and games, competitions. Lively and fun. $10.80 per year to U. S., sample copy $1.10.

Games & Puzzles, 11 Tottenham Court Road, London WIA 4xF, England.

THE FUTURIST

A bi-monthly journal of forecasts, trends, and ideas about the future. The magazine presents an objective, reasoned approach to future study and does not advocate particular ideologies. Recent issues have examined teaching and education in the future, behavior control, work and leisure, and world planning. $12.00 per year, sample copy $2.00.

World Future Society, P.O. Box 30369, Washington, DC 20014.

TEACHER MAGAZINE

Teacher, one of the two big "establishment" elementary school magazines(*Instructor* is the other) often has games and activities useful for introducing young kids to concepts in math, statistics, and computers. "Clothespin Probability," "What Comes Next?" (Jan 1975) and "A Game for all Reasons" (April 1975) are three such games. Watch the column, "Creative Classroom" for the best ideas. Subscription $10/year; sample copy $1.25.

Teacher, P.O. Box 800, Cos Cob, CT 06807.

SIMULATION/GAMING/NEWS

S/G/N is now in a 32-page 8½ x 11 newsprint format which seems to have improved its very professional and comprehensive coverage of the world of simulations and serious games. It frequently carries a page or two of computer gaming news, has excellent game reviews and an annotated bibliography which appear every issue, as well as news, views, and articles about gaming and education. Bi-monthly, $6.00/year; sample copy $1.00.

Simulation/Gaming/News, Box 3039, University Station, Moscow, ID 83843.

WOMEN IN ENGINEERING

IEEE Transactions on Education, Vol. E-18, No. 1 (Feb 1975) is a special issue on women in engineering. The 14 papers in the issue discuss the growth of women in engineering programs, opportunities in various types of engineering, minority programs, and starting early with orientation in junior high school. Some of the papers are real eye openers to our changing world. Single copy $2.50.

Thelma Estrin (IEEE Committee on Professional Opportunities for Women), Brain Research Institute, Univ. of California, Los Angeles, CA 90024.

HP65 CALCULATOR NOTES

Would you believe BAGELS, CRAPS, FOOTBALL, GUNNER, HEX-A-PAWN, and PING-PONG for a pocket calculator? And much, much more? I didn't until I saw *65 NOTES*, an absolutely fascinating publication of the HP-65 (and 55) Calculator Users Club. Packed with useful information to HP-65 owners – routines, beginners corner, how to handle alphabetics(!), tips, and games. The March 1975 issue is a double issue (45 pages) entirely devoted to games. Monthly, $10 per year, $1.00 sample copy, $2.00 March games issue.

Richard J. Nelson, HP-65 Users Club, 2541 W. Camden Place, Santa Ana, CA 92704.

LEARNING AIDS

SMARTY CAT

A slide rule for young children to help in the four basic arithmetic operations. A blue cat's face moves along a red, white, and blue rule – addition and subtraction is on one side, multiplication and division on the other. Probably most useful for slow learners. $1.50 each.

Involv-o-Products, P.O. Drawer 3966, San Angelo, TX 76901.

GAMES AND SIMULATIONS

SHANTI – THE GAME OF EVERLASTING PEACE

Shanti is a game for two to four players. It is designed to illustrate principles of cooperation, harmony, and serving others. The objective is to move from student to Master to Bodhisattra and then to move freely on the board helping others. The "board" is actually a hand silk-screened muslin cloth and the markers are beautifully hand carved and painted. It is obvious that hours of loving labor (cottage industry kind) go into every set. $12.95.

Kanthaka Press, Box 696, Brookline Village, MA 02147.

DESIGN YOUR OWN GAME

This booklet by Harvard McLean and Michael Raymond contains a wealth of pragmatic instruction in how to construct simulations and instructional games. The process is described step by step with examples to illustrate most points. $1.75.

The Simulation and Gaming Association, RR No. 2, Greentree Rd., Lebanon, OH 45036.

THE GUIDE TO SIMULATIONS/GAMES FOR EDUCATION & TRAINING

By David Zucherman and Robert Horn, this is the most comprehensive guide to educational games and simulations. The second edition contains over 600 entries grouped into 20 categories. Each entry contains detailed information on playing (age level, number of players, playing and preparation time), materials, the objectives, nature, and role of the game. Also price and source information. Several articles on the use of simulations are also included. $15.00.

Information Resources, Inc., P.O. Box 493, Lexington, MA 02173.


Earl Newman

A new learning activity from Creative Computing . . . .

Learning, Innovation, and Animals

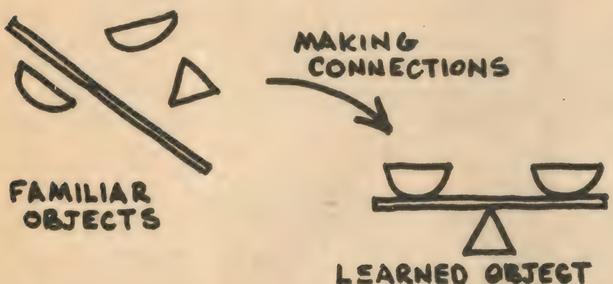
by David Ahl

This activity is unusual in several ways. First, the computer program ANIMAL is used for data storage and retrieval rather than for computations. Second, the latter part of the activity (Exercises 5-11) has nothing to do with computers per se. Its entire objective is to stretch minds in imaginative new directions. I believe the whole future of the computer rests on imaginative and creative new applications and that it's too easy to get caught up in the traditional, and somewhat mundane data processing rut. So here's an opportunity to let your mind wander and have fun in some strange and wonderful new places.

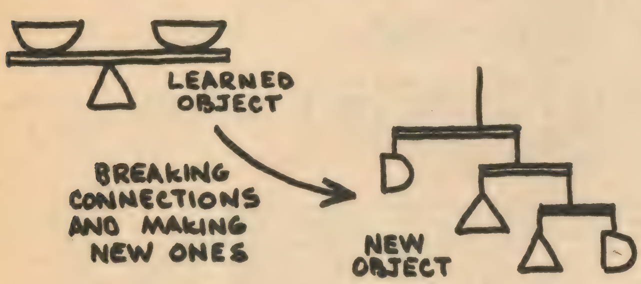
Illustrations for this article are all by Earl Newman, Summit Star Rt, Box 51, Blodgett, OR 97326.

* * *

The processes of learning and innovation, seemingly different, are really closely related.



In the *learning* process, we take familiar objects, facts, feelings, attitudes, and values and make connections to learn about new things. These connections between familiar things can take many forms. Looking at the same thing from a different point of view is one way of making a connection. Another way of making connections is by comparison. We'll look into this process at greater length in the next section, "Identifying Animals by Comparison".



In the process of *innovation*, we take familiar things, assembled or in pieces, and break the familiar connections. Then we try to make new connections to make something new, different, or better. What processes are used in innovation? Generally the same ones used in learning! Comparison, for example, is a good innovative technique as we'll see in "Strange and Familiar".

Identifying Animals by Comparison

When a young child looks at an ABC primer there isn't much to distinguish a dog from a horse. Then one day he discovers that a horse is BIGGER than a dog. Wow! Now there is one way to tell the two apart.

This is an example of the all important process of identification by comparison. Comparison involves finding a common descriptive facet about the things to be compared and then determining whether the objects are similar or different on that facet. For example, let's compare our horse (a pinto) with a dog (pointer).

Characteristics	*Horse*	*Dog*	
Size	Large	Small	Different
Marking	Spots	Spots	Same
Color spots	Brown	Black	Different
Tail	Long	Long	Same
Ears	Pointed	Drooping	Different
Used by man	Hunting	Hunting	Same

So we see on the six dimensions we've looked at, the horse and dog are similar on three and different on three. As we grow older, we continue to refine this process until we can distinguish betweeen very similar things (cocker spaniel and springer spaniel, for example).

One way to learn more about this process of comparison to identify things and also to sharpen your own descriptive skills is to teach someone else to identify similar things by comparison. The computer program ANIMAL is just such a willing "someone" waiting to be taught.

In playing ANIMAL, we will teach the computer to identify various animals by asking questions that can be answered with a yes or no. When you first start with the computer, you'll find it knows very little. It asks you to think of an animal. Let's say you think of a gorilla. The computer will ask:

DOES IT FLY? NO (your reply)
IS IT A FISH? NO

So you see the computer knows only a BIRD (no specific kinds) and a FISH (again, no varieties). After you respond NO to the question, "Is it a fish?" the computer says:

THE ANIMAL YOU WERE THINKING OF WAS A? GORILLA

And now we come to the crux of the comparison process as the computer says:

PLEASE TYPE IN A QUESTION THAT WOULD DISTINGUISH A GORILLA FROM A FISH
?DOES IT HAVE FUR
FOR A GORILLA THE ANSWER WOULD BE? YES

The next time through the program, if you said your animal was not a fish, the computer would ask, "Does it have fur?" Gradually through this process the computer builds up its repertoire of animals.

Notice that where the computer asked for a question to distinguish a gorilla from a fish, we could have said:

? DOES IT HAVE FINS
FOR A GORILLA THE ANSWER WOULD BE? NO

In other words, animals can be distinguished with either yes or no questions.

EXERCISE 1

For each of the following pairs of animals, write *two* questions that will distinguish between them. Write one question so that it can be answered "yes" for the first animal in the pair; the other, "no"

DOG	TIGER	MOOSE
HORSE	PUMA	RAM
ELEPHANT	CAMEL	OCELOT
HIPPOPOTAMUS	LLAMA	CHEETAH

EXERCISE 2

There are many possible ways to distinguish between two things. For each of the following pairs of animals, write *seven* questions that will distinguish between them.

OSTRICH	PENGUIN
GIRAFFE	GORILLA

EXERCISE 3

Choose one or two "families" of animals. Go to an encyclopedia, wild life book, or other source and find out all the members of the family and their distinguishing characteristics. To start you off, here are the names of some of the members of the cat family:

LION	OCELOT
TIGER	CHEETAH
PUMA	PANTHER
LEOPARD	JAGUAR
OUNCE	CAT, SIAMESE
COUGAR	CAT, PERSIAN
LYNX	

EXERCISE 4

Play ANIMAL on the computer. Teach it your family of animals from Exercise 2. If members of the class have chosen different families of animals, when you are finished, the computer should be able to identify just about any animal.

SAMPLE RUN OF ANIMAL

```
THINK OF AN ANIMAL AND THE COMPUTER WILL TRY TO GUESS IT...

ARE YOU THINKING OF AN ANIMAL? YES
DOES IT SWIM? YES
IS IT A FISH? NO
THE ANIMAL YOU WERE THINKING OF WAS A ? SEAL
PLEASE TYPE IN A QUESTION THAT WOULD DISTINGUISH A SEAL FROM A FISH
? DOES IT HAVE FLIPPERS
FOR A SEAL THE ANSWER WOULD BE? YES
ARE YOU THINKING OF AN ANIMAL? YES
DOES IT SWIM? NO
IS IT A BIRD? NO
THE ANIMAL YOU WERE THINKING OF WAS A ? ELEPHANT
PLEASE TYPE IN A QUESTION THAT WOULD DISTINGUISH A ELEPHANT FROM A BIRD
? DOES IT HAVE A TRUNK
FOR A ELEPHANT THE ANSWER WOULD BE? YES
ARE YOU THINKING OF AN ANIMAL? YES
DOES IT SWIM? NO
DOES IT HAVE A TRUNK? NO
DOES IT GO 'ARF? NO
DOES IT HAVE RETRACTIBLE CLAWS? YES
IS IT A CAT? NO
THE ANIMAL YOU WERE THINKING OF WAS A ? TIGER
PLEASE TYPE IN A QUESTION THAT WOULD DISTINGUISH A TIGER FROM A CAT
? DOES IT LIVE IN THE JUNGLE
FOR A TIGER THE ANSWER WOULD BE? YES
ARE YOU THINKING OF AN ANIMAL? YES
DOES IT SWIM? NO
DOES IT HAVE A TRUNK? NO
DOES IT GO 'ARF? NO
DOES IT HAVE RETRACTIBLE CLAWS? NO
IS IT A BIRD? NO
THE ANIMAL YOU WERE THINKING OF WAS A ? COW
PLEASE TYPE IN A QUESTION THAT WOULD DISTINGUISH A COW FROM A BIRD
? DOES IT GIVE MILK AND GO 'MOO'
FOR A COW THE ANSWER WOULD BE? YES
ARE YOU THINKING OF AN ANIMAL? YES
DOES IT SWIM? NO
DOES IT HAVE A TRUNK? NO
DOES IT GO 'ARF? NO
DOES IT HAVE RETRACTIBLE CLAWS? NO
DOES IT GIVE MILK AND GO 'MOO'? NO
IS IT A BIRD? NO
THE ANIMAL YOU WERE THINKING OF WAS A ? GOAT
PLEASE TYPE IN A QUESTION THAT WOULD DISTINGUISH A GOAT FROM A BIRD
? DOES IT GIVE MILK AND EAT TIN CANS AND ALMOST ANYTHING ELSE
FOR A GOAT THE ANSWER WOULD BE? YES
ARE YOU THINKING OF AN ANIMAL? YES
DOES IT SWIM? YES
DOES IT HAVE FLIPPERS? NO
IS IT A FISH? NO
THE ANIMAL YOU WERE THINKING OF WAS A ? WHALE
PLEASE TYPE IN A QUESTION THAT WOULD DISTINGUISH A WHALE FROM A FISH
? IS IT THE LARGEST KNOWN MAMMAL
FOR A WHALE THE ANSWER WOULD BE? YES
ARE YOU THINKING OF AN ANIMAL? LIST
ANIMALS I ALREADY KNOW ARE
SEAL           ELEPHANT       DOG            CAT            TIGER
COW            BIRD           GOAT           FISH           WHALE
```

Versions of ANIMAL exist in BASIC and various assembly languages. The most comprehensive BASIC version (by Art Luehrmann) is available for $2.00 for a paper tape and 50¢ for the listing from Program Librarian, Kiewit Computation Center, Dartmouth College, Hanover, NH 03755.

Interlude

Here are some facts about animals that may help you with the exercises in this chapter.

* * *

The keenest sense of smell exhibited in all nature is that of the male silkworm moth. It can detect the sex signals of a female which is 6.8 miles away!

* * *

The Gaboon viper has the longest fangs of any snake. The specimen kept in the Philadelphia Zoo in 1963 was a little careless and bit itself to death.

* * *

Speed is so essential to the survival of the gazelle that nature has endowed it with the ability to run almost from the moment it is born. While most animals are weak and wobbly in the first few days after birth, a two-day-old gazelle can outrun a full-grown horse.

* * *

Gazelles, prairie dogs, wild asses, and many other animals never drink water. They have a special chemical process which transforms a part of their solid food into water.

* * *

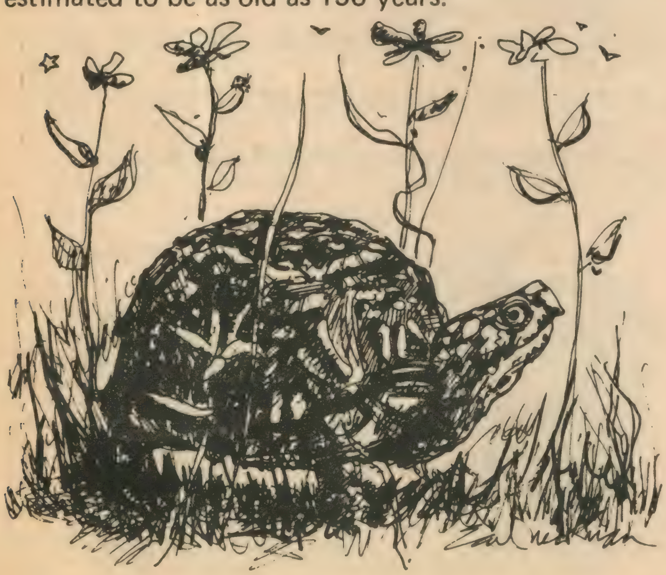
The world's fastest animal is the cheetah. It has been timed at 70 miles per hour, but many believe that it can do even better over a short haul. Sometimes called the hunting leopard, the cheetah has long been used in India to track down the black buck, the Indian antelope, and other fast game.

* * *

The animal that takes the longest time to make its debut is the elephant. Its gestation period is 645 days or more than 21 months.

* * *

The most long-lived animal is the giant tortoise of the Galapagos Islands. Specimens have been estimated to be as old as 190 years.



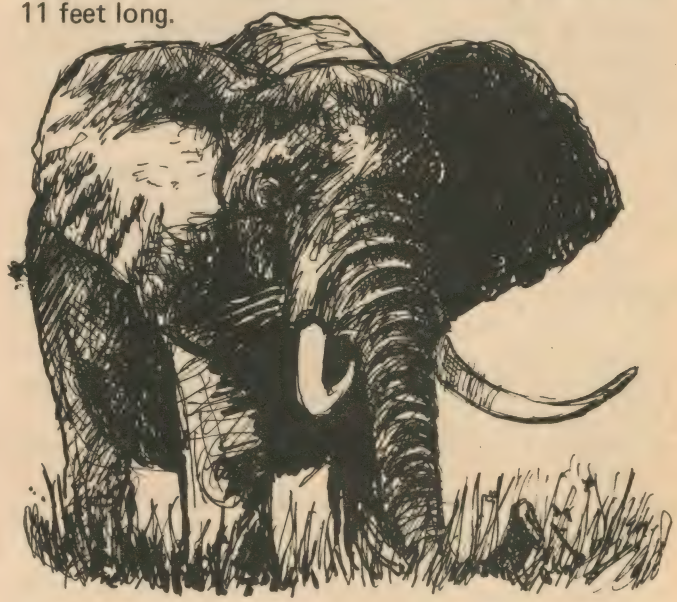
The blue whale is the largest and most powerful animal ever to have graced the planet. The largest accurately measured specimen was captured off Scotland in 1926; it measured 109 feet 4¼ inches in length. A whale caught off Argentina five years later is said to have weighed 195 tons.

* * *

The longest of all worms is the *Lineus longissimus*, or "living fishing line worm." In 1964, a specimen washed ashore at St. Andrews, Scotland, after a storm. It measured more than 180 feet in length.

* * *

The tusks of some male African elephants eventually become so heavy that their owners must frequently rest them in the forks of trees. The longest African elephant tusk on record was some 11 feet long.



The wildcat is the most vicious fighter in the animal kingdom. Asleep, it resembles a gentle housecat — in a fight, it is a furry ball of rage. This spitfire's speed gives it an advantage over most other animals. In one swift leap, it can rip open its enemy's throat with its razor-like teeth.

* * *

The world's largest rodent is the capybara, also called the carpincho or water hog. A native of tropical South America, it can attain a length of 3½ to 4½ feet and a weight of 150 pounds.

* * *

When in mortal danger, many animals feign death. But none do this as convincingly as the American opposum and the dingo, a wild dog of Australia. The dingo will allow its captor to beat it unmercifully until the chance to escape presents itself. The entrapped opossum will assume its famous "possum" pose, which is to lie limp with its tongue hanging out of its mouth and its eyes open and rolled back.

* * *

Although the whale weighs over a hundred tons and the mouse tips the scales at only a few ounces, they develop from eggs of approximately the same size.

Strange and Familiar

We've seen from the animal exercises how we can make finer and finer comparisons until we can distinguish almost any kind of animal. But now, suppose that instead of comparing similar things (animals), we make comparisons between things that are outwardly different, between things that might conflict with each other. In doing the following exercises, let your mind stretch a bit. Try to be aware of not only the way you see things, but how they might be seen by an insect or a rock or an old man or someone on another planet.

EXERCISE 5

For each of the questions below, choose the answer you think is best. Then write a sentence or two to explain your choice. Here is a sample question with two completely different sample answers. Neither answer is right or wrong. What is important is the reason *you* have for *your* choice.

Q. What needs more protection? TURTLE or ROCK

A. Turtle — because rocks are not even alive.

A. Rock — because a turtle can regrow a break in its shell. If a rock cracks it can't mend itself.

The first time you read the question, the obvious response (Turtle) will probably spring to your mind. However, once you go beyond the obvious, you might find other interesting and innovative possibilities (Rock). Remember — there is *no* one *right* answer.

1. Which is stronger? SCISSORS or FACIAL TISSUE
2. Which is dirtier? DIRT or DETERGENT
3. Which is more powerful? A BULLDOZER or YOUR CONSCIENCE
4. Which is lighter? HELIUM BALLOON or JOY
5. What color is love? Why?

EXERCISE 6

Here are five more questions. The instructions are the same as in Exercise 5.

1. Which is more talkative? A STREAM or THE WIND
2. Which is more alive? AN OSTRICH EGG or A RICE PLANT
3. Which is brighter? A FLAME or A CHILD'S LAUGH
4. Which lasts longer? A PAIR OF JEANS or A DROP OF RAIN
5. Which sees more? YOUR FINGERS or YOUR EARS

EXERCISE 7

In this exercise we will be looking for the connectives between two dissimilar things. The connections may or not be obvious. Write a sentence or two answer to each of the following questions. Remember — there is no right answer.

1. How is AN ICEBERG like a BIG IDEA?
2. How is s SUNSET like a MIRROR?
3. How is a TREE like SELF RESPECT?
4. How is the BEACH like TIME?
5. How are INTEGRITY and the OCEAN alike?

EXERCISE 8

For this exercise you should have a partner. For each question, you should come up with a one or two word answer. Then trade papers with your partner who must explain *your* answer. For example:

QUESTION: What ANIMAL behaves like a DELIVERY TRUCK?

YOUR ANSWER: A Kangaroo.

(Trade papers with your partner)

PARTNER EXPLANATION: A kangaroo is like a delivery truck because it carries its young around in a pouch.

Stretch your comparisons. Try to stump your partner. However, your partner can challenge you. Therefore, don't go so far out that you can't make the connection yourself. If you can't explain your connection, you are stumped!

1. A CANDLE is like what ANIMAL?
2. What aspect of COMPUTERS reminds you of TARROT cards?
3. What part of the HUMAN BODY is like an APPLE?
4. What MECHANICAL THING is like an IGUANA?
5. What ANIMAL behaves like SATURDAY?

EXERCISE 9

Get a new partner and do these five questions the same as Exercise 8.

1. What PLANT could best teach a child SELF CONFIDENCE?
2. The idea of COMFORT may have come from what ANIMAL?
3. A MAGNET behaves like what part of the HUMAN BODY?
4. The phrase EXACT DISORDER could describe what?
5. What might be described by the words, STRAIGHT TWIST?

EXERCISE 10

Now it's your turn to help compose the questions. Below are five questions similar to those in Exercises 5 and 6. It's up to you to provide a choice of two possible answers. Then exchange your paper with a new partner and answer the questions the same way as you did in Exercises 5 and 6. For example:

QUESTION: Which is larger?

YOUR ANSWER: A HIPPOPOTAMUS or A MOTHER'S LOVE

(Trade papers with your partner who then chooses an answer and explains it in one or two lines.)

1. Which teaches self-control?
2. Which is most hungry?
3. Which is more beautiful?
4. Which is like a New England church?
5. Which costs more?

EXERCISE 11

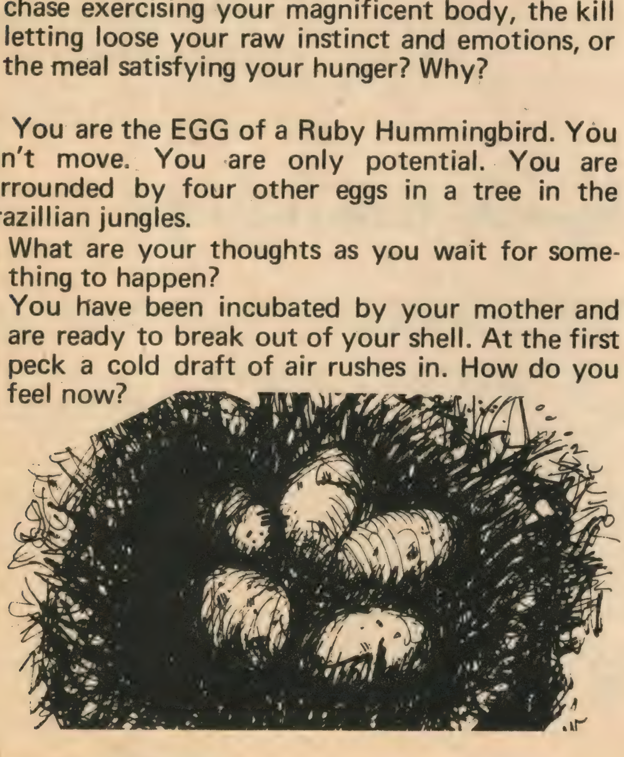
In this exercise, you'll have to stretch your imagination a bit because you're going to look at the world from an entirely new viewpoint of some other thing. You must try to feel the way that thing does. FEEL the thing. ACT the thing. BE the thing while you write a one or two line answer to each question. (Remember – there is no right answer.)

A. You are a CHEETAH. You can run at 70 mph for short hundred yard bursts. You are tawny with small deep brown spots. You are 5 feet long and weigh 110 pounds.

1. You are hungry. You see an antelope at a water hole about 200 yards away. Describe your feelings of anticipation.
2. You have chased, killed, and eaten the antelope. Which gave you the most satisfaction – the chase exercising your magnificent body, the kill letting loose your raw instinct and emotions, or the meal satisfying your hunger? Why?

B. You are the EGG of a Ruby Hummingbird. You can't move. You are only potential. You are surrounded by four other eggs in a tree in the Brazillian jungles.

1. What are your thoughts as you wait for something to happen?
2. You have been incubated by your mother and are ready to break out of your shell. At the first peck a cold draft of air rushes in. How do you feel now?



C. You are an ACORN. You have just fallen 60 feet from an oak tree and have rolled next to a giant boulder. The soil is soft and fertile and after the winter snows you find you have settled about a half inch into the soil, just enough for you to crack open and send out roots.

1. You are a tiny acorn. Do you have any feeling for what you'll look like when you grow up? How do you know?
2. You have begun to sprout. How do you feel about the enormous boulder practically on top of you?
3. No rain has fallen recently. How do your roots feel digging for nourishment without water? One root bumps into a sewer pipe; how does it feel?

D. You are a HEADLIGHT FILAMENT. You are very fine and made out of carbon and tungsten. You are a part of a car which has been driven almost 80,000 miles. You have been turned on and off over 3,000 times. You know the driver of the car relies on you to see at night and you have never let him down.

1. It is night. The tingle of the first surge of current comes through you as it has so many times in the past. How do you feel?
2. The owner of the car is arranging to have the car towed to an auto scrap yard. The motor and transmission are worn out but you still work fine. How do you feel toward your owner? Toward the other parts of the car? Toward the owner of the scrap yard? Toward the other old headlights around you?

E. It is the first day of the deer hunting season. A buck has been shot in the fleshy part of his buttocks but has not fallen. Hunters surround him on two sides. You are the FEAR inside the deer. You are not the deer but the raw, naked, panicky fear inside his brain and nerves.

1. You are the fear that is driving the deer to spring away from the hunters despite his pain. How do you, FEAR, feel as the deer becomes weaker and weaker from the wound and the exertion of running?
2. What are the good and bad things you feel as FEAR?

EXERCISE 12

Make up one "be another thing" exercise like A through E in Exercise 11. Pick something that will cause the person doing the exercise to really stretch his or her imagination. Exchange your paper with a partner and do each other's exercise. When you have finished discuss your "answers" with each other.

EXERCISE 13

Of all the exercises done by you and your classmates in this chapter, which one caused you to stretch your imagination the most? Why? Did you stretch more when dealing with animals, plants, objects, attitudes, feelings, or values?

A new learning activity from Creative Computing . . . .

You Don't Need a Computer for These

by David Ahl

No computer? Then try these activities in class or at home.

Build a "computer". Using junk and leftovers (boxes, cans, fabric, yarn, spools, bits of plastic, etc.) build your own computer. (Send a photo of your creation to *Creative Computing* and we'll print the best ones).

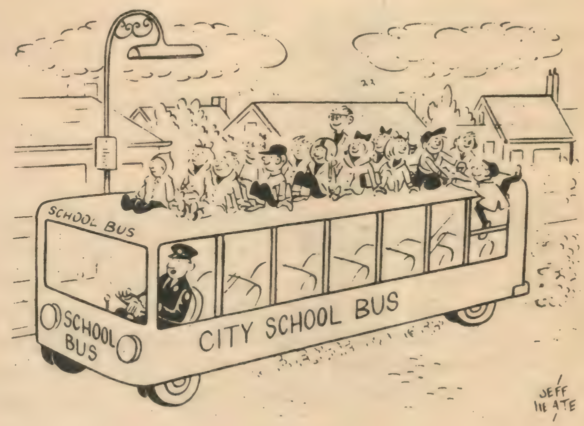
Clip a story. Look for stories or articles mentioning computers in newspapers, magazines, and non-computer publications. What was the role of the computer? Was the reporting, in your opinion, accurate?

Draw a computer. What does a computer look like in a medical laboratory, a school, a factory, on board a spacecraft? Draw your view.

Write a story. Will computers take over the world, develop individual personalities, become as small as a wrist watch, etc? What will it do in the home? For grocery shoppers? For beggars? Let your speculations wander in a story of your own. (Send your stories to *CC.*)

Produce a glossary of computer terms. Many "computer" words are not in the dictionary. How can you find such words and their meanings. You ought to be able to find at least 30 words peculiar to the computer field (actually there are several hundred), and at least 10 words or acronyms that are not in standard dictionaries.

Visit a computer center. Try a local community college, university, bank, or company. But be sure to arrange your visit well in advance so you don't show up on payroll check day.



"Well, I'm glad to see that ONE day out of the year you kids can stop all that screaming and jumping around and behave like ladies and gentlemen"

The 10ᶜ Computer and Other Games

by Gwyn Lyon
Gates Elementary School
Acton, Massachusetts

The teaching of math to kids is traditionally divided into lecture and drill. In the re-thinking of a second grade math program, I decided to incorporate some different approaches to aspects of math computation. A game such as Input-Output can be noisily exciting, and effectively teach the relationships between sets of numbers.

INPUT-OUTPUT

A game for the Blackboard or Overhead Projector. Divide the acetate into two columns or make two columns on the blackboard. Place a numeral in the left or Input column. Through the function of the magic-black box-computing-teaching machine, the number is transformed into a new number.

	teaching machine = ?
Input	Output
4	7

Place a new number in the input column, and using the same teaching machine function, a new number appears in Output.

Input	Output
4	7
0	3
	answer: t.m. = +3

Continue to add numbers until a student guesses what function the teaching machine is set for.

Second graders can do problems of this level of difficulty:

	teaching machine = ?
Input	Output
4	11
7	17
10	23
	answer: t.m. = 2x + 3

The game can be adapted for many types of algebraic equations:

	teaching machine = ?
Input	Output
9	4
25	6
4	3
	answer: t.m. = $\sqrt{x}$ +1

As an added bonus, a child with such learning problems as would exclude him from the successful completion of a traditional worksheet can often excel in "head" games that require no written response.

THE 10¢ COMPUTER

The teacher must also begin to rely more on manipulative materials to move the child securely from the concrete to the abstract. Hence, the ten cent computer:

Materials: a large grocery box. 1 sheet of acetate, 8" x 10".

Method: Cut the box up so that you have one large sheet of cardboard, without seams, that will completely cover the stage of the overhead projector. Mine is 12" x 11½", but measure yours to be sure.

Cut four holes in the top third of your cardboard. Save the cut-outs! Tape the sheet of acetate so that it covers the holes. Then hinge the covers back over the holes with tape.

A Simple Binary Game: Beginning at the left side, mark the numerals 1, 2, 4, 8 on each acetate-covered hole.* Close all the covers. Place the computer on the stage of the projector, and turn on the projector. Lift the first cover on the left. Computer now shows "1". Write on the blackboard, 1000.

Close all the covers. Lift cover marked "2". Write on blackboard, 0100, and explain that because the light is on in the "2" position, the binary notation is 0100.

Close all covers. Explain that all numbers up to 15 can be shown with only the four lighted positions. How can you make 3? 7? 14?

Binary Numers for the 10¢ computer

●●●●	0	●●●0	8
0●●●	1	0●●0	9
●0●●	2	●0●0	10
00●●	3	00●0	11
●●0●	4	●●00	12
0●0●	5	0●00	13
●00●	6	●000	14
000●	7	0000	15

By this time, the children will have grasped the idea, and can work quite competently up to 32, using only the blackboard notations. If you want to carry on, you can make an "advanced" computer with 6 holes that can record numbers all the way up to 63!

*Normal binary number notation goes right-to-left, i.e. 1 = 0001, 6 = 0110, 8 = 1000, etc. If you think your kids can grasp this, give it a try.

Suggested Bibliography

Ahl, David. *Getting Started in Classroom Computing.* Digital Equipment Corporation, Maynard, Massachusetts.

Adler, Irving and Ruth. *Numerals, New Dresses for Old Numbers.* John Day, Company. New York, N. Y.

Kenyon, Raymond. *I Can Learn about Calculators and Computers.* Harper and Row. New York, N. Y.

A new learning activity from Creative Computing . . . .

Beginning With BASIC

by William R. Morrison
Curtis Jr. High School
Sudbury, Massachusetts

The act of sitting down at a computer terminal creates a closed world. Even first-time users become absorbed, lose track of time, and seek only to further whatever dialogue they are involved in. Can you remember the pleasures of getting the machine to do what you wanted it to? Or the frustrations of many error messages? These two activities are designed for junior-high school students to use as part of an introductory unit on computers. They are intended to help make that first experience informative; to let students learn step by step how the computer responds to various commands; and finally to let students form their own images of the computer as a partner in a dialogue.

A. EXERCISES IN IMMEDIATE MODE

This first guide takes the student from immediate mode to a simple program. It presents a minimum of rules, emphasizing instead a series of actions, and lets the student draw whatever conclusions he can. The impact of these first impressions will later be the foundation for a more formalized presentation, the objects of which should be familiar. Explanations tend to turn students off unless they have already had an experience that makes them eager to know why.

Some Things to Try With a Computer that Speaks BASIC:

1. Try some PRINT statements:

   PRINT MY COMPUTER SPEAKS BASIC

   What happened? Did you forget something? Correct the command and try again. Will the computer accept something like: 2+5=4? Can you make it print such arithmetic nonsense? Now print your name and the date.
2. Try this: PRINT "SOME ARITHMETIC"; 22*33;35/70;5–9,7+9 What happens when you use the ; or the ,? How does the computer handle fractions? Try a few, such as 1/3+5/8, and see.
3. Here is a short program. Type it in and try it out.

```
10 INPUT A
20 INPUT B
30 PRINT A "TIMES" B "IS" A*B
40 GOTO 10
```

   If something went wrong, remember that everything you type in has meaning to the computer, every quote mark, comma, etc.
4. Let's make some changes and see how this program can be improved. You don't have to type the whole thing over again, of course; change the lines by typing the line number and its line. First, though, stop the computer from running the old program by typing CTRL/C.

   Change line 20: Type: 20 LET B=1
   Change line 40: Type: 40 LET B=B+1
   Add a line 50: Type: 50 GOTO 30

   What happened to old line 20? Find out by asking the computer to LIST the program. Now, try a RUN of the program. Stop the program with CTRL/C – does CTRL/C stop the computer while it is LISTing a program?

   Give the program a title with:

   5 PRINT "TIMES TABLE"

   Do you see why line numbers go up by 10's? Now make the program stop automatically when it gets to A times 10:

   45 IF B=10 THEN 60

   Of course you need line 60!

   60 STOP

   If B is *not* 10, what does the computer do?
5. You should now have a program that asks for a number and then types out all the multiples of that number up to 10 times that number. Go ahead – pick a number and hand in a RUN and a LIST of the program.
6. Can you:
   a) Further change the program to add, subtract or divide instead of multiplying?
   b) Change the program to print the multiplications or divisions, etc., in a table form?
   c) (Difficult!) Make the computer print the table starting at 1 x 1, going to 1 x 10, then starting over by itself at 2 x 1, 2 x 2, etc.?

B. EXTENDING A SIMPLE PROGRAM TO SOLVE COMPLEX PROBLEMS

This hands-on guide is for somewhat more proficient students, and is designed to illustrate some ways of adapting a simple program to more complex uses. It presents the program as a resource, a given tool to be used for solving a problem. As an added benefit, students are helped to review and extend their knowledge of some important mathematical concepts.

A Problem to Solve and a Program to Help:

Problem:

Find out if 1517 is a prime number. If it is not prime, find at least two divisors.

Solution:

Instead of looking for a table of prime numbers (unfair!) or doing a whole mess of division problems to look for divisors, and maybe not finding any, use this simple program:

```
10 LET A=1517
20 INPUT B
30 PRINT A/B
40 GOTO 20
```

This program tells the computer to:

- Remember the number 1517 (Line 10)
- Ask for a number — it will type ? and wait (Line 20)
- Divide 1517 by that number (Line 30)
- And go back for another divisor (Line 40)

You still have to give the computer numbers to try as divisors, but at least *it*, not you, will be doing the work.

Question:

Remember the Sieve of Eratosthenes? Even if you don't, do you have to divide 1517 by all the multiples of 2, or just by 2? Now, since there are no other even primes, why not divide 1517 by only odd numbers after you try 2? Why not leave out multiples of 3, 5, 7 etc.?

Information:

Here are some prime numbers to try as divisors: 2 3 5 7 11 13 17 19 23 29 31 37 41 43 . . .

Problems:

Change the above program to print:

a) Your name
b) Each division problem as well as its answer
c) Using lines such as LET B=3 and LET B=B+2, all the divisions of 1517 by odd numbers until you stop the computer with a CTRL/C

The above exercises are merely suggestions for making the computer's self-sufficiency pay off and helping the student maintain his own private dialogue with the machine.

A Puzzle For Fun

A warden had eight prisoners in separate cells, arranged in the manner shown below.

He decided to rearrange the prisoners so their numbers would be in consecutive order, reading counter-clockwise around the circle, with the center cell left empty.

The warden started by moving a prisoner into the empty cell, then moving the other prisoners one at a time, always into the cell vacated by the previous occupant. All but No. 5, who was never moved from his cell.

Prisoners can only be moved to a cell adjacent to their own, i.e., only 7 or 3 could move to the empty cell.

Can you figure out how the warden did it in 17 moves?

Odd or Even?

by Jeffrey Moskow (Student)
Lexington High School, Mass.

My first computer programming instructor defined a computer as an expensive, fancy, overgrown adding machine. Bearing this in mind it is not difficult to see that a computer is primarily concerned with numbers. One of the most important things to remember when programming a computer is that whatever you are trying to accomplish should be reduced to numerical operations before writing the program. The differences between numbers can be used to write many programs, both simple and complex. The following program is only concerned with two kinds of integers, odd and even.

```
 10 PRINT "THIS PROGRAM TAKES ANY INTEGER AND TELLS"
 20 PRINT "IF IT IS ODD OR EVEN"
 30 PRINT "PLEASE GIVE ME A NUMBER";
 40 INPUT X
 50 IF X/2=INT(X/2) THEN 80
 60 PRINT X;" IS AN ODD NUMBER"
 70 GOTO 30
 80 PRINT X;" IS AN EVEN NUMBER"
 90 GOTO 30
100 END

RUN

THIS PROGRAM TAKES ANY INTEGER AND TELLS
IF IT IS ODD OR EVEN
PLEASE GIVE ME A NUMBER? 34
 34  IS AN EVEN NUMBER
PLEASE GIVE ME A NUMBER? 57
 57  IS AN ODD NUMBER
PLEASE GIVE ME A NUMBER? 133
 133  IS AN ODD NUMBER
PLEASE GIVE ME A NUMBER?
↑C
```

This simple program is based on the fact that even numbers are divisible by 2 while odd numbers are not. This same principle can be used to simulate the tossing of a coin. The only difference is that this program uses the random number function.

```
10 RANDOMIZE
20 PRINT "THIS PROGRAM SIMULATES THE TOSSING OF A COIN"
30 PRINT "HOW MANY TOSSES";
40 INPUT X
50 FOR N=1 TO X
60 LET F=INT(2*RND(0))+1
70 IF F/2=INT(F/2) THEN 100
80 H=H+1
90 GOTO 110
100 T=T+1
110 NEXT N
120 PRINT "IT WAS HEADS ";H;" TIMES"
130 PRINT "IT WAS TAILS";T;" TIMES"
140 END

THIS PROGRAM SIMULATES THE TOSSING OF A COIN
HOW MANY TOSSES? 15
IT WAS HEADS  9  TIMES
IT WAS TAILS 6  TIMES
```

Will the results for 15 tosses always be the same?

Will the numbers of heads and tails be closer if the number of tosses is greater? Why?

Programming:

Try to write a program that shoots dice.

Try to write a program that allows the user to input a number and the computer will output whether or not the number is an integer.

Cryptic Puzzle

by Denis Kaminski
George Washington Jr. HS
Ridgewood, NJ

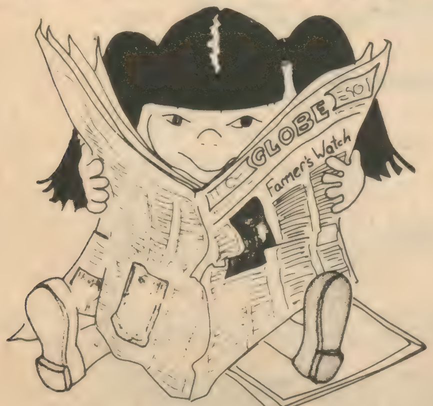
Below are 11 computer words put into a code. Each letter of the alphabet was replaced with a different letter. Remember that if "S" stands for "Q" in one word, it will be the same throughout the list. A clue is given with the first word.

P Q X U O (A popular beginner's language)

M D F G F Q C	G Z B Z G S W Z
B D D W	O D F Z
P H M M Z F	U P Y
D H G W H G	Y Z Y D F S
U C W H G	M B D J O R Q F G X

Odd or Even?

Is the number of letters used in a headline of a newspaper odd or even?



Find out for 20 different headlines.

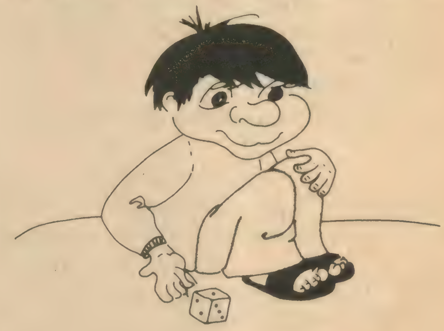
Odd or Even?

Choose a shelf full of books.
Is there an odd or even number of books on that shelf?

Find out for 20 different book shelves.

Odd or Even?

Roll one of the dice.
Is the number of spots odd or even?

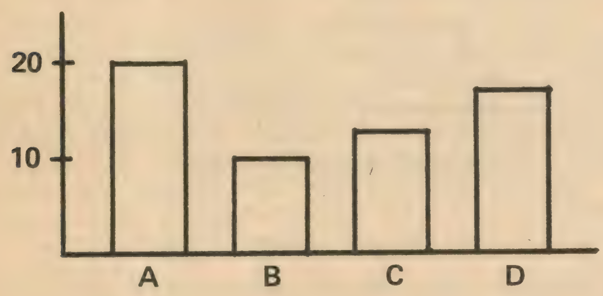


Find out for 20 different rolls of the dice.

Alphabet Statistics

Using short paragraphs taken out of textbooks, magazines, or the newspaper, have each student count up the number of A's, B's, C's, and D's and list these in order of their frequency. When everyone is done, compare results. Chances are in most written material, A will be the most frequent letter followed by D, C, and B.

Have students plot each of their results on a bar graph.



This is a good exercise to introduce the idea of scaling. For example, a long paragraph may have 100 or more A's, whereas a short one might have fewer than 20. However, for a meaningful plot, one division on the graph paper (¼″) might equal ten letters when 100 letters are to be plotted or only two letters when 20 are the maximum in any bar.

A simple computer program can be written to accept as input the results from each student and then compare his percentages with those of the entire class.

PROGRAM LISTING

```
LISTNH
10 PRINT "STATISTICAL LETTER ANALYSIS" \ PRINT
20 PRINT "ENTER 'END' AFTER LAST STUDENT" \ PRINT
30 DIM A(100), B(100), C(100), D(100), T(100), N$(50)
40 I=I+1 \ PRINT
50 INPUT "NAME"; N$(I) \ IF N$(I)="END" THEN 200
60 PRINT "HOW MANY OF EACH LETTER"
70 INPUT "A", A(I)
80 INPUT "B", B(I)
90 INPUT "C", C(I)
100 INPUT "D", D(I)
105 REM
        INDIVIDUAL TOTALS

110 T(I)=A(I)+B(I)+C(I)+D(I)
120 A1=A1+A(I) \ B1=B1+B(I) \ C1=C1+C(I) \ D1=D1+D(I)
130 GOTO 40
190 REM
        COMPUTE TOTAL AND PERCENTAGES

200 T=A1+B1+C1+D1 \ N=I-1
210 A2=A1*100/T \ B2=B1*100/T \ C2=C1*100/T \ D2=D1*100/T
215 REM
        PRINT RESULTS

220 FOR I=1 TO N
230 PRINT FOR M=1 TO 5
240 PRINT "-"; FOR M=1 TO 50
250 PRINT \ PRINT "ANALYSIS FOR "N$(I) \ PRINT
260 PRINT ,"CLASS","YOUR","CLASS","YOUR"
270 PRINT "LETTER","TOTAL","TOTAL","  %","  %"
280 PRINT "A",A1,A(I),A2,A(I)*100/T(I)
290 PRINT "B",B1,B(I),B2,B(I)*100/T(I)
300 PRINT "C",C1,C(I),C2,C(I)*100/T(I)
310 PRINT "D",D1,D(I),D2,D(I)*100/T(I)
320 NEXT I
330 PRINT FOR M=1 TO 5
340 PRINT "-";FOR M=1 TO 50
350 PRINT \ PRINT \ PRINT "THAT'S ALL FOLKS!" \END

READY
```

Programming Problems to Start With

Just getting started in programming? Whether you are learning BASIC, FORTRAN, APL, or some other language, here are eight simple problems to program and one more difficult one. The first seven of them should not take over ten statements in BASIC or FORTRAN. (We will print the best reader-submitted program for Problem #9 two issues from now).

1. Compute and print out the sum of the digits from 1 to 10.
2. Comute and print out the sum of the digits individually squared, from N1 to N2, which are inputs.
3. Input N numbers. Compute and print out the product of the even digits.
4. Generate and print out N two-digit random numbers; and also print out the largest one of these, where N is an input.
5. Generate K1 two-digit random numbers, and print out the fraction which are smaller than your age, where K1 and your age are inputs.
6. Read in K2 numbers as DATA and print them out sorted from smallest to largest.
7. Compute and print out a table of the present value of $1.00, for a rate, number of periods, and print out increment, which are all inputs.
8. Prepare a program to draw five cards at random from a 52-card deck, and print out the suit and value of each card.
9. Write a program that will print all permutations of N things taken N at a time for all N<=10.

SAMPLE RUN

```
RUNNH
STATISTICAL LETTER ANALYSIS

ENTER 'END' AFTER LAST STUDENT

NAME? DON GROSS
HOW MANY OF EACH LETTER
A? 20
B? 10
C? 13
D? 17

NAME? SALLY MELLOR
HOW MANY OF EACH LETTER
A? 49
B? 15
C? 27
D? 36
```

Each class member inputs his or her results here.

```
NAME? END
```

When all results are in, type "END". Then output starts. Tear off output for each person on line

```
--------------------------------------------------
ANALYSIS FOR DON GROSS

              CLASS          YOUR          CLASS          YOUR
LETTER        TOTAL          TOTAL           %              %
A              579            20          33.7216        33.3333
B              292            10          17.0064        16.6667
C              362            13          21.0833        21.6667
D              484            17          28.1887        28.3333
```

etc.

Probability

by David C. Johnson, University of Minnesota

PREREQUISITES
Basic notions of probability including P(E) = 1 - P(E')

DISCUSSION
The CAMP project, University of Minnesota, has conducted research and development activities on the use of the computer as a problem-solving tool in school mathematics grades 7 - 12. The following problem while a "take-off" on the classical Birthday Problem has a number of real applications relative to expected occurrence of given events:

from the everyday: What is the probability of at least two girls wearing the same style and color outfit at a party with say 30 girls invited (assuming some given number of basic styles, say g, and number of colors, c, or gxc different outfits -- e.g., if g=8 and c=10 then gxc=80.)

to a problem in manufacturing and sales: How many different styles and colors are needed to give a low probability ($p < .10$) to the event that two or more families in the same neighborhood (of 100 families) will purchase identical automobiles (if, on the average, 10% of the families purchase a new XXX each year.) Note: the problem is actually a little more complex than this, but the statement should provide a general idea -- the assumption is also made that people like their cars to be different.

PROBLEM
The situations posed above can be stated in purely mathematical terms. The three problems posed below appear in the CAMP exercises in the book *Elements of Probability* by Robert J. Wisner, Scott, Foresman and Company, 1973, appropriate for a high school course in probability.

1. First, to warm up — write a computer program to calculate the probability that at least two people in a group of n people will have the same birthday. (Hint: since the 365^n may become very large, you will have to design a procedure to calculate 365/365 x 364/365 x 363/365 x 362/365 . . .)
2. Now for *the* problem:
   a. Write a general program which considers n people selecting an alternative (or having a characteristic) from m equally likely possibilities. What is the probability that at least two will select the same alternative? You might think of this as n people each picking a number between 1 and m and writing it down -- what is the probability thatat least two will pick the same number? (Of course, $m > n$ or the probability is 1.)
   b. Use your program to determine how many numbers you will need to use at a party with 12 people to give yourself better than a 50-50 chance of having two pick the same number (you might like a probability of about .75). Do you see the similarity between this and the manufacturing problem? Actually conduct the number experiment with some groups of friends -- how well do the experimental results agree with the mathematical? Note that the experiment can be done by asking your friends to pick a favorite color or object from a list with m items -- but, you have to be cautious here; not all of the items may be equally liked by your friends -- what does this do to your computation?

Compounding

by Charles A. Reeves, Florida State University

▶ Try to fold a sheet of paper onto itself as many times as you can (i.e., fold it in half, then in half again, then again, etc.). What is the largest number of folds you can make? Someone has claimed that it is impossible to make more than 8 folds, no matter what size you start with!

But imagine for a moment that it *is* possible to fold it over onto itself a large number of times. The thickness of one sheet of notebook paper is about .004 inches. If you could fold it 50 times, how high would the stack be?

▶ Your rich uncle deposited $1000 in a savings account for you the day you were born. The account draws 6% simple interest, and the earnings are added back into the account each year. But your uncle didn't tell you about this — you found out when his will was read. He died when you were forty years old — how much did you get?

For those who want more: Same problem as above, but the interest rate is ½% each month instead of 6% per year. How much more money, if any, would you get this way?

▶ Consumer prices rose an average of 8.8% during 1973. Let's round this off to 9%, and assume that prices continue to go up this much *every* year.

Pick out an item that you think you might like to buy when you're an adult, and for which you know the present price. Write a program that will report to you how much the item will cost in the year 2000 AD.

▶ Your father gives you a penny as a gift on your first birthday. He promises to double the amount of the gift each year until you reach your 21st birthday. How much will you get from him on this birthday?

For those who want more: Have the computer print the amount you will receive on the 21st birthday, and also the *total* amount you will have gotten through the years.

▶ Erie County in upstate New York is one of the most heavily polluted areas in the United States. In a study of the residents of the county it was found that the number of people dying from respiratory diseases is *doubling* every five years. In 1950 there were 263 deaths attributed to respiratory diseases. How many deaths will there be in the year 2050 AD, assuming this same rate of increase every 5 years?

▶ The population of the world increases almost 2% each year over what it was the year before. In 1970, the world population was about 3.6 billion, or 3,600,000,000.

Have the computer calculate what the world population will be in the year 2000 AD.

▶ A salmon starts a 100 mile journey upstream to the placid lake where she was born. Each day she is able to swim 3 miles upstream, but each night when she sleeps she is pushed 2 miles back downstream. Exactly how many days will it take her to reach the quiet spawning grounds?

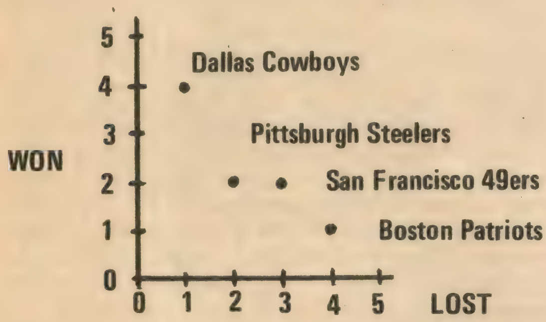
▶ The bristleworm can reproduce by splitting itself into 24 segments, each of which grows a new head and a new tail. What is the maximum number of bristleworms that could be obtained in this fashion, starting with only 1 worm, after ten "splittings"?

Sports Special

Look for math exercises in various sports that your students follow so avidly. Here are some examples of things to be computed in different sports:

Baseball	Batting averages
Basketball	Points per minute
Hockey	Penalty minutes per period
Football	Pass completion percentages

Another interesting exercise is to make a prediction model. It is easiest to do with the win-lose record for each team. For example, about halfway through the football season, the win-lose record for four teams might look like this:



Using this graph, a team that is higher and to the left has a greater chance of beating one lower and right. The further apart the teams, the greater point spread would be expected, i.e., the 49ers might be expected to beat the Patriots by one touchdown, whereas Dallas might be expected to tromp Boston by three touchdowns. Make a big chart for the class that can be updated daily or weekly and predict all the games each week. How accurate is this prediction model? Try other variables instead of win-lose record, such as third down conversions or total yardage gained.

Some students might wish to write a computer program for the prediction model. In such a program, several variables can be averaged together or even given different "weights" to make an even better model. A word of warning: No model, no matter how many variables are considered, is perfect. Indeed, if the outcome of sporting events could be perfectly forecasted, there would be no more "sport".

Here is the output from one such program:

TEAM1	EXPECTED SCORE	TEAM2	EXPECTED SCORE
BOSTON	19	BUFFALO	17
NY JETS	21	CLEVELAND	14
NY GIANTS	10	MIAMI	28
OAKLAND	21	DENVER	18
etc.			

So You Think You Know BASIC?

List the seventeen fundamental statements in the BASIC language

1______ 7______ 13______
2______ 8______ 14______
3______ 9______ 15______
4______ 10______ 16______
5______ 11______ 17______
6______ 12______

A. If there were a decree that said you had to use fewer statements, draw a line through the 5 statements you could easily do without. Are there 5 others that you could get along without?

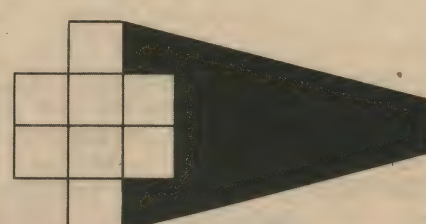
B. Circle the four that really do the most for you and which you would hold on to until the very end.

Look back over your list and your decisions and consider:

C. After getting rid of all 10 statements asked for in A., is there anything you really cannot do? If there is, you ought to think about changing some of your decisions.

D. Which statements are necessary, and which just make programming easier? What value do you place on these latter statements?

E. Consider other items in life similarly. What possessions do you enjoy (make a list of 12)? Which ones could you give up if you had to? How important to you are things that are nice but not necessary?



Insert the numbers from 1 through 8 in the eight boxes, one digit to a box, in such a manner that there are no consecutive numbers next to each other, horizontally, vertically or diagonally.

John R. Cossen
Bellerose, N.Y.

DOUBLING UP

by FRANK TAPSON

TAKE a piece of paper—you may use any size you please—and fold it in half. Then fold it in half again, and yet again, and again . . . How many times do you think you can do this? If you have never met this problem before then try it before you read any further—you will very probably receive a surprise. Have a guess first before actually trying to do the folding and then see how far you get.

Many people on meeting this little problem for the first time are prepared to say that, provided the paper is large enough then it may be folded in half any number of times. Well, as you might have discovered by now, after 7 such foldings the task becomes extremely difficult, and if not impossible then it will almost certainly be after the next fold. It is interesting to look at what in fact happens.

After our first fold, the piece of paper we have to work on next is double the thickness of the original. Another fold of this piece doubles the thickness so that we now have 4x the thickness of the original. Folding again will once more double-up on the thickness so that we have $(2 \times 2 \times 2)$ 8 thicknesses of paper. This is followed by 16 thicknesses after the next fold, then 32, then 64, and 128 after the 7th fold. Assuming that the paper we are using is one-thousandth of an inch thick (not the thinnest possible but still a flimsy paper) then after folding it 7 times we have a piece which is one-eighth of an inch thick, or about the thickness of a piece of stout card. Now such a card could certainly be folded in half generally, but there is an added difficulty. Just as the thickness has been doubled with each fold so the area has been halved, and after only 6 foldings we are usually trying to bend something which is not much bigger than an extra-large postage stamp, which is why that piece of 'stout card' is so difficult to fold.

It is interesting to wonder how far the process might be taken if a piece of super-large paper were used. Let us assume it is still one-thousandth of an inch thick, but that we can start with a piece the size of a football-pitch. Go on—have a guess, how many times would you manage to fold it in half?

Some might wish to argue about the precise stage at which the task becomes impossible, but if the 13th folding can be made, it produces something which is about four feet square and eight inches thick. Now think about bending that!

Once we start folding by speculation (and not by actually trying to do it) it becomes fascinating to go on with the process. For instance, just suppose we were able to get an extremely large piece of paper and fold in in half exactly 100 times and, having done that we wished to stand on top of it—how long a ladder would we need to get to the top? By now you have no doubt some idea of what to expect—or have you? After the 26th fold we have a "piece of paper" which is just over a mile thick so you might think we are going to need a fairly tall ladder for 100 folds. Keep going—the 53rd fold gets us just past the sun, and if you think that we are at least over half way then you have failed to see what doubling is all about. The 83rd fold gets us somewhere near the centre of our galaxy, from which it follows that the 84th fold puts us out on the other side and still going. And there we will let the matter rest, if anyone can work out 'precisely' where the top of our work will be after the 100th fold do let us know. We might be able to use it as a navigational aid for inter-stellar travel!

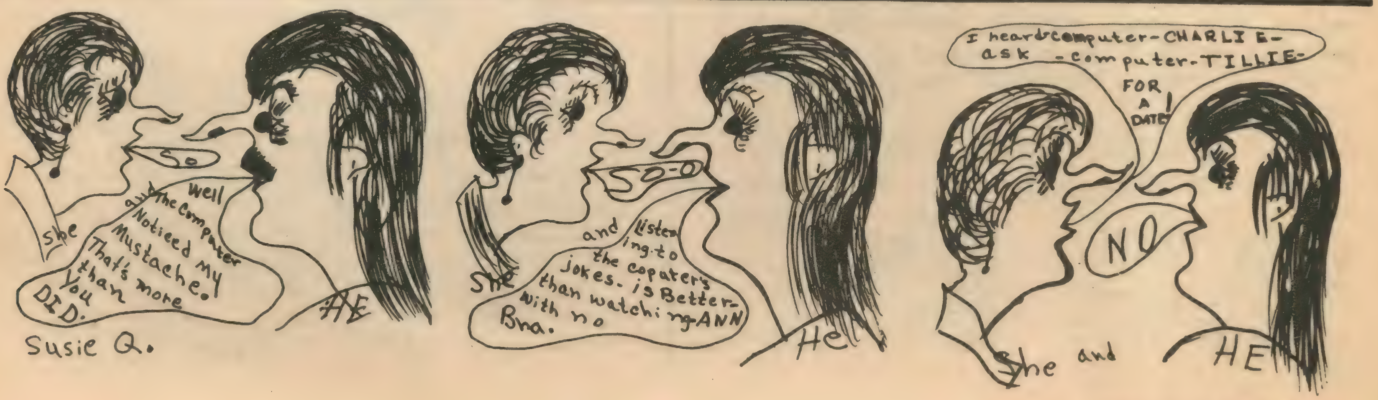
This simple concept of the growth of the doubling sequence has had a fascination for those concerned with the lighter side of mathematics for many years. Perhaps the most famous is the story told around the invention of the chess-board, how the king was so pleased that he offered the inventor any reward that the inventor cared to name. This was expressed as 'one grain of corn on the first square of the board, two grains on the second square, four grains on the third square and so on . . .' The king thought this is a very light price to pay for such a great game and readily agreed. However, he was not at all pleased to learn that the total quantity of grain required could not be supplied by the entire world output of grain for several years to come. Some accounts of the story claim that he had the inventor beheaded for imposing such a mathematical joke upon royalty! Re-telling this story in his mammoth work *A History of Chess*, H. J. R. Murray says that the quantity of grain needed is such as to cover England to a uniform depth of 38.4 feet. The actual *number* of grains needed to fulfil the stated conditions is 2^{64}-1, a figure which also occurs in connection with the story woven around the Tower of Hanoi.

Another form of the story involves either the sale of a horse, or the shoeing of one. In either case the price is fixed at a farthing (over a hundred years ago) or a penny for the first nail in its shoes, doubled-up for the second nail, doubled again for the third nail and so on. The only serious disagreement appears to be concerning the total number of nails (I have stories giving 6, 7, and 8 nails per shoe).

A story can also be woven around the telling of a secret to two friends, each of whom tells it to two other (different) friends, each of whom . . . Assuming that the actual telling occupies just one minute, and that another minute is lost in scurrying off to find someone else to tell the secret to-how many people will know after one hour has elapsed from the initial telling? By now of course you will have some idea of what is happening and won't be too surprised to learn that by the end of the hour 2,147,483,647 people would know the secret. Since this is just over one-half of the present total world population, it hardly could be called a secret any more! The same story has been presented differently by asking, under the above conditions, in a village of a given number of inhabitants, how soon would it be before everyone knew the secret?

There is a surprising growth rate in the simple matter of doubling at every stage of the sequence. Just think of it next time you fold a piece of paper in half, and don't go on for too long lest you should fall off the top!

For the curious, the exact value of 2^{100} is—1,267,650,600,228,229,401,496,703, 205,376.



A new learning activity from Creative Computing . . . .

Turtle Geometry Without Hardware

Robert S. McLean
The Ontario Institute for Studies in Education
Toronto, Canada
and
Claude and Colette Pagano
La Seyne, France

Papert's turtle (Papert and Solomon, 1972) is by now probably well known to most people who work with computers and young children.* The basic outline of the use of the turtle as a concrete environment in which to teach concepts of elementary programming and problem solving at an early age is available in the literature. The turtle itself is now commercially available, and the ideas embodied in the real turtle can be simulated on a CRT as well, often with better results.

Any consideration of the turtle has assumed the availability of a computer to support it. What can a teacher who has been exposed to the ideas inherent in the use of the turtle do if he or she doesn't have the equipment? This is the problem that two teachers in France faced last year. They worked out an analog of the computer turtle situation which uses the children themselves as the "machinery". These notes of their activity illustrate their approach and detail several additions to the usual turtle problems. While these notes are fairly explicit in the suggestion of activities for teacher and child, they are intended as a guide. Any user will certainly modify them and expand the range of possible activities.

The basic problem environment consists of an object that moves toward a target in a space containing obstacles according to specific rules of movement. The environment is concretized for the child by the analogy of the movement of a turtle which has the goal of finding its food, a leaf of lettuce. The activities are first of a physical sort, moving from normal play behavior to more conceptual, program directed activity. The first activities consist of movement of the child on the ground.

A. The leaf of lettuce is placed on the ground and a child goes towards it playing the role of the turtle.

B. Some obstacles are placed on the ground and the "child turtle" describes his movement while he is going to the lettuce.

C. The "child turtle" is now blindfolded. Another child directs the turtle towards his food: 1) by taking him by the hand; 2) by audible signals, and; 3) by oral orders, first without obstacles and then with obstacles.

During this phase, the children look along the orders given to the turtle for those which are the most useful (which give the most knowledge with the minimum ambiguity) noting that the turtle doesn't retrace his steps and does not move sideways like a crab. (If a live turtle is brought to class some days ahead of the game, the children will discover this for themselves.) The person conducting the game can point out the following three instructions: forward, turn right, and turn left.

*A future issue of *Creative Computing* will carry an article describing turtle geometry and the LOGO computer language.

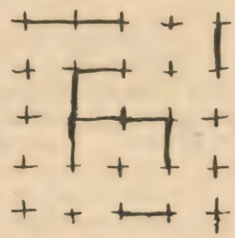
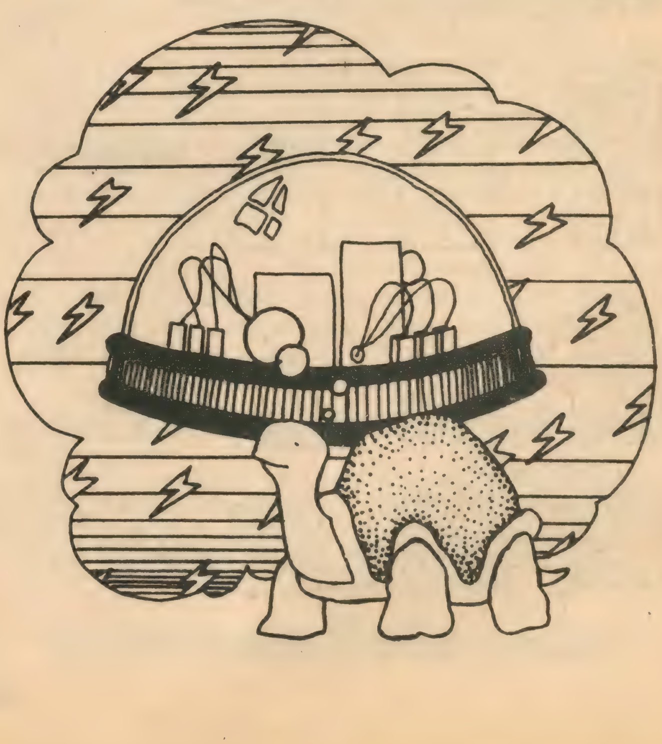


Figure 1

D. In order to make these three instructions more precise, the ground is ruled off in squares, thus forward becomes move forward one square. Turn right becomes turn 90 degrees to the right without leaving the present square. Turn to the left becomes turn 90 degrees to the left without leaving the square. By the choice of these three instructions, the movement of the turtle, which is really a continuous action, is modeled in a succession of elementary actions defining a succession of elementary states. Each square communicates with its four neighbors by an opening and each obstacle is provided by closing the boundary between two adjacent squares.

Once again, the "child turtle" first evolves and describes his task with the aid of the three instructions. Second, he executes to the extent possible the orders of another child, and third starts over with a blindfold if he can recognize by touching the borders of the squares — for example with the aid of a string placed on the ground.

more



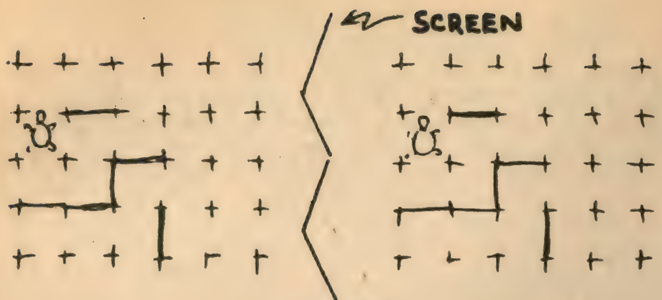


Figure 2

E. Two identical networks are placed on each side of a folding screen. One team has some obstacles in network A and communicates to the other team by messages the position of these obstacles such that the other team can produce the same networks. (It is good to let the children think about the different ways of coding the information.) When the networks A and B are identical including their obstacles, a "child turtle" is placed on each network in the same position in the corresponding squares. Each instruction executed by turtle A is transmitted to turtle B who executes it in turn. Note, if the child turtles are very advanced, one can ask them to move symmetrically on two symmetric networks.

Once the above activities have made concrete the basic geometry and rules of the game, the group can begin to consider more abstract representations.

First, a representation of the state of the turtle is invented. A picture of the turtle can be drawn and progressively stylized as in figure 3.

Figure 3

Next, the operators are encoded to give the basis of a rudimentary programming language for the turtle (figure 4). The information given by the color and by the shape of the symbol is, in fact, redundant as the symbol is placed systematically in the quadrant of the square situated at the right of the initial turtle and of the final turtle.

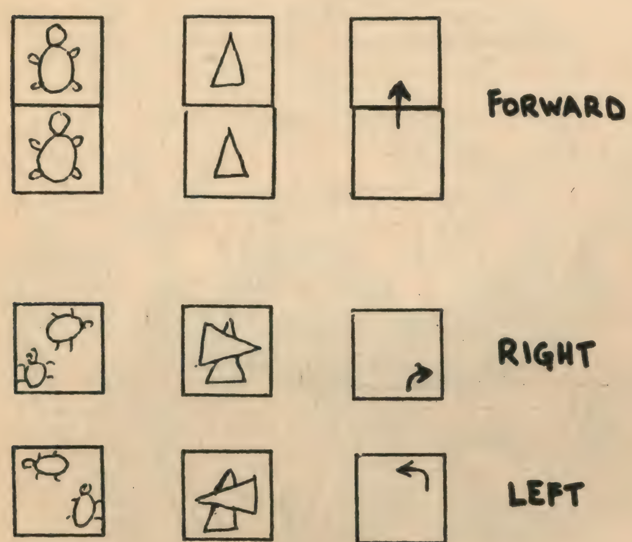


Figure 4

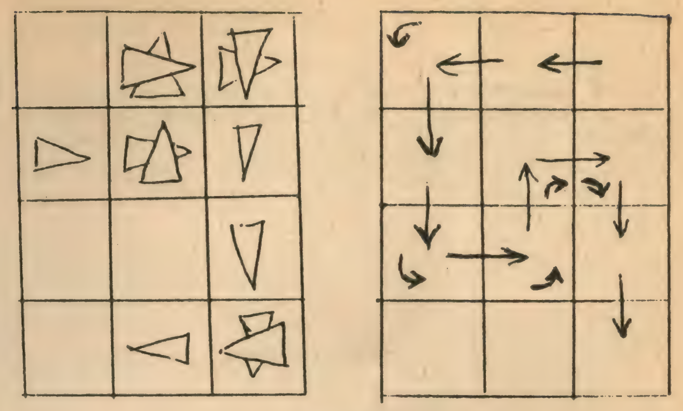


Figure 5

We can propose certain games of coding and decoding. For example, code the path of the turtle in figure 5. One can also decode a string of commands: FR, FFR, FR, FL, FF, LF, LF, LF, FR. Find a path permitting the turtle to find its lettuce in figure 6 and then code it. In figure 6, also code the path which permits the turtle to return to its initial square in its initial position.

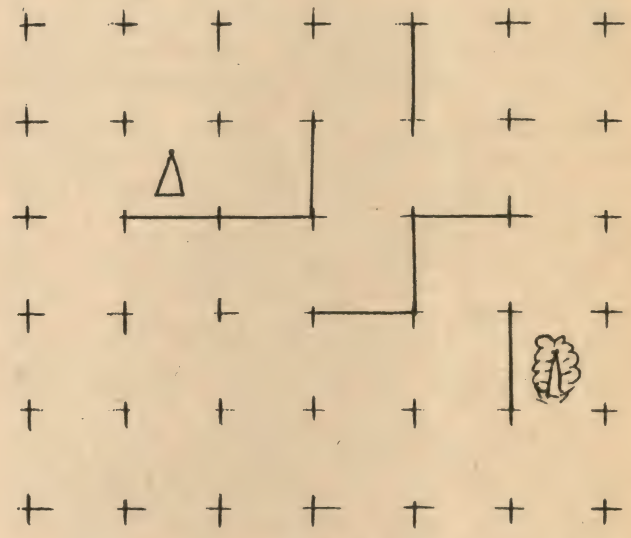


Figure 6

With these basics of coding for the turtle, the games played with the real environment eariler can be replayed with a more abstract representation. It is possible to extend the basic games in many ways; the following is a list of variations that can be used to introduce additional concepts.

We can demonstrate the isomorphism of the movement of the turtle from one square to another and that of an ant crawling along a wire as in figure 7.

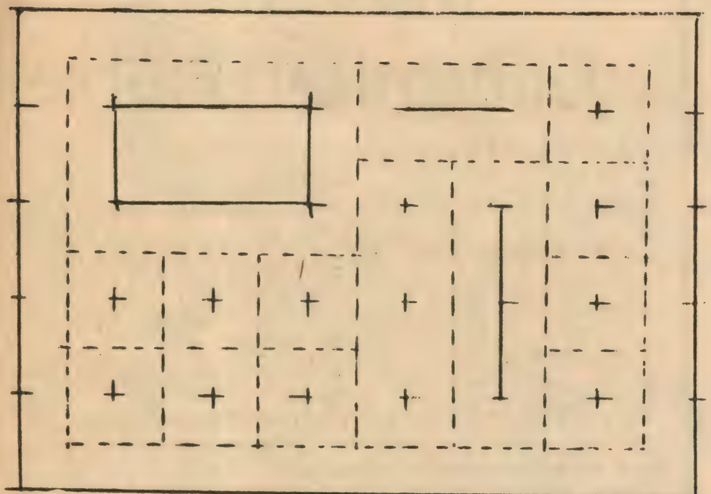


Figure 7

The notion of the length of a path can be introduced by taking as one unit the advance from one square to the next and thus the length of the path is the number of arcs followed by the ant above.

The notion of distance between two points can be introduced through the problem of finding the shortest path between two squares. One can count these shortest paths and can find the difference between squares on a grid for any case given (figure 8).

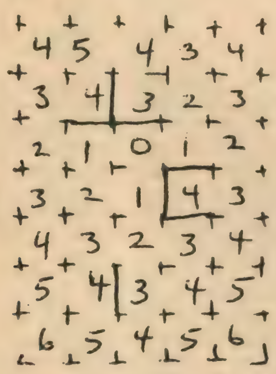


Figure 8

The notion of a hamiltonian path can be explored by finding the longest path without passing through the same square twice. There is also the question of whether one can pass through all of the squares by that means.

The notión of transportation time can be studied by associating an elementary time with each elementary action. For example, forward may take four seconds, turning right – 3 seconds, and turning left – 2 seconds. Thus one can look for the most rapid paths. There can be a race between two or more turtles subject to the constraint that they do not simultaneously occupy the same square.

One can add the possibility that the lettuce moves. This movement can be deterministic (that is to say, known in advance by the turtle) or random (a game of strategy).

The idea of a non-square network can be introduced. The turtle can move in a network of hexagonal mesh (which is equivalent to a triangular mesh for the ant) with three orders: forward – as above, right – meaning turn 60 degrees to the right, and left – which means turn to the left by 60 degrees, plus two variants (RR and LL permitted or not). Another possible variation of the networks is triangular for the turtle or hexagonal for the ant which do not permit the command FF (figure 9).

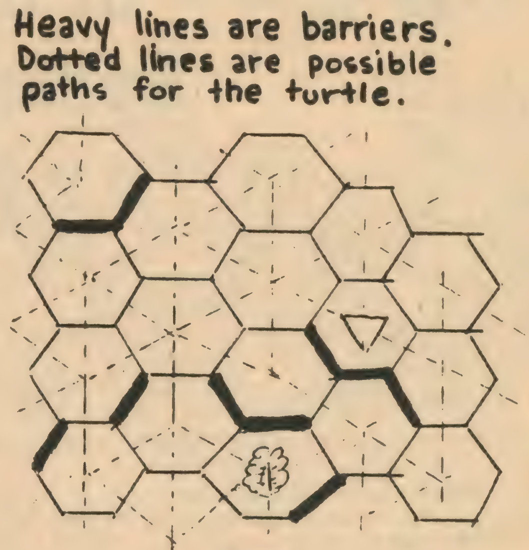


Figure 9

Conditional instructions can be introduced. The four tests of the real turtle to see if it is touching an obstacle in the front, left, right or rear can be adopted as TEST CLEAR TO FRONT (RIGHT, LEFT, or REAR). These can be adopted with the LOGO like TEST followed by IFTRUE or IFFALSE sort of structure or it can be more like other programming languages and use an IF . . . . THEN (action) type of format.

Additional forms of tests can be invented, such as test whether the turtle can see its food (by looking straight ahead) or smell it. The latter sense can be considered either directional or not, and may be a function of distance from the target.

The list of applications of turtle geometry is far from being limited. The choice of this theme in the elementary school permits many possibilities, and removing the requirement for computer equipment has made this approach attractive. In effect, each child thus establishes his program, executes it and analyzes it.

Reference

Papert, S. and Solomon, C. Twenty things to do with your computer. *Educational Technology,* 1972, 9(4), 39-42.

"Continued in their present patterns of fragmented unrelation, our school curricula will insure a citizenry unable to understand the cybernated world in which they live."
Marshall McLuhan – 1964

Still a Few Bugs in the System

It bugs us here at *Creative Computing* when the mass media blame various problems on the computer. Even people in government, business, and schools find the computer a convenient scapegoat for problems actually caused by a programmer, keypuncher, faulty data collection techniques or other non-computer facets.

In this continuing column, we'll reprint articles or quotes which blame various catastrophies or problems on the computer. It's up to you, the reader, to decide whether the computer is actually to blame. Also, if you spot an appropriate item for the "Bugs" column, please send it in.

A COMPUTERIZED bill had this notice on the bottom: "Failure to receive this bill is no excuse for non-payment of the amount shown below."

Chicago Tribune

SHREVEPORT, La. (AP) – Gas rates have gone up like everything else. Just ask Ruth Brister. Her bill went from around $14 last month to $42,474.58.

"I flipped completely," she said.

"The computer went haywire and some of those bills got out," a customer representative at Arkla Gas Co. explained.

Toledo Blade, 3/30/75

National Student Computer Fair

In conjunction with the National Computer Conference to be held in New York during June 1976 there will be a student computer fair. Unlike previous years, this will *not* be a "computer science" fair, but will have a much broader scope including entry categories such as public access computer projects; creative fiction, poetry, and art about computers; computer applications in the home; computer games and recreations; hardware projects; etc.

Furthermore, the fair will be truly national in geographic coverage. The committee is seeking the cooperation of local and regional computer and computer education groups to hold preliminary fairs and evaluations of projects for entry into the National Fair. Associations, consortia and groups such as BIT, NCTM, MECC, Lawrence Hall, LIRICS, etc. have been or will be contacted to perform preliminary judging. If you are a member of a local group and wish to be included in this process, please write to the National Student Computer Fair Director: Dr. Sema Marks, Director of Academic Computing, CUNY, 535 East 80 St., New York, NY 10021.

'76 NCC • NEW YORK • JUNE 7-10, 1976

Students Stuff the Contest Box

by Robert Meyers

PASADENA, Calif. – It was enough to crack the golden arches.

Twenty-six science and math students at California Institute of Technology here, looking for something to do while studying for final exams in March, stuffed more than 1.1 million computer-printed entries into a give-away contest sponsored by the McDonald's hamburger company.

When the drawings were held about two weeks ago, the students had won 20 per cent of the total prizes, including a $7,000 car and $3,000 in cash. McDonald's promised to change its rules.

"It's amazing how much free time you can find during final exams week when you're really looking for it," said Steve Klein, 21, a junior information sciences major.

Klein and Dave Novikoff, 21, Barry Megdal, 19 and Becky Hartsfield, 18, all students at the science-oriented school here, were intrigued by the give-away contest being sponsored by the McDonald's Operators Association of California. The rules called only for an entry to be printed on a 3x5 inch card, by a person who was over 18 with a valid driver's license. "Enter as often as you wish," the rules invited.

The students did. In late March the 26, all members of Page House, a residential and dining facility, spent $350 to buy about 20 hours' printing time on an IBM 370/158 computer. They produced 52 boxes of paper, each box of which contained 2,700 pages, and each page of which contained eight valid McDonald's entries.

"There were 1.2 million entries at first," Megdal, a sophomore electrical engineering student says, "but by the time we got through cutting the paper up into individual entries, there were only 1.1 million."

Each of the 26 students involved in the tension-breaking project thus found that the computer had printed his name 40,000 times. Dividing up into eight groups, the students took their ballots to 98 of the 190 participating McDonald's stores in Southern California.

When the management of the fast-food chain learned of the prank, its reaction was hot enough to sizzle a french fry.

". . . The students acted in complete contradiction to the American standards of fair play and sportsmanship," boomed a press release. "Their actions had the effect of depriving individuals and families of improved odds of winning the prizes."

The company reported getting letters from outraged citizens. Newspapers and television stations sent reporters to sniff out the story. Burger King, a fast-food competitor of McDonald's gleefully gave Caltech a $3,000 scholarship in honor of the stunt.

McDonald's, however, spent a great deal of time trying to figure out what action to take with regard to the computer-printed entry forms. The company finally decided to honor them all, but to give duplicate prizes to the general public for every Caltech entry that was drawn.

That action cost the participating dealers an extra $10,000 on top of the $50,000 already allotted.

The prizes were presented last Tuesday, May 20. Becky Hartsfield, a freshman physics major, was given the keys to a new Datsun 710 station wagon, which she immediately turned over to a chapter of the United Way.

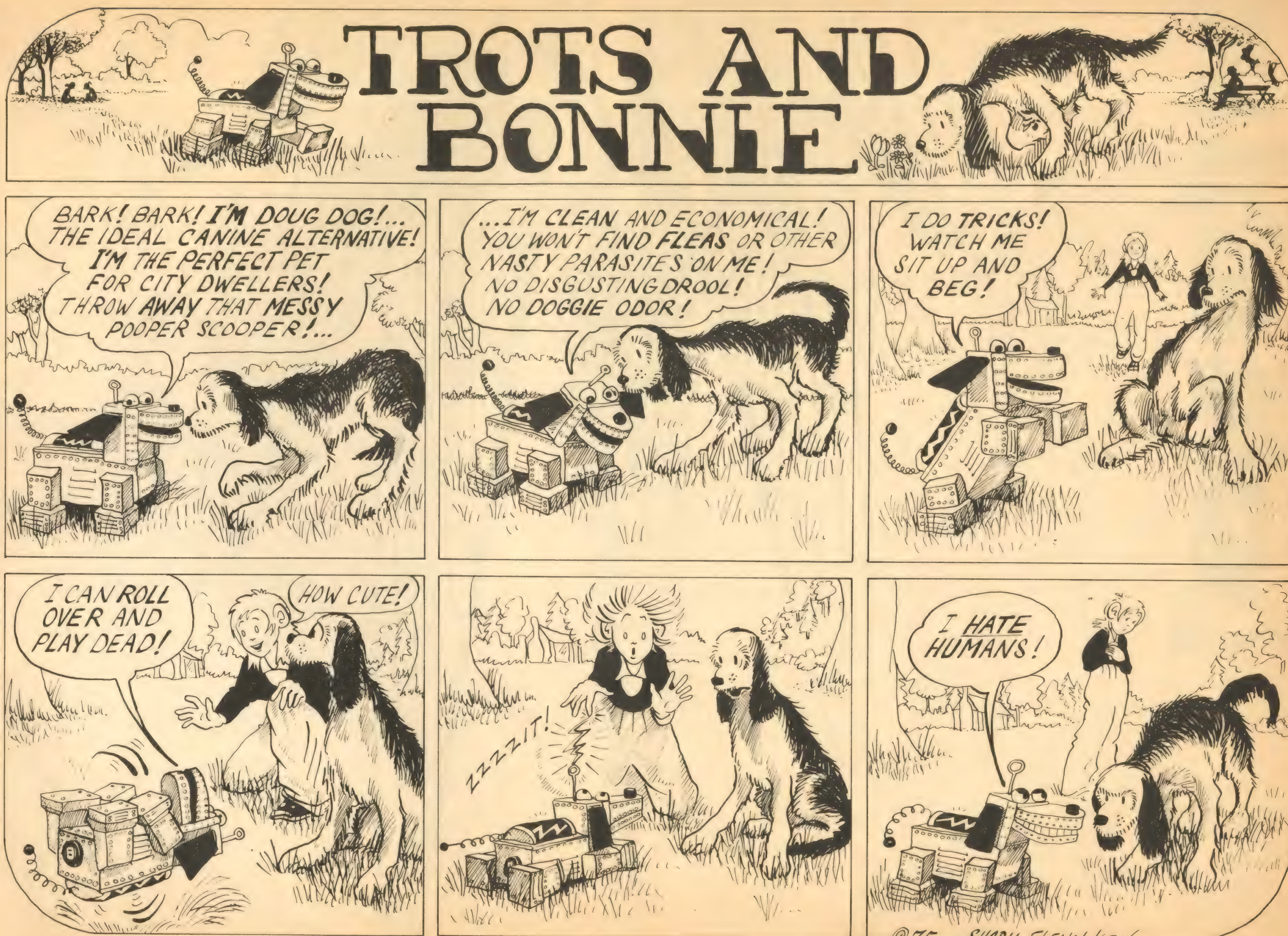
The top prize – a more expensive car and a year's supply of groceries – went to a non-student.

The students say they will keep the check for $3,000, and use it to pay for the taxes and license on the station wagon, to improve their living quarters, to buy micro-wave ovens for the house, and to pay off the cost of buying time on the computer to print the entries in the first place. "No one will make a profit on this," Novikoff said.

At the awards presentation on Tuesday, Novikoff invited Ronald McDonald, the clown character who represents the hamburger chain, to have dinner that night at Page House.

Ronald, however, ate elsewhere.

Washington Post 5/21/75. Thanks to Nelson Griggs, Boyds, MD for sending us the clipping.


TROTS AND BONNIE
BARK! BARK! I'M DOUG DOG!... THE IDEAL CANINE ALTERNATIVE! I'M THE PERFECT PET FOR CITY DWELLERS! THROW AWAY THAT MESSY POOPER SCOOPER!...
...I'M CLEAN AND ECONOMICAL! YOU WON'T FIND FLEAS OR OTHER NASTY PARASITES ON ME! NO DISGUSTING DROOL! NO DOGGIE ODOR!
I DO TRICKS! WATCH ME SIT UP AND BEG!
I CAN ROLL OVER AND PLAY DEAD!
HOW CUTE!
ZZZZIT!
I HATE HUMANS!
©75—SHARY FLENNIKEN

creative

computing

"Computer Art for People"

Draw the Bug From the Computer and Win Prizes! ! !

Ever thought of the very worst kind of computer bug or program bug? One that is really frightening. Is it one that eats your program away from inside a DO or FOR loop? Or is it a nasty disk file identifier insect? Does the thought of these bugs scare you? Or do you try not to think about their horrible grizzlyness?

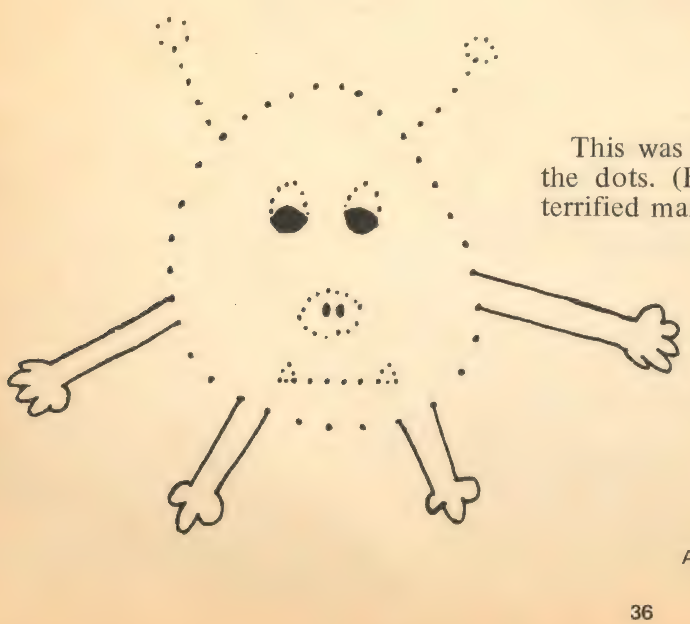
If you can come to grips with yourself to draw that which when seen by someone will give them nightmares for the rest of their life you can win. But it takes more than artistic talent. It takes raw Guts with a capital G. But if you have the sensitivity and, of course, guts, you are eligible to win BIG, BIG prizes. Chances are you may have already WON!! How does that grab you?! Read on!

All you need to do is think of the ugliest, scariest computer bug you could ever imagine to meet . . . or not want to meet!!! And draw it. That's all. Easy, isn't it?!

So hurry up and grab a pencil. Let your inner fears pour out before the closing date of October 20, 1975. Entries received after the deadline will be automatically disqualified and incinerated.

INSTRUCTIONS

1. In the space provided for you at the right, draw your computer bug.
   a. The bug must fit within the space provided.
   b. Color drawings always welcome, but please do not use ethnic colors, i.e., black, red, yellow, or white.
   c. Only one entry per family. Two or more will be regarded as cheating, and all said entries will be disqualified and incinerated.
2. Enclose a signed photograph of yourself. This is for identification purposes. And if your drawing loses, your picture may win!!! Ha-ha!!!
3. Be sure to send $5.00 with the drawing. This fee covers "processing charge". This includes mailroom assortment, judging by our panel of distinguished computer program bug exterminators, and incineration of drawing if disqualified. Losing drawings will be returned. We regret that the $5.00 will not.



This was last year's winning drawing. Just join the dots. (Had we drawn it out, we could have terrified many of your friends.) – DHA

Adapted from something similar in *The Harvard Lampoon.*

On Computer Languages

"David Ahl, I want to punch you in the nose," was my greeting from Adele Goldberg at NCC in Anaheim. Perhaps she most vividly summarizes the frustration that some of our readers have with the high percentage of BASIC language material we run in *Creative Computing.* Adele is at the Xerox Palo Alto Research Center where "Dynabook" with the tremendously rich and powerful language, SMALLTALK, has been developed by Alan Kay.

Adele is not alone in her view as other letters and conversations suggest. Let me try to briefly summarize the various arguments and positions that have been expressed to me over the last few months pertaining to various languages from their supporters.

ALGOL. The major language in Europe. Surely *Creative Computing* doesn't wish to ostracize its growing body of subscribers in 11 European countries.

APL. A powerful, sophistocated language with a rich vocabulary and conciseness of expression equalled by few other languages.

Assembly Languages (Various). Sophistocated users are soon going to outgrow BASIC or other high-level languages and there is much to be gained from learning to program on the assembly or machine language level.

BASIC. Accounts for nearly 70% of the usage in schools and colleges today. An easy-to-learn language which "everybody" knows.

COBOL. The major language of the business data processing community. Doesn't *Creative Computing* have an obligation to serve the person heading for a career in EDP?

FORTRAN. The major language of the scientific community and still the only (or main) language on many college campuses. Also, the first widely-used high-level language with all that such a history implies.

LOGO. First developed at Bolt, Beranek and Newman (Wally Feurzeig) with further work going on at MIT's AI lab (Seymour Papert) and General Turtle (Alan Papert). Part of the "Mathland" approach for introducing young students to mathematical concepts.

PILOT. A BASIC-like language oriented to young children or other "naive" computer users. Has been used successfully in Montessori schools with preschool age children. Can be run on many minicomputers.

PL1. Different language but substantially similar comments as APL.

SMALL TALK. An incredible "gee whiz" language with fantastic graphics and animation capability. Has to be seen to be appreciated (or believed).

I haven't heard anything from fanciers of PLATO, SNOBOL, RPG, FOCAL, etc. but I'm sure I will sooner or later. However, the point is this: *Creative Computing* is attempting to serve a broad cross-section of the educational computing using community – middle and secondary school students and teachers, college users, public access groups (libraries, museums, storefronts), researchers (to a limited extent), and the general public (to a limited extent). So what mixture of languages should I run? Frankly, I don't exactly know. My gut feeling is that it should be mostly material that will run on the majority of subscribers' computer systems. But also articles and information should probably appear about less popular languages to expose people to other alternatives. For the most part, I'll probably avoid specific machine languages, languages specific to one computer (e.g. NEAT), EDP languages (COBOL, RPG), and other specialty fare. I don't believe that *Creative Computing* should promote any one language as we are currently accused of doing with BASIC. However, I do have the very real problem that I can only print what is submitted.

As the readership of *Creative Computing* grows in sophistocation and your demands change, I'll do my best to remain responsive. A diversity of manuscripts and your thoughtful cards and letters will help keep me on the right track.

–DHA

Toward A Human Computer Language

Alexander B. Cannara
IMSSS, Stanford, California

Why must (should) computer-programming languages be different from human languages? Please trust that this question can be linked intimately to our topic: "computer languages for education". But first, I'll argue that "computer languages" and "for education" redundantly describe that topic: (1) people use computers, thus computer languages, for work or play; (2) anyone engaged in productive enterprise (work or play) now and then learns something new (to him), so programming can be an educational experience; (3) because the computer is a very general tool in the realm of human thought, it is really an educational tool for all people. All designers of programming languages should keep that in mind. We want a computer language to allow us to define objects and their interactions as naturally as possible, so that we can set them off on computational explorations of worlds that we, not the language, constrain.

Obviously, people are at the center of the relationship between the computer and humanity. We design computers and define languages for using them, and these programming languages allow us to communicate our thoughts not only to a machine, but to ourselves and to other people as well. Some argue that the latter is the more important function. In any case, any language is for communication and a limited language limits the communication of those who use it.

Programming languages (or any other formal languages), are special, in contrast with human languages, because we do not yet understand our own language processing well enough to be able to construct any reasonable facsimile of it within or without Turing-computable limitations. In other words, no one has written a computer program that could pass the "Turing-test" and sensibly communicate with a human in human language over arbitrary scope and time. Research with that kind of goal in mind now aims at building upon success in limited contexts and is often coupled with psychological models of thought, memory, etc. (e.g. T. Winograd, Psychology Today, May 1974; or J. S. Brown, Proceedings of the ACM 1974 Conference). So a partial answer to the opening question is: "computer languages cannot yet be as broad as human languages".

Given that programming languages are designed by sometimes frail, sometimes insightful humans who cannot describe what it is they do to communicate, but want to make a machine do it, what limitations of current programming languages should make us unhappy and spur

us on to better designs? And, since every programmer was once a novice, shouldn't we be concerned with the nature of a programming language's structure and the ease with which its full powers can be grasped and applied to problems that its users wish to solve? I'll use some notable failures in language design to illustrate things with which we should not be satisfied and to show why the answer to the latter question should be an emphatic "yes".

Almost everyone is familiar with one of the more primitive programming languages like Fortran and Basic (proper nouns both, and deserving of no more capitals than God and Man). Why is "primitive" a justifiable modifier? For example, consider Fortran (historically an acronym for "formula translator"). Created as a way of making algebraic expressions more easily entered (by humans) into a machine, thus more readable than lists of individual machine instructions, it served to express formulas in notation consistent with common mathematics — the same notation for all machines having a Fortran-translator program (compiler). Two decades ago, this was a step in closing the gap between computer and human vernaculars. But even then it was clear that a useful machine, which can parse "(2appl es+3lemons)/1.8fruitperpound", must also handle control commands, like "do this then do that until something happens", with clarity and ease. Hackneyed design gave Fortran, among other defects, less power to express the control of a program's execution than to parse an algebraic expression, and so eternally placed arcane and unnecessary burdens upon its users. Similar remarks apply to Basic, originally designed as a simple, stopgap recoding of machine language for users of early time-sharing systems. Some examples:

English/Algebra

If (2 apples + 3 lemons) / 1.8 fruit per pound is > 0 then do this, if it is = 0 then do that, otherwise do the other thing.

Fortran

```
      FRUTLB = 1.8
      IF ((2*APPLES+3*LEMONS)/FRUTLB) 100,200,300
100   . . .
200   . . .
300   . . .
```

Basic

```
10    LET X = (2*A+3*L)/1.8
20    IF X < 0 THEN 100
30    IF X = 0 THEN 200
40    IF X > 0 THEN 300
100   . . .
200   . . .
300   . . .
```

Algol

```
real x, fruitperpound; fruitperpound := 1.8; if (x :=
(2*apples+3*lemons)/fruitperpound) > 0 then this else if
x = 0 then that else theotherthing;
```

The Fortran and Basic examples highlight the glaring inconsistency of easy algebra and sorry obfuscation of control. Suppose we had to write the algebra as:

```
10    2
20    APPLES
30    *
40    3
50    LEMONS
60    +
70    1.8
80    /
```

Would or should we be happy with such a lobotomized language? Perhaps for a $20 calculator. And never mind trying to recall what goes on at lines 100, 200 and 300, which may be pages away (or cards away on archaic systems) in the program's text. Yet many people, deprived of access to anything else, believe that Fortran, Basic, etc. are the essence of programming! Perhaps our true shame is that we inflict our most scurrilous languages upon our school-children, in whose lives the vast potential of the computer will have greatest meaning. We've mused that there is no complete and satisfactory language, not yet — certainly if too many of us continue to accept naively the ravings of foppish languages there may never be.

It's obvious which of the examples above (Algol, over a decade old) most closely approaches English and some of this is just nice form. We naturally like to call things (variables, labels, procedures) by names that indicate to us their significance. Notice that Fortran and Basic limit even this simple ability by restricting name lengths and character content (it is irrelevant that different versions of these languages impose different limits). Utter trivia like non-free-field syntax (e.g. Fortran statements must begin after column 6 (or is it 5?)) further subordinate the programmer's daily convenience to the one-time convenience of the language designer. Sometimes, on some machines, some such restrictions result from necessary trade-offs. But even on the smallest machine, a compiler or interpreter that reasonably sacrifices niceties of form should never be allowed to sacrifice the essentials of consistency and intelligibility.

Notice that Fortran misappropriates "=" as the assignment operator and leaves the user with either: "IF (X — 2) 100, 200, 100" or "IF (X .EQ. 2) GO TO 200" as means for testing equality (in this case of X and 2). Basic at least retains the usual meaning of "=", but at the price of adding an extra, prefix operator "LET". As a result of these kinds of patchy designs, such languages are poorly defined and give the user no consistent means for mastering their syntax. Special cases abound. Locally, the syntax may be postfix, prefix or infix and the same element (e.g. "IF" in Fortran, "=" in Basic) may have different meanings, depending upon its context. Human languages do the same sort of thing, but not at such low levels.

Fortran "format" statements provide perhaps the most horrendous example of a language patched within a language. Mastery of their syntax alone has provided jobs and income for many programmers. Interestingly, the tendency is to modify Fortran and Basic by adding "features". Since the languages have no particular structure anyway, these are cut-and-paste enterprises. One notable result has been Pl/1, which fearlessly combines the syntaxes of Algol, Cobol and Fortran into something like the worst of all possible worlds! It solves few of the inconsistencies and frivolities of either Fortran or Cobol (e.g. noise words which seem to mean what they mean in English but mean nothing!) and abuses a well-defined language: Algol-60. Unfortunately, Pl/1 seems to be the product of bigger-is-better reasoning. When a language is designed in such a way as to make it harder to read or to explain, then we should be suspicious that it lacks usefulness as well as elegance. It is of course true that not everyone who uses a computer has clear, elegant languages at his or her disposal, but we should all be aware of needs and opportunities for changing such situations. Here are two examples of a guessing game written by children in Basic and Logo (which borrows ideas from Algol and Lisp):

Basic

```
10 PRINT "DO YOU NEED INSTRUCTIONS?"
20 INPUT A$
30 DIM A$(1)
40 IF A$="N" THEN 80
50 PRINT "PLEASE GUESS A NUMBER BETWEEN 1 AND
60 PRINT "THEN I WILL GIVE YOU A HINT"   100,";
70 LET X=INT((100*RND(0)+1)
80 INPUT R
90 IF X=R THEN 150
100 IF R < X THEN 130
110 PRINT "GUESS LOWER"
120 GOTO 80
130 PRINT "GUESS HIGHER"
140 GOTO 80
150 PRINT "YES!"
160 GOTO 10
```

Logo

```
TO START
10 PRINT "DO YOU NEED INSTRUCTIONS?"
20 IF IS REQUEST "YES"
   THEN PRINT "PLEASE GUESS A NUMBER BETWEEN
   1 AND 100, THEN I WILL GIVE YOU A HINT"
30 QUIZ RANDOM 1 100 REQUEST
END

TO QUIZ :X: :GUESS:
10 IF EQUAL :X: :GUESS: THEN WON ELSE LOST
   LESS :GUESS: :X:
END

TO WON
10 PRINT "YES!"
20 START
END

TO LOST :TOOLOW:
10 IF :TOOLOW: THEN PRINT "GUESS HIGHER"
                ELSE PRINT "GUESS LOWER"
20 QUIZ :X: REQUEST
END
```

To which language should we prefer to expose ourselves or our children? Keep in mind that most machines including microprocessors have long had all the instructions needed to easily realize Algol-like control, which Fortran-like languages use only for parsing formulas. A decade ago Fortran was wasting abilities of most machines it ran on, now it and its relatives are simply more wasteful of more machines.

The actual range of computations and data-types could also be a topic of discussion (e.g. Fortran and Basic do numerical work easily, string manipulations with difficulty). Apart from attempting elegance, some languages serve particular computational audiences (e.g. Snobol for strings, Lisp for lists/functions). However, nearly all languages suffer idiosyncratic limitations when type definitions lend predisposition of meaning to data. Profusion of data-types is all right as long as it does not hamper the user's own constructions. The enterprise of reality or programming is message processing and the effect (meaning) of a particular message depends upon the action of the computational chunk (object) that receives it. This merges with the idea of an extensible language in which the user can construct new entities which truly expand the power of the language. Procedure definitions in Algol, Pascal, Logo, etc. constitute only a beginning (subroutines in Fortran and Basic are but trivial excursions into machine language, because they lack even the syntactic status of primitives). Education and computers can be linked via language extensibility.

We want the machine to understand us in our vernacular or a sensible approximation of it. We want to phrase programs as solutions to our problems in clear, powerful prose. For these purposes, no existing language may ever be satisfactory as it stands. Including English, etc.! Thus the need for extensibility — syntax and semantics being under our control during the course of a computation. Languages which allow the creation of new types of objects and new examples of existing types have been around for some time, Simula for instance. Generally their births were motivated by desires to model real-world systems by simulation (e.g. of forests, hospitals, etc.). Their abilities to create new types and examples of objects that intercommunicate at will allows one to begin to program with a greater sense of reality. But restrictions remain, Simula is fixedly Algol-like, Simscript is incomprehensibly couched in Fortran syntax, and so on. Perhaps the ideal is a language which captures the essence of "objectness" and intercommunication, yet is itself an editable object accessible to its users. Exciting work along such lines is being done by some people and will surely leave a mark on our computational future.

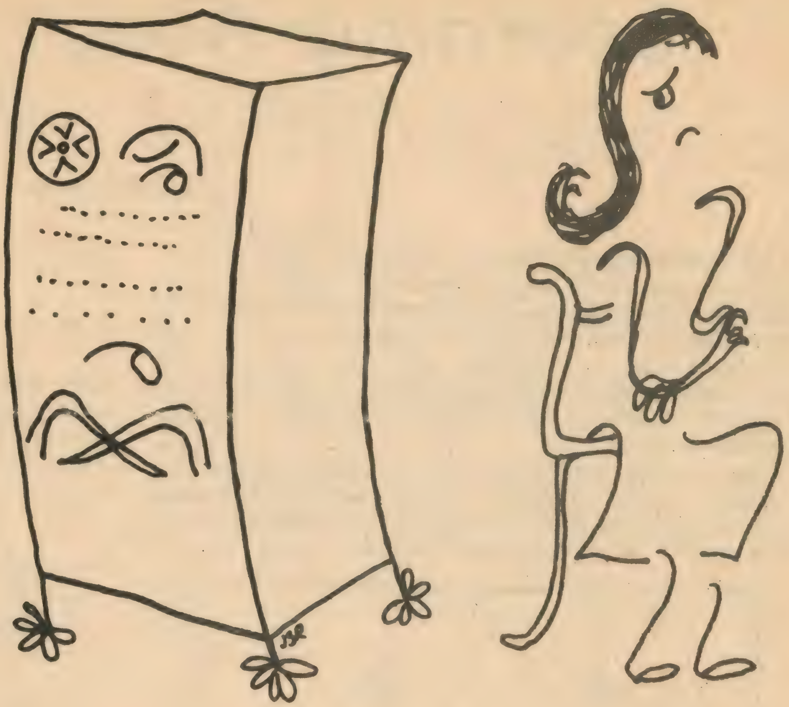
So, the rest of the answer to the opening question is: "certainly every effort should be made to breach the gap between computer and human language". Particularly in educational settings, where it is paramount that the student's language be the language through which learning proceeds. The goal isn't English, or any other known language in particular, but the ability to construct whatever objects/languages one wants.

[Ed. Note: The next issue of Creative Computing *will carry an article on extensible languages (such as ECL) as powerful programming tools. The article will also discuss communication between different computer languages (such as Fortran and Basic). — DHA]*

T-SHIRT DESIGN (Albert Einstein)

To order, see page 2.

How to Cope With Your Computer



Susan S. Most
Cape Elizabeth High School, ME

It cannot think. It's just a big metal box with a mess of wires inside and a myriad of lights continually flashing on and off. Ingenious device that it is, that box is capable of solving infinitely complicated problems in less time than one's lips can say "computerman". Alas, but if you don't know how to operate it, you might as well take it to the dump.

One cannot imagine the sheer frustration a person feels when faced with the task of operating a computer for the first time. You're standing in a room where people look disgustingly productive with yards of yellow paper spewing from their teletypes. A continuous clicking, clicking noise comes at your ears in such an important and official manner that you shiver a bit at the idea that perhaps someday you might be in charge of one of these "things". Your feeling of impending importance is burst like the rush of air from a pin-pricked balloon.

You can't even read what it prints, to say nothing of telling it what to do. Bits of incomprehensible brilliance flow forth each fraction of a second as the carriage on the teletype races back and forth; and the lights on the front panel merrily wink at your ignorance.

It's not a question of just saying in plain English what fantastic chore you want it to perform — this "box" doesn't understand plain English. Since the box really has no "brain" inside — only wires; the operator must learn "the tongue" that it does understand. This code includes such startlingly clear terms as TAD, DCA, JMS, etc., and those are cinchy compared to 7402 and, how's this — ready: 111 000 010 101. Imagine a number like that meaning something to anybody. Ridiculous, you think? Well, to the box it's crystal clear. That is, of course, assuming you're using a computer that understands that language. To add another fly to the ointment, there are many languages which are not interchangeable with machines designed by other companies.

The most helpful tool to the harried individual attempting to coexist with a computer is a diagram called a "flowchart". It's not a plumber's helper nor a sea captain's guide. Rather, a flowchart is a way to clarify on paper exactly what sequence of calculations you want the computer to carry out. At the same time it helps you chart your train of thought, assuming of course there is some flow to your ideas.

Once the flowchart has been written, don't think telling the computer to follow it will be a cinch — that's the hardest part. Being a brainless genius, the computer must be told exactly, in the correct order, what to do. Coding is not the language of martians, but an analyst's shorthand — usually the initials of processes one wants the computer to do.

Yet understanding a code and having written a flowchart still cannot guarantee success. Writing a program containing all the operations to get the correct answer can cause such frustration that one could be driven to his nearest analyst — in psychoanalysis, that is. This novice programmer nearly climbed a flagpole trying to think "like the box". Take heart, however, eventually you win.

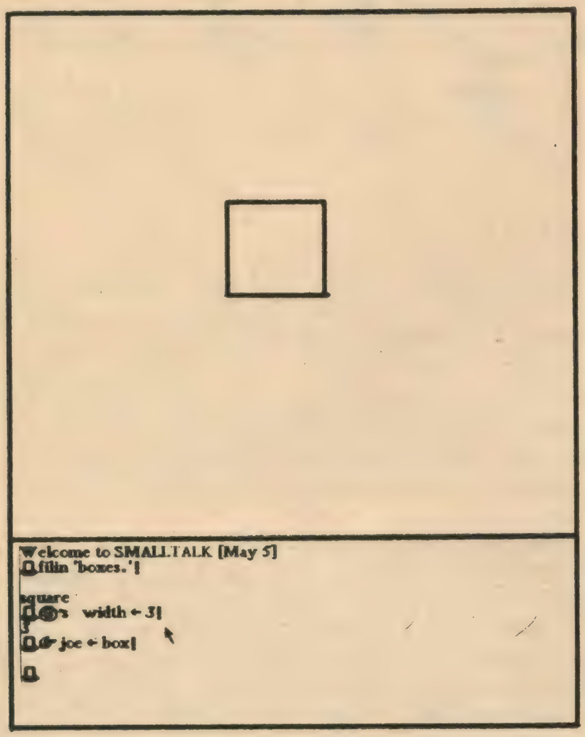
Learning About Smalltalk

by Marian Goldeen

My name is Marian Goldeen. I'm an eighth grade student at Jordan Junior High School in Palo Alto, California, and I would like to tell you about how I got started working with computers at Xerox and the class I taught.

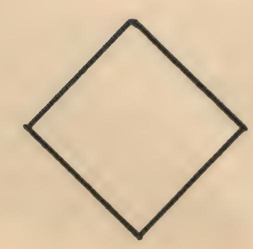
It all started in December, 1973 when I was in the seventh grade. There were four people in my class who were interested in taking a course about the computer language Smalltalk at Xerox.

When we first started we were shown how to start the machines up, and file in our one file, which had already been written onto our disks. These files contained some programs that would draw boxes like this.



These boxes could turn on their axes,

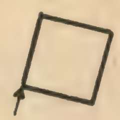
grow,



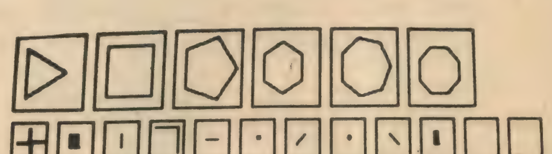
and shrink.

Later on we learned how to change the programs which had been created and drew these boxes so that we could do different things with them, for instance, move them to different places on the screen.

There was a little rectangular object to the side of the keyboard, called the mouse. When you moved the mouse around a corresponding pointer on the screen moved around too. We learned how to make the boxes follow the mouse pointer.

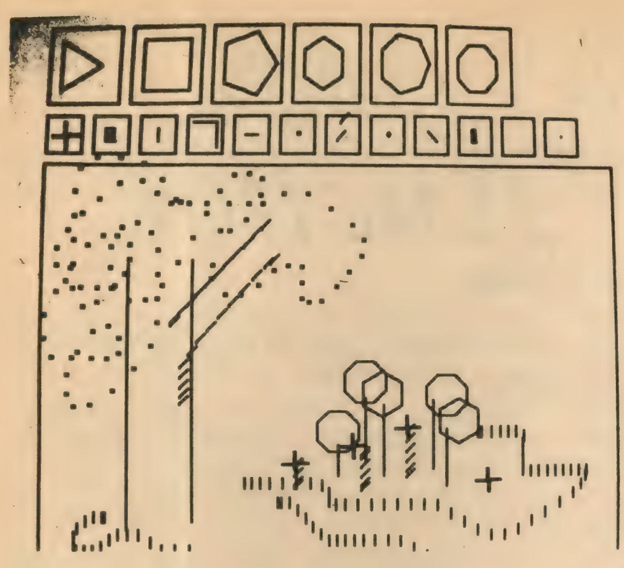


After we had learned just about everything there was to know about boxes we were able to create our own programs (Gulp). I don't know what the two boys in the class did, but Colleen and I created a painting program. It was fairly complicated. To run it you first had to set up the menu.



You would point with the mouse to the box that contained the shape you wanted to draw with, then press the top mouse button. Now the shape would be a paint brush and you could draw pictures.

"Smalltalk" is an interactive language developed by Alan Kay at Xerox Palo Alto Research Center. It is easy to learn and use, has powerful text display and manipulation capabilities, extensive graphics and animation features, and very high computational speed on several parallel levels. It runs on a small computer specifically dedicated to running Smalltalk and has not, at this time, been implemented on any general purpose computer. We'll have more on Smalltalk in future issues.

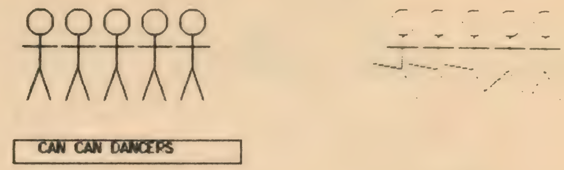


By now school had just gotten out and we had stopped taking our Smalltalk classes for the time being. At the beginning of July a class was organized for Barron Park kids and I came up too, to help tutor. These kids started at about the same place I had but since they're not too important for the moment I'll go on about me. I started working very hard on stickfigures and finally came up with something that looked like this.

The headless horseman, eh?! Well, I decided that a stickfigure without a head would never do. So I gave it a head.

Now, what's so great about a stickfigure if it won't do something for you? I taught my stickfigures how to play baseball.

Then, I decided that what I really needed were some can-can dancers.

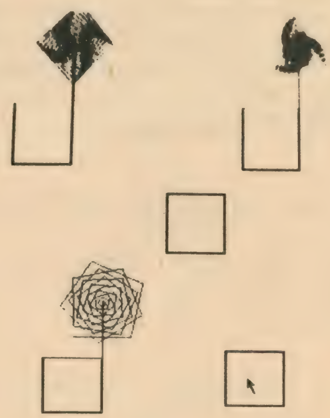


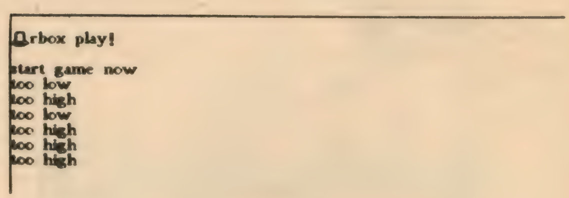
After that I worked the original dance routine in so one figure danced a solo!

At about this time I noticed that out of all the kids from Barron Park (about twenty) only two or three were girls. Horror of Horrors!! So I decided that when school came around I would teach a computer class for girls only!

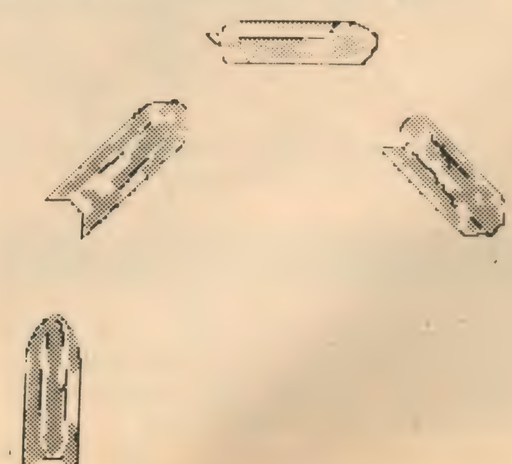
When school started that's exactly what I set about to do. Unfortunately the woman I arranged it with convinced me that there should be boys in my class. I ended up with three girls and two boys.

I started them off with boxes. Then they, just as I had, went on to do their own programs. Lisa started off with a guessing game,

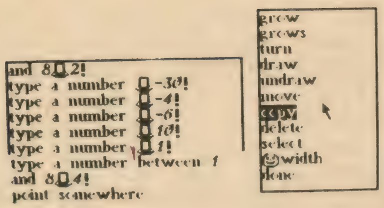




Kathy did some rocketships and *oddshapes,*



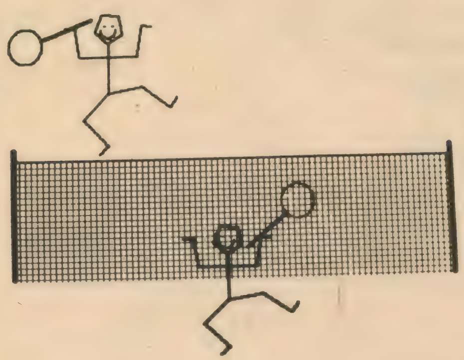
and Susan started a painting program.



A few weeks later Lisa, having finished her guessing game, joined Kathy on the rocketships.

At this point I stopped teaching them and they started working on their own. I went back to my stickfigures. I decided that these figures needed to have elbows, knees, hands and feet. To enable the computer to do this faster I changed the head into a pentagon.

With a lot of elaboration my figures can now play badminton.



Now I'm working on a galaxy simulation where I can selectively send different stars into nova.

"The computer is incredibly fast, accurate, and stupid. Man is unbelievably slow, inaccurate, and brilliant. The marriage of the two is a force beyond calculation."

Leo Cherne

Life Auction

THE AUCTION

This is an interesting and enjoyable exercise. Each person is asked to rank in order the following items:

Your ranking:	Item:
______	1. Ability to be self-sufficient
______	2. Active and satisfying life
______	3. Ability to influence others (ideas)
______	4. Ability to draw love from others
______	5. Power over things (fix cars, grow vegetables, program computers, build boats, etc.)
______	6. Ability to be a caring person
______	7. Active and satisfying athletic life
______	8. Opportunities for risk and adventure
______	9. Intellectual ability
______	10. Good health
______	11. Wealth
______	12. Approval by the opposite sex
______	13. Intellectual stimulation
______	14. Physical attractiveness
______	15. Prestige (not "social") family life
______	16. Ability to initiate and maintain friendships
______	17. Resilience (ability to bounce back
______	18. Ability to give love
______	19. Socially significant activity
______	20 Close and supportive
______	21. Artistic ability

Then in small groups of three or four, members share their lists and discuss what each considers to be really important in life. Then the larger group comes together and the auction begins. Each person is given 20 chips or cards, each worth one thousand dollars. Each item is auctioned off and in the spontaneity and excitement of the auction, the participants discover what they think is really important to them. They also learn a great deal about others in the group.

I LEARNED THAT I . . . .

After completing the above exercise, complete one or more of the following statements:

I learned (or relearned) that I . . . .

I noticed that I . . . .

I was surprised to see that I . . . .

I was pleased (disappointed) that I . . . .

Because it is important that everyone be as open and supportive as possible, it is essential that any member feel free to "pass" at any time.

Try writing a computer program to analyze the responses of the class to these exercises.

(The above excerpted from an article by Jim Wilson from the January 1973 Loomis-Chaffee Bulletin.)

A new learning activity from Creative Computing . . . .

What Do You Value?

by Sally Richards

There's a great deal of talk (and action, too) these days about values; value clarification, strategies, the processes involved in "valuing," and about the revaluation of values. Based on the premise that our lives, our surroundings, and our roles are constantly shifting and changing, it becomes critically important that each of us knows what we value. That is, we must know who we are, what we want and where we're going.

Values have become a central issue in school, home, church, and business. Value clarification is being sought extensively as a vehicle for achieving personal growth and fulfillment, as well as a basis for decision making and problem solving.

How does one know what to value? How do you know what you are for or against?

In these troubled, confused, but also exciting times, we need people who know who they are, who know what they want out of life, and who can name their names when controversy rages. People who are not so vulnerable to a demagogue, or to blandness, or to safety. Values are the basis upon which people decide what they are for and against, or where they are going and why. In other words, they give direction to life. But, with the many divergent values viable today, it is oftentimes difficult to know where you stand.

The following exercises (sometimes called strategies) can be used to help you determine what is important to you? What do you value? After doing these yourself, you can perhaps try them on friends, family, or students and compare some of their ideas and alternatives to your own.

As quickly as you can, list 20 things in life which you really, really love to do. There are no right or wrong answers about what you should like.

Using the suggested code below, the next step is to code the 20 items listed above.

1. Place the number 1 by any item which costs more than $5 each time you do it.

2. Put a 2 by any item which involves some RISK. The risk might be physical, intellectual, or emotional. (Which things in your own life, that you love to do, require some risk?)

3. Using a 3, record any of the items on your list you think your FATHER and MOTHER might have had on their list if they had been asked to make such a list at your age.

4. Place either a 4 or a 5 next to each item; the 4 is for items you prefer doing with PEOPLE, the 5 for items you prefer doing ALONE.

5. Place a number 6 by any item that would not have been on your list 5 years ago.

6. Place a number 7 by any item you think will not be on your list 5 years from now.

7. Finally, go down through your list and indicate the date when you last did each item. Use an 8 for things you have done today, a 9 for those done within the last month, and a 10 for things you haven't done in more than a month.

1.							
2.							
3.							
4.							
5.							
6.							
7.							
8.							
9.							
10.							
11.							
12.							
13.							
14.							
15.							
16.							
17.							
18.							
19.							
20.							

Now . . . how about writing a program that will analyze the responses that you and your class have made.

THIS IS A COMPUTERIZED ANALYSIS OF YOUR RESPONSES TO THE VALUE STRATEGY GAME '20 THINGS I LOVE TO DO'.

THIS ANALYSIS WILL PASS NO FINAL JUDGMENTS ON YOUR RESPONSES. IT WILL, HOWEVER, COMPILE THE NECESSARY STATISTICS THAT WILL AID YOU IN MAKING YOUR OWN EVALUATION OF THE THINGS YOU VALUE IN YOUR LIFE.

HOW MANY THINGS ARE CONTAINED IN YOUR LIST (FROM 1 TO 20)? 20

ENTER THE CODE NUMBERS THAT YOU HAVE PLACED NEXT TO ITEM# 1 . TYPE A 99 AFTER ALL OF YOUR NUMBERS HAVE BEEN ENTERED.

? 5
? 10
? 99

ENTER THE CODE NUMBERS THAT YOU HAVE PLACED NEXT TO ITEM# 2 . TYPE A 99 AFTER ALL YOUR NUMBERS HAVE BEEN ENTERED.

? 1
? 4
? 10
? 99
:
:

YOUR COMPUTERIZED ANALYSIS . . . . .

YOU ARE FORTUNATE. IN THESE INFLATIONARY DAYS ONLY 30 PERCENT OF THE THINGS YOU LOVE REQUIRE MONEY.

YOU MUST BE CAUTIOUS!
ONLY 15 PERCENT OF YOUR VALUED ACTIVITIES

INVOLVE PERSONAL RISK.

YOU AND YOUR PARENTS VALUE LESS THAN HALF OF THE SAME THINGS.

PEOPLE ARE INVOLVED IN LESS THAN 50 PERCENT OF YOUR FAVORITE ACTIVITIES.

YOU ENJOY DOING THINGS THAT INVOLVE YOU ALONE MOST OF THE TIME.

2 OUT OF 10 TIMES YOUR ITEMS WOULD NOT HAVE APPEARED ON YOUR LIST FIVE YEARS AGO.

YOU KNOW WHAT YOU VALUE. 0 PERCENT OF YOUR ITEMS WILL NOT BE ON YOUR LIST IN 1978.

35 PERCENT OF YOUR VALUED ACTIVITIES HAVE BEEN ENJOYED WITHIN THE LAST 24 HOURS.

WITHIN THE LAST MONTH YOU HAVE DONE 60 PERCENT OF THE THINGS YOU LOVE TO DO.

AND IN 4 OUT OF 10 CASES YOU HAVE NOT BEEN INVOLVED IN YOUR VALUED ACTIVITIES FOR MORE THAN A MONTH.

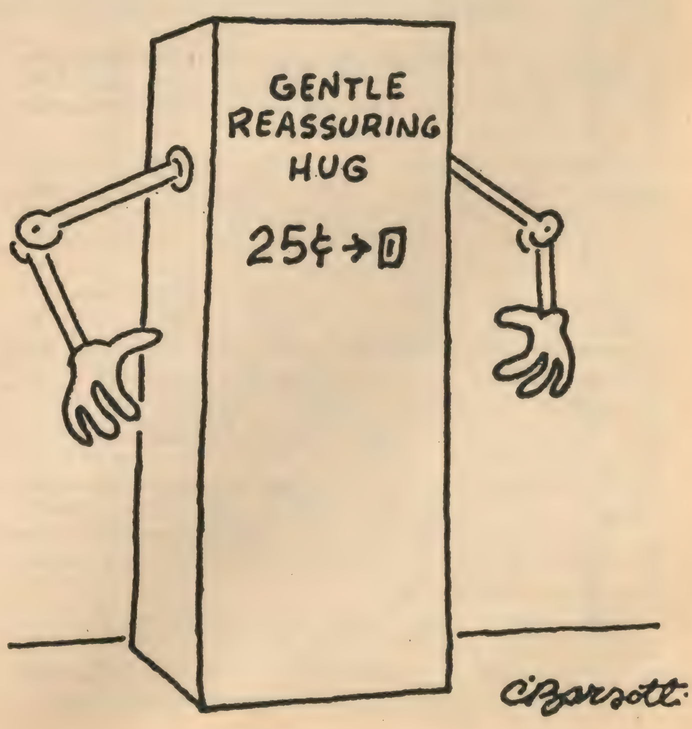
YOUR COMPUTER ANALYSIS OF THE THINGS THAT YOU VALUE IS COMPLETE. WE HOPE THAT YOU ARE HAPPY WITH THE RESULTS. PERHAPS THERE ARE SOME CHANGES YOU WOULD LIKE TO MAKE OR PERHAPS YOU HAVE LEARNED SOMETHING ABOUT YOURSELF BY PLAYING '20 THINGS'. ONLY YOU CAN EVALUATE THE STATISTICS OF THIS ANALYSIS.
IT MIGHT BE INTERESTING TO GO BACK OVER YOUR LIST AND ASSIGN PRIORITIES TO YOUR ITEMS. IT MIGHT GIVE YOU SOME ADDITIONAL INSIGHT TOWARD THE THINGS THAT YOU VALUE.

After you have written your analysis program look back over the results.

- Can you identify any patterns in the things that you value?
- Did you learn something new about yourself?
- Are there things you are pleased with?
- Is there anything you would change?
- How might you go about it?
- Are there some things you like to do that you haven't done lately? Why? What can you do about it?
- Does establishing priorities for your entries give you additional insight about you and your values?

Try playing "20 Things" several times this year. Maybe you'll learn something new about yourself each time!

This article was adapted from "What do You Value" by Sid Simon that appeared in the Spring-Summer issue of *Forum*, a semi-annual publication for educators by J. C. Penney Co. Ask for an issue at your local Penney's store. This article was reprinted with permission from *EDU* No. 7 published by Digital Equipment Corp.



Building A Computer of Your Own

by Stephen B. Gray

During the last year, getting a computer of your own has become much easier than before. There are three companies offering microcomputer kits, and the prices of commercial minicomputers are coming down with every new model.

Not long ago the only way to obtain a computer without paying several thousand dollars was to buy an old vacuum-tube model, or to go the very difficult route of building a transistor one from scratch. Even now there is an occasional vacuum-tube machine available, but the drawbacks are formidable: many are so large they require a large barn to store, they need a great deal of air-conditioning and electrical power, and some tubes can be very expensive to replace. Schematics are needed to get the computer working and maintained, but they are almost never available. Even with some of the older transistor computers, updated schematics are usually impossible to obtain. Now and then the prototype of a recent transistor computer can be bought cheaply, but again, usually without schematics, so the buyer has two choices: take months or years to trace out every connection, or rewire most or all of the machine.

As in amateur radio, many computerniks would never think of buying a ready-built machine; they feel compelled to build one. Up until quite recently, this task has proved to be so difficult that only a couple of dozen computer hobbyists in the country had operating digital computers of any real complexity, and nearly all of them were electronics engineers in the computer industry.

The problem in building a computer from scratch is that so many areas of specialization are involved: logic, input/output, memory, peripherals, and mechanical skills such as packaging, back-plane wiring, metal-working, plastics, and many others.

Although many of the computer hobbyists are engineers who design their own circuits, most non-engineers must rely on published information, and although several dozen books and manuals contain computer schematics, they have serious limitations. A book may show schematics of various portions of a computer – arithmetic unit, memory, control circuits – but none show how to connect them together, and anyway, they are usually only partial schematics. Minicomputer manuals containing schematics can be bought, but many of the parts are identified only by a manufacturer's code number.

Even supposing an amateur computer-builder did get hold of complete schematics and all the parts, the one big stumbling block that has thrown many is core memory. It's still expensive to buy when new, and when surplus, it may contain broken cores, or perhaps it became surplus because it couldn't pass the manufacturer's quality control. Getting a core memory to work still separates the men from the boys, if there are still any who want to try it, now that semiconductor memory has become so readily available and cheap.

Surplus computer PC boards have been available for some years, but nearly all of them are without the "tab" ends, broken off to make sure the boards won't find their way back into commercial computers.

Even after the advent of the 54/74 series of integrated digital circuits and the various CPU-on-a-chip microcomputer circuits, there was still a dearth of information on just how to build a computer from what was offered.

In 1972, Intel introduced two sets of chips for microprocessors, the MCS-4 and MCS-8. The chips themselves weren't too expensive, in relation to what had been available before, but the cost of having the read-only memories programmed was several times that of the set of chips.

The CPU chip in the MCS-8 set is the 8008, which is the basis of two computer-hobbyist kits currently available. The SCELBI-8H, first offered in late 1973, is available in a variety of ways: as an assembled and tested computer with a 4K memory, at $1239, or in kit form for $1149; as a set of five printed-circuit cards with a 1K memory, $498; individual cards, from $55 to $195; "unpopulated" cards (without components), a set of five for $109; and various other combinations. Several interface cards are available, for making use of an oscilloscope readout, audio cassette-tape memory, or Teletype. Two dozen programs are available, including keyboard-to-CRT display, assembler, Teletype memory dump, magnetic-tape bootstrap loader, etc.

Incidentally, for those who have learned FORTRAN or BASIC in school or elsewhere, having to program at the assembly-language level can be very tedious, uninspiring, and error-prone.

The July 1974 *Radio-Electronics* described the Mark-8, also built around the Intel 8008 microprocessor, and also programmed in the Intel assembly language. A minimum Mark-8, with 256 8-bit words, is about $300. The construction manual for the Mark-8, which also gives information on obtaining a set of PC boards, is $5.00 from *Radio-Electronics*.

The Altair 8800 (*Popular Electronics,* Jan. and Feb. 1975) is based on the Intel 8080 chip, faster and with more instructions than the 8008, and is sold by MITS for $542 with 256 words of memory; with 1K words, $701 in kit form, or $938 assembled. Larger systems are also available, including the Basic I, with 8K words of memory, audio-cassette interface, serial input/output card, a computer terminal with keyboard and 32-character display, and BASIC software, for $2393 in kit form. Peripherals include a disc drive ($1480 kit) and a line printer.

Other kits and peripherals are available elsewhere; a recent issue of *Radio-Electronics* contained two construction articles, on a CRT terminal and an interface for connecting the terminal to a Teletype or cassette recorder, ads for the SCELBI-8H, MITS Altair 8800, and the CRT terminal, plus three ads for microcomputer kits (and two for semiconductor memories) from surplus-parts companies.

A series of publications on the "Experimenter's Computer System" is offered by M. P. Publishing, including one for $2.50 on a microcomputer CPU based on the Intel 8008, with 256 8-bit words; another for $2.50 on an audio-cassette mass-storage system, with schematics; a third about an I/O controller, etc. These were available on a subscription basis; similar material now appears in the ECS monthly magazine; the first issues go heavily into assembly-language programs.

There are several newsletters of interest to the computer hobbyist. The Amateur Computer Society has published a 6-page newsletter about every three months since 1966, with information about computers that ACS members have built, relevant books and magazine articles, where to buy parts, parts for sale or trade by members, and information on kits and other commercially available items.

"The Computer Hobbyist" newsletter was founded in 1974, and leans toward circuits and assembly-language programs. The first issue had Part One of a long article on "A Graphics Display for the 8008" with two programs, a Surplus Summary, and Notes on the 8008 Instruction Set.

The Micro-8 Computer Users Group was originally the Mark-8 Group, but widened its horizons when several more micros appeared; an extensive newsletter is published, with information on what its members have achieved with micros, interfaces and peripherals, comments on kits, various circuits, etc.

As for the future, minicomputers have been getting smaller and cheaper, and calculators more complex and cheaper, so it may be only a matter of time before a hand-held computer is available for a few hundred dollars.

[Ed Note: I feel I should share with you the last paragraph of Steve's cover letter for this article. "Incidentally, I almost added a paragraph at the very end, on 'What are you going to do with your computer?' but I felt this would make a good subject for another article, where it can be examined at length, on the trivial uses that home computers are put to, simply because there isn't much real need for one, other than for fun and games." Readers: let's hear from you on this —DHA]

Amateur Computer Society
260 Noroton Ave.
Darien, Conn. 06820

The Computer Hobbyist
Box 295
Cary, North Carolina 27511

Micro-8 Computer Users Group
Cabrillo Computer Center
4350 Constellation
Lompoc, Calif. 93436

M. P. Publishing Co.
Box 378
Belmont, Mass. 02178

MITS
6328 Linn, N.E.
Albuquerque, New Mexico 87108

Radio-Electronics
200 Park Ave. South
New York, N. Y. 10003

SCELBI Computer Consulting, Inc.
1322 Rear, Boston Post Road
Milford, Conn. 06460

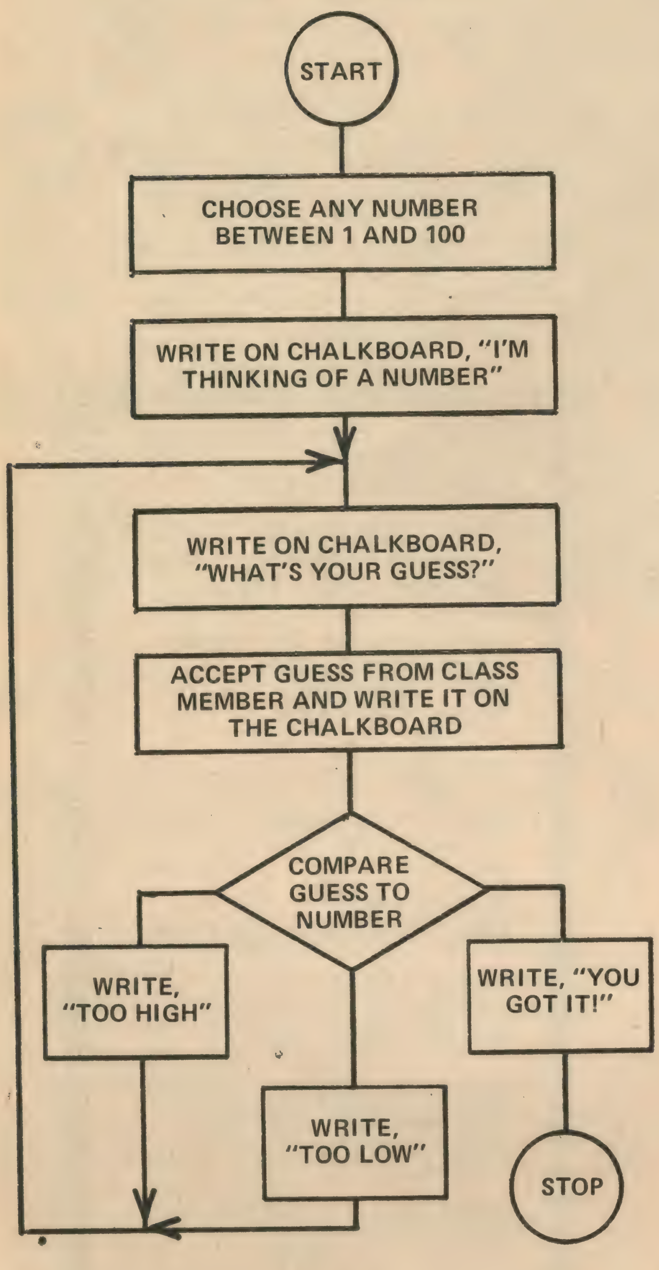
Sphere (computers and kits)
96 East 500 South
Bountiful, Utah 84010

? ? ? ¿ ? ? GUESS ? ? ? ? ¿ ?

GUESS is a simple introduction to the idea of a computer program and computer games. It also introduces the notions of comparison and searching a list.

Before playing GUESS on the computer, you may wish to play it in class at the chalkboard. One student is selected to be the computer. He is given the flowchart below and is asked to follow it and write on the chalkboard the statements in parentheses as well as each guess. Students take turns being the computer.

FLOWCHART



After playing the game manually, play GUESS on the computer. Divide the class into small teams of 2 or 3 members each. Have each team play 10 games on the computer and keep track of the number of guesses required to get each number. Compute the average guesses for each team and for the entire class. After playing the class should consider the following questions:

1. Why should it never take more than 7 guesses to find the number?

2. What was the average guesses for the class? How does this compare to the theoretical average (using the correct guessing strategy) of 5.3?

3. What are the maximum number of guesses required to find a mystery number between:

1 and 10	1 and 64
1 and 63	1 and 1000

PROGRAM LISTING

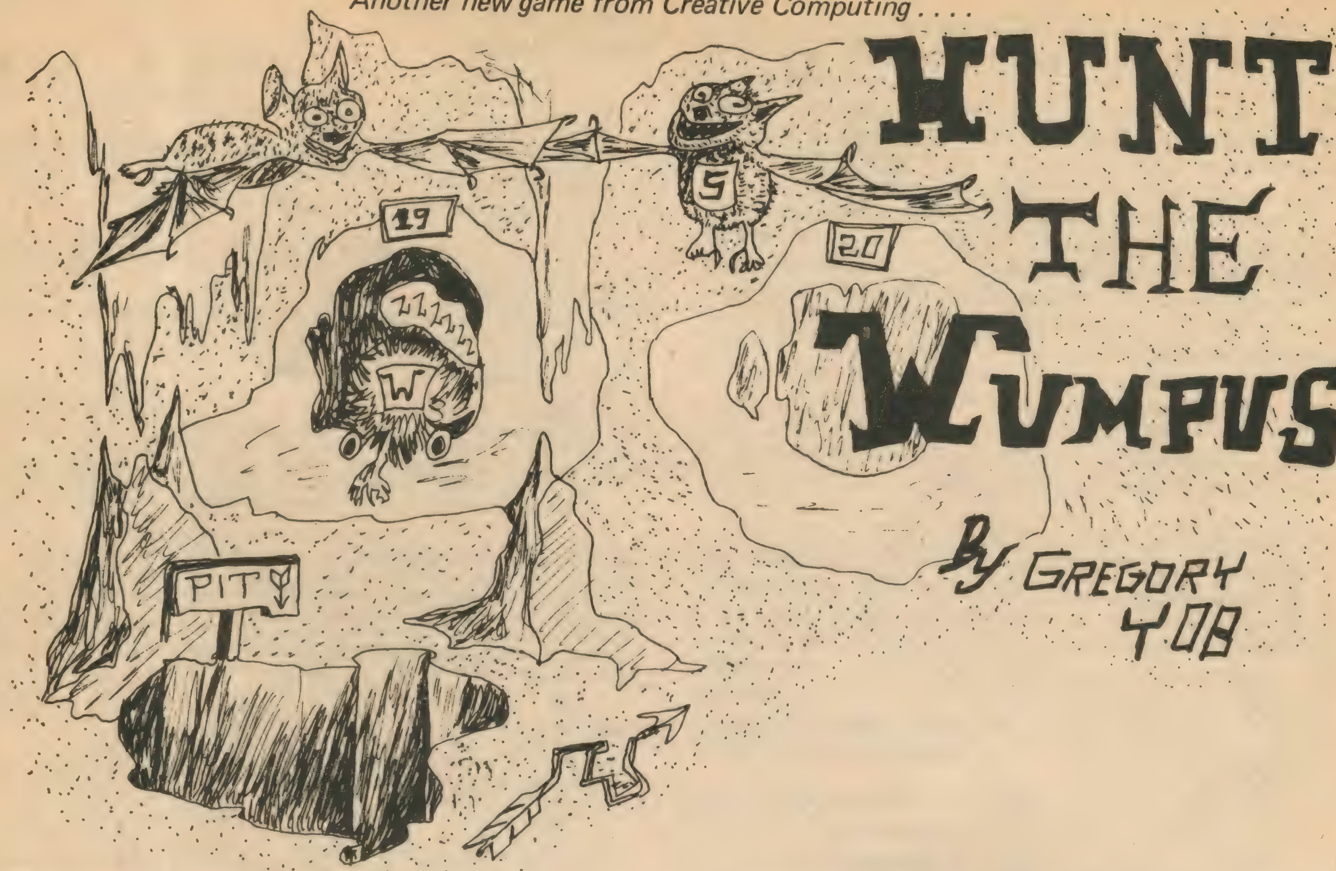
```
 1 PRI "THIS IS A NUMBER GUESSING GAME.  I'LL THINK"
 2 PRI "OF A NUMBER BETWEEN 1 AND ANY LIMIT YOU WANT."
 3 PRI "THEN YOU HAVE TO GUESS WHAT IT IS."
 4 PRI
 5 PRI "WHAT LIMIT DO YOU WANT";
 6 INPL
 7 PRI
 8 L1=INT(LOG(L)/LOG(2))+1
10 PRI"I'M THINKING OF A NUMBER BETWEEN 1 AND"L
11 G=1
14 PRI "NOW YOU TRY TO GUESS WHAT IT IS"
15 M=INT(L*RND(0))+1
20 INP N
21 IF N>0 THEN 25
22 GOSUB70
23 GOTO1
25 IF N=M THEN 50
30 G=G+1
31 IF N>M THEN 40
32 PRI "TOO LOW.  GUESS AGAIN."
33 GOTO 20
40 PRI "TOO HIGH.   GUESS AGAIN."
42 GOTO 20
50 PRI "THAT'S IT!  YOU GOT IT IN"G"TRIES.
52 IF G<L1 THEN 58
54 IF G=L1 THEN 60
56 PRI"YOU SHOULD HAVE BEEN ABLE TO GET IT IN ONLY"L1".
57 GOT 65
58 PRI"VERY ";
60 PRI"GOOD!
65 GOSUB70
66 GOTO10
70 FOR H=1 TO 5
71 PRI
72 NEXT H
73 RETURN
99 END
```

Note: BASIC statements in this program are abbreviated to their first 3 letters.

SAMPLE RUN

```
THIS IS A NUMBER GUESSING GAME.  I'LL THINK
OF A NUMBER BETWEEN 1 AND ANY LIMIT YOU WANT.
THEN YOU HAVE TO GUESS WHAT IT IS.

WHAT LIMIT DO YOU WANT? 100

I'M THINKING OF A NUMBER BETWEEN 1 AND 100
NOW YOU TRY TO GUESS WHAT IT IS
? 50
TOO HIGH.   GUESS AGAIN.
? 25
THAT'S IT!  YOU GOT IT IN 2 TRIES.
VERY GOOD!




I'M THINKING OF A NUMBER BETWEEN 1 AND 100
NOW YOU TRY TO GUESS WHAT IT IS
? 50
TOO HIGH.   GUESS AGAIN.
? 25
TOO LOW.  GUESS AGAIN
? 37
TOO HIGH.   GUESS AGAIN.
? 31
TOO HIGH.   GUESS AGAIN.
? 28
TOO LOW.  GUESS AGAIN.
? 29
TOO LOW.  GUESS AGAIN.
? 30
THAT'S IT!  YOU GOT IT IN 7 TRIES
GOOD!
```

Another new game from Creative Computing . . . .



The Genesis of Wumpus

Two years ago I happened by People's Computer Company (PCC) and saw some of their computer games — such as Hurkle, Snark, and Mugwump. My reaction was: "EECH!!" Each of these games was based on a 10 x 10 grid in Cartesian co-ordinates and three of them was too much for me. I started to think along the lines of: "There has to be a hide and seek computer game without that (exp. deleted) grid!!" In fact, why not a topological computer game — Imagine a set of points connected in some way and the player moves about the set via the interconnections.

That afternoon in meditation the phrase "Hunt the Wumpus" arrived, and Wumpus was born. He's still a bit vague in physical detail as most dedicated Wumpus hunters know, but appearances are part of the game. (If you like, send me a picture of your version of a Wumpus. Perhaps friendly Dave, our editor, will publish the best one in *Creative Computing*.) The grid I chose was the vertices of a dodecahedron — simply because it's my favorite Platonic solid and once, ages ago, I made a kite shaped like one. The edges became the connecting tunnels between the caves which were the set of points for the game.

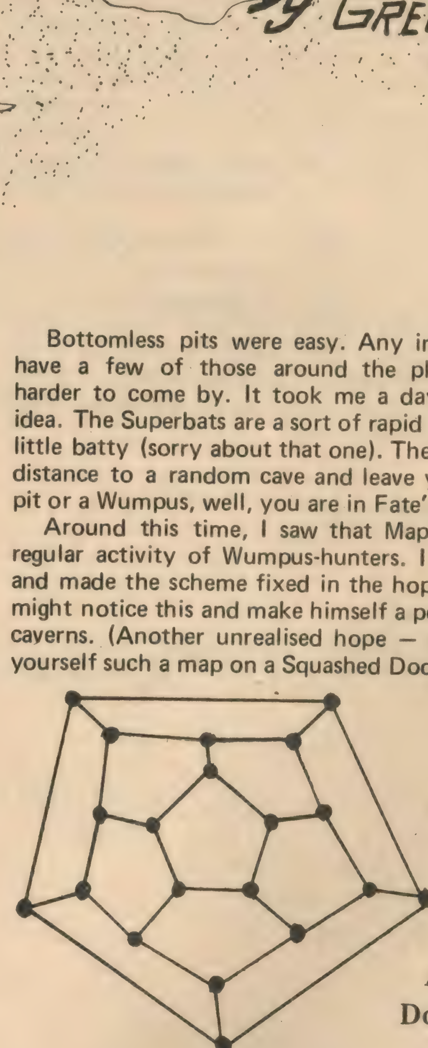
My basic idea at this time was for the player to approach the Wumpus, back off, and come up to him by going around the dodecahedron. To my knowledge, this has never happened . . . most players adopt other strategies rather than this cold-blooded approach.

Anyway . . . how to get the Wumpus! How about an arrow which could turn corners as it goes from room to room. Let the hunter tell the arrow where to go and let it fly. The shortest round trip without reversals is 5 caves — and thus the Crooked Arrow.

Hmmm . . . How does one sense the Wumpus? It's dark in yonder cave, and light would wake him up. If one got one cave away, the wumpus's distinct smell would serve as a warning. So far, so good . . . but Wumpus is still too easy, so let's find some appropriate hazards for the caves.

Bottomless pits were easy. Any imaginary cave would have a few of those around the place. Superbats were harder to come by. It took me a day or two to get that idea. The Superbats are a sort of rapid transit system gone a little batty (sorry about that one). They take you a random distance to a random cave and leave you there. If that's a pit or a Wumpus, well, you are in Fate's hands.

Around this time, I saw that Map-making would be a regular activity of Wumpus-hunters. I numbered the caves and made the scheme fixed in the hopes a practised player might notice this and make himself a permanent map of the caverns. (Another unrealised hope — as an exercise, make yourself such a map on a Squashed Dodecahedron).



A Squashed Dodecahedron

To start the game fairly, Wumpus, Hazards, and Hunter are located on different points at the start of the game. Each game starts with random choices of location, but the hunter may restart with the same set-up if he chooses. This allows re-plays if the hunter, say, fell into a pit on the first move.

Wumpus was nearly done in my mind . . . (hint to a games-writer: Have a clear notion of your game before you start coding it. This saves MUCH confusion.) yet I felt it was a bit dull. Once you found the Wumpus all you had to do was shoot it. To fix this, the Wumpus was given a little life. If you shot an arrow or moved into his cave, he woke up and chose to move to a neighboring room or to the same room (one of 4 choices). If you and the Wumpus were in the same room after he moved, he ATE YOU UP!!

Around here I noticed that the pits and the bats didn't affect the Wumpus. To explain this, I added some color by making him heavy and with the legendary sucker feet. After all, evolution works in strange ways!! If you are a Wumpus fiend, make a version of Wumpus in which he avoids pits and superbats can carry him only one room (with the possibility of being dumped into your cave). This can be done by making the wumpus moving procedure a subroutine.

I wrote Wumpus and dropped it off at PCC. Then I went home and dreamed up Wumpus II which will be covered in the next issue of *Creative Computing.*

The Birth of Wumpus

Around a month later, I went to the Synergy conference at Stanford, where many of the far-out folk were gathered to share their visions of improving the world. PCC had a few terminals running in a conference room and I dropped by. To my vast surprise, all of the terminals were running Wumpus and scraps of paper on the floor with scrawled numbers and lines testified that much dedicated Wumpus-hunting was in progress. I had spawned a hit computer game!!!

Later, PCC published Wumpus in its newsletter (If you haven't seen it, write them for a subscription: P.O. Box 310, Menlo Park, Cal. 94025), and Wumpus appeared in all sorts of unlikely places. I have reports of Wumpus written in RPG, a listing of one in FORTRAN, a rumor of a system command of 'to Wumpus' on a large corporation's R&D computer system and have even seen an illustrated version for the Hazeltine CRT terminal!!

WUMPUS TAPES, ETC.

I can be found at:

Gregory Yob
PO Box 354
Palo Alto, Calif. 94301

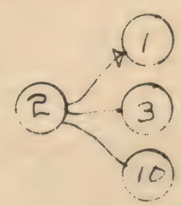
Paper tapes of Wumpus, Wumpus 2 and Wumpus 3 are available and cost $5.00 each.

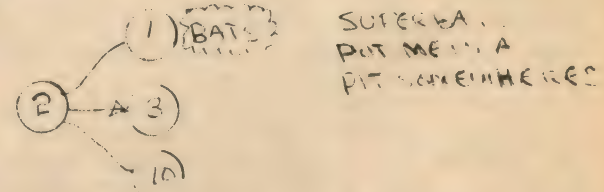
May your arrows remain straight. —Gregory Yob.

SAMPLE RUN

```
INSTRUCTIONS (Y-N)?Y
WELCOME TO 'HUNT THE WUMPUS'
  THE WUMPUS LIVES IN A CAVE OF 20 ROOMS. EACH ROOM
HAS 3 TUNNELS LEADING TO OTHER ROOMS. (LOOK AT A
DODECAHEDRON TO SEE HOW THIS WORKS-IF YOU DON'T KNOW
WHAT A DODECAHEDRON IS, ASK SOMEONE)

    HAZARDS:
 BOTTOMLESS PITS - TWO ROOMS HAVE BOTTOMLESS PITS IN THEM
     IF YOU GO THERE, YOU FALL INTO THE PIT (& LOSE!)
 SUPER BATS - TWO OTHER ROOMS HAVE SUPER BATS. IF YOU
     GO THERE, A BAT GRABS YOU AND TAKES YOU TO SOME OTHER
     ROOM AT RANDOM. (WHICH MIGHT BE TROUBLESOME)

    WUMPUS:
 THE WUMPUS IS NOT BOTHERED BY THE HAZARDS (HE HAS SUCKER
 FEET AND IS TOO BIG FOR A BAT TO LIFT).  USUALLY
 HE IS ASLEEP. TWO THINGS WAKE HIM UP: YOUR ENTERING
 HIS ROOM OR YOUR SHOOTING AN ARROW.
     IF THE WUMPUS WAKES, HE MOVES (P=.75) ONE ROOM
 OR STAYS STILL (P=.25). AFTER THAT, IF HE IS WHERE YOU
 ARE, HE EATS YOU UP (& YOU LOSE!)

    YOU:
 EACH TURN YOU MAY MOVE OR SHOOT A CROOKED ARROW
   MOVING: YOU CAN GO ONE ROOM (THRU ONE TUNNEL)
   ARROWS: YOU HAVE 5 ARROWS. YOU LOSE WHEN YOU RUN OUT.
   EACH ARROW CAN GO FROM 1 TO 5 ROOMS. YOU AIM BY TELLING
   THE COMPUTER THE ROOM#S YOU WANT THE ARROW TO GO TO.
   IF THE ARROW CAN'T GO THAT WAY(IE NO TUNNEL) IT MOVES
   AT RANDOM TO THE NEXT ROOM.
     IF THE ARROW HITS THE WUMPUS, YOU WIN.
     IF THE ARROW HITS YOU, YOU LOSE.

    WARNINGS:
     WHEN YOU ARE ONE ROOM AWAY FROM WUMPUS OR HAZARD,
    THE COMPUTER SAYS:
 WUMPUS-  'I SMELL A WUMPUS'
 BAT   -  'BATS NEARBY'
 PIT   -  'I FEEL A DRAFT'
```

```
HUNT THE WUMPUS


BATS NEARBY!
YOU ARE IN ROOM   2
TUNNELS LEAD TO   1        3         10

SHOOT OR MOVE (S-M)?M
WHERE TO?1
ZAP--SUPER BAT SNATCH! ELSEWHEREVILLE FOR YOU!
YYYIIIIEEEE . . . FELL IN PIT
HA HA HA - YOU LOSE!
SAME SET-UP (Y-N)?Y
HUNT THE WUMPUS


BATS NEARBY!
YOU ARE IN ROOM   2
TUNNELS LEAD TO   1        3         10

SHOOT OR MOVE (S-M)?M
WHERE TO?3

YOU ARE IN ROOM   3
TUNNELS LEAD TO   2        4         12
```

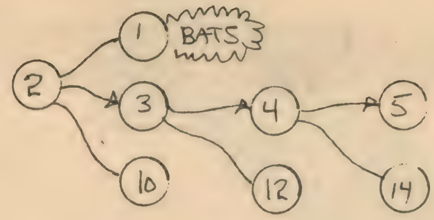




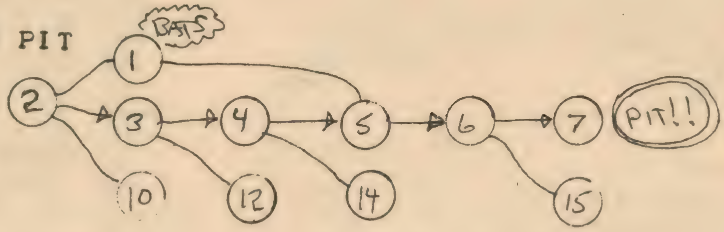
more

```
SHOOT OR MOVE (S-M)?M
WHERE TO?4

YOU ARE IN ROOM  4
TUNNELS LEAD TO  3      5      14

SHOOT OR MOVE (S-M)?M
WHERE TO?5

BATS NEARBY!
YOU ARE IN ROOM  5
TUNNELS LEAD TO  1      4      6

SHOOT OR MOVE (S-M)?M
WHERE TO?6

I FEEL A DRAFT
YOU ARE IN ROOM  6
TUNNELS LEAD TO  5      7      15
```

JUST KEEP ON TRUCKIN!

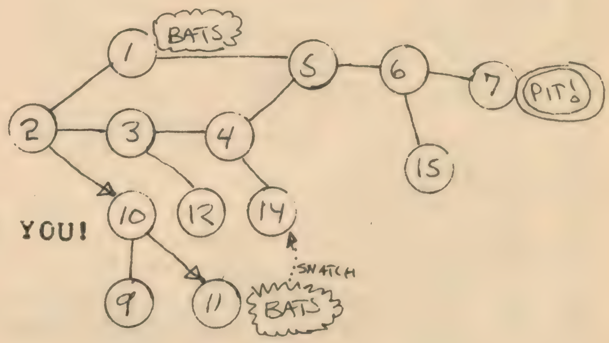



```
SHOOT OR MOVE (S-M)?M
WHERE TO?7
YYYIIIIEEEE . . . FELL IN PIT
HA HA HA - YOU LOSE!
SAME SET-UP (Y-N)?Y
HUNT THE WUMPUS
```

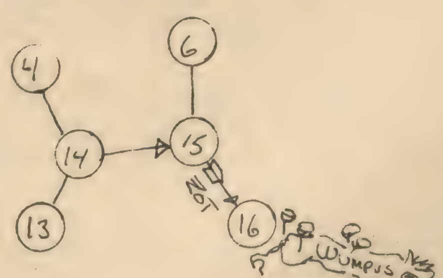


BLEW IT AGAIN!!

```
BATS NEARBY!
YOU ARE IN ROOM  2
TUNNELS LEAD TO  1      3      10

SHOOT OR MOVE (S-M)?M
WHERE TO?10

BATS NEARBY!
YOU ARE IN ROOM  10
TUNNELS LEAD TO  2      9      11

SHOOT OR MOVE (S-M)?M
WHERE TO?11
ZAP--SUPER BAT SNATCH! ELSEWHEREVILLE FOR YOU!

YOU ARE IN ROOM  14
TUNNELS LEAD TO  4      13     15

SHOOT OR MOVE (S-M)?M
WHERE TO?15
```




```
I SMELL A WUMPUS!
YOU ARE IN ROOM  15
TUNNELS LEAD TO  6      14     16

SHOOT OR MOVE (S-M)?S
NO. OF ROOMS(1-5)?1
ROOM #?16
AHA! YOU GOT THE WUMPUS!
HEE HEE HEE - THE WUMPUS'LL GETCHA NEXT TIME!!
```



CAN YOU FIT THIS MAP INTO THE OTHER ONE ABOVE? FIGURE OUT HOW I KNEW THE WUMPUS WAS IN 16.

PROGRAM LISTING

```
0010  REM- HUNT THE WUMPUS
0015  REM:  BY GREGORY YOB
0020  PRINT "INSTRUCTIONS (Y-N)";
0030  INPUT I$
0040  IF I$="N" THEN 52
0050  GOSUB 1000
0052  REM- ANNOUNCE WUMPUSII FOR ALL AFICIONADOS ... ADDED BY DAVE
0054  PRINT
0056  PRINT "     ATTENTION ALL WUMPUS LOVERS!!!"
0058  PRINT "     THERE ARE NOW TWO ADDITIONS TO THE WUMPUS FAMILY";
0060  PRINT " OF PROGRAMS."
0062  PRINT
0064  PRINT "     WUMP2:  SOME DIFFERENT CAVE ARRANGEMENTS"
0066  PRINT "     WUMP3:  DIFFERENT HAZARDS"
0067  PRINT
0068  REM- SET UP CAVE (DODECAHEDRAL NODE LIST)
0070  DIM S(20,3)
0080   FOR J=1 TO 20
0090    FOR K=1 TO 3
0100    READ S(J,K)
0110    NEXT K
0120   NEXT J
0130  DATA 2,5,8,1,3,10,2,4,12,3,5,14,1,4,6
0140  DATA 5,7,15,6,8,17,1,7,9,8,10,18,2,9,11
0150  DATA 10,12,19,3,11,13,12,14,20,4,13,15,6,14,16
0160  DATA 15,17,20,7,16,18,9,17,19,11,18,20,13,16,19
0170  DEF FNA(X)=INT(20*RND(0))+1
0180  DEF FNB(X)=INT(3*RND(0))+1
0190  DEF FNC(X)=INT(4*RND(0))+1
0200  REM-LOCATE L ARRAY ITEMS
0210  REM-1-YOU,2-WUMPUS,3&4-PITS,5&6-BATS
0220  DIM L(6)
0230  DIM M(6)
0240   FOR J=1 TO 6
0250   L(J)=FNA(0)
0260   M(J)=L(J)
0270   NEXT J
0280  REM-CHECK FOR CROSSOVERS (IE L(1)=L(2),ETC)
0290   FOR J=1 TO 6
0300    FOR K=J TO 6
0310    IF J=K THEN 330
0320    IF L(J)=L(K) THEN 240
0330    NEXT K
0340   NEXT J
0350  REM-SET# ARROWS
0360  A=5
0365  L=L(1)
0370  REM-RUN THE GAME
0375  PRINT "HUNT THE WUMPUS"
0380  REM-HAZARD WARNINGS & LOCATION
0390  GOSUB 2000
0400  REM-MOVE OR SHOOT
0410  GOSUB 2500
0420  GOTO O OF 440,480
0430  REM-SHOOT
0440  GOSUB 3000
0450  IF F=0 THEN 390
0460  GOTO 500
0470  REM-MOVE
0480  GOSUB 4000
0490  IF F=0 THEN 390
0500  IF F>0 THEN 550
0510  REM-LOSE
0520  PRINT "HA HA HA - YOU LOSE!"
0530  GOTO 560
0540  REM-WIN
0550  PRINT "HEE HEE HEE - THE WUMPUS'LL GETCHA NEXT TIME!!"
0560   FOR J=1 TO 6
0570   L(J)=M(J)
0580   NEXT J
0590  PRINT "SAME SET-UP (Y-N)";
0600  INPUT I$
0610  IF I$#"Y" THEN 240
0620  GOTO 360
1000  REM-INSTRUCTIONS
1010  PRINT "WELCOME TO 'HUNT THE WUMPUS'"
1020  PRINT "  THE WUMPUS LIVES IN A CAVE OF 20 ROOMS. EACH ROOM"
1030  PRINT "HAS 3 TUNNELS LEADING TO OTHER ROOMS. (LOOK AT A"
1040  PRINT "DODECAHEDRON TO SEE HOW THIS WORKS-IF YOU DON'T KNOW"
1050  PRINT "WHAT A DODECAHEDRON IS, ASK SOMEONE)"
1060  PRINT
1070  PRINT "     HAZARDS:"
1080  PRINT " BOTTOMLESS PITS - TWO ROOMS HAVE BOTTOMLESS PITS IN THEM"
1090  PRINT "     IF YOU GO THERE, YOU FALL INTO THE PIT (& LOSE!)"
1100  PRINT " SUPER BATS - TWO OTHER ROOMS HAVE SUPER BATS. IF YOU"
1110  PRINT "     GO THERE, A BAT GRABS YOU AND TAKES YOU TO SOME OTHER"
1120  PRINT "     ROOM AT RANDOM. (WHICH MIGHT BE TROUBLESOME)"
1130  PRINT
1140  PRINT "     WUMPUS:"
1150  PRINT " THE WUMPUS IS NOT BOTHERED BY THE HAZARDS (HE HAS SUCKER"
1160  PRINT " FEET AND IS TOO BIG FOR A BAT TO LIFT).  USUALLY"
1170  PRINT " HE IS ASLEEP. TWO THINGS WAKE HIM UP: YOUR ENTERING"
1180  PRINT " HIS ROOM OR YOUR SHOOTING AN ARROW."
1190  PRINT "     IF THE WUMPUS WAKES, HE MOVES (P=.75) ONE ROOM"
1200  PRINT " OR STAYS STILL (P=.25). AFTER THAT, IF HE IS WHERE YOU"
1210  PRINT " ARE, HE EATS YOU UP (& YOU LOSE!)"
1220  PRINT
1230  PRINT "     YOU:"
1240  PRINT " EACH TURN YOU MAY MOVE OR SHOOT A CROOKED ARROW"
1250  PRINT "   MOVING: YOU CAN GO ONE ROOM (THRU ONE TUNNEL)"
1260  PRINT "   ARROWS: YOU HAVE 5 ARROWS. YOU LOSE WHEN YOU RUN OUT."
1270  PRINT "   EACH ARROW CAN GO FROM 1 TO 5 ROOMS. YOU AIM BY TELLING"
1280  PRINT "   THE COMPUTER THE ROOM#S YOU WANT THE ARROW TO GO TO."
1290  PRINT "   IF THE ARROG CAN'T GO DHAT WAY(IE NO TUNNEL) IT MOFES"
1300  PRINT "   AT RANDOM TO THE NEXT ROOM."
1310  PRINT "     IF THE ARROW HITS THE WUMPUS, YOU WIN."
1320  PRINT "     IF THE ARROW HITS YOU, YOU LOSE."
1330  PRINT
1340  PRINT "    WARNINGS:"
1350  PRINT "     WHEN YOU ARE ONE ROOM AWAY FROM WUMPUS OR HAZARD,"
1360  PRINT "    THE COMPUTER SAYS:"
1370  PRINT " WUMPUS-  'I SMELL A WUMPUS'"
1380  PRINT " BAT   -  'BATS NEARBY'"
1390  PRINT " PIT   -  'I FEEL A DRAFT'"
1400  PRINT ""
1410  RETURN
2000  REM-PRINT LOCATION & HAZARD WARNINGS
2010  PRINT
2020   FOR J=2 TO 6
2030    FOR K=1 TO 3
2040    IF S(L(1),K)#L(J) THEN 2110
2050    GOTO J-1 OF 2060,2080,2080,2100,2100
2060    PRINT "I SMELL A WUMPUS!"
2070    GOTO 2110
2080    PRINT "I FEEL A DRAFT"
2090    GOTO 2110
2100    PRINT "BATS NEARBY!"
2110    NEXT K
2120   NEXT J
2130  PRINT "YOU ARE IN ROOM "L(1)
2140  PRINT "TUNNELS LEAD TO "S(L,1);S(L,2);S(L,3)
2150  PRINT
2160  RETURN
2500  REM-CHOOSE OPTION
2510  PRINT "SHOOT OR MOVE (S-M)";
2520  INPUT I$
2530  IF I$#"S" THEN 2560
2540  O=1
2550  RETURN
2560  IF I$#"M" THEN 2510
2570  O=2
2580  RETURN
3000  REM-ARROW ROUTINE
3010  F=0
3020  REM-PATH OF ARROW
3030  DIM P(5)
3040  PRINT "NO. OF ROOMS(1-5)";
3050  INPUT J9
3060  IF J9<1 OR J9>5 THEN 3040
3070   FOR K=1 TO J9
3080   PRINT "ROOM #";
3090   INPUT P(K)
3095   IF K <= 2 THEN 3115
3100   IF P(K) <> P(K-2) THEN 3115
3105   PRINT "ARROWS AREN'T THAT CROOKED - TRY ANOTHER ROOM"
3110   GOTO 3080
3115   NEXT K
3120  REM-SHOOT ARROW
3130  L=L(1)
3140   FOR K=1 TO J9
3150    FOR K1=1 TO 3
3160    IF S(L,K1)=P(K) THEN 3295
3170    NEXT K1
3180   REM-NO TUNNEL FOR ARROW
3190   L=S(L,FNB(1))
3200   GOTO 3300
3210   NEXT K
3220  PRINT "MISSED"
3225  L=L(1)
3230  REM-MOVE WUMPUS
3240  GOSUB 3370
3250  REM-AMMO CHECK
3255  A=A-1
3260  IF A>0 THEN 3280
3270  F=-1
3280  RETURN
3290  REM-SEE IF ARROW IS AT L(1) OR L(2)
3295  L=P(K)
3300  IF L#L(2) THEN 3340
3310  PRINT "AHA! YOU GOT THE WUMPUS!"
3320  F=1
3330  RETURN
3340  IF L#L(1) THEN 3210
3350  PRINT "OUCH! ARROW GOT YOU!"
3360  GOTO 3270
3370  REM-MOVE WUMPUS ROUTINE
3380  K=FNC(0)
3390  IF K=4 THEN 3410
3400  L(2)=S(L(2),K)
3410  IF L(2)#L THEN 3440
3420  PRINT "TSK TSK TSK- WUMPUS GOT YOU!"
3430  F=-1
3440  RETURN
4000  REM- MOVE ROUTINE
4010  F=0
4020  PRINT "WHERE TO";
4030  INPUT L
4040  IF L<1 OR L>20 THEN 4020
4050   FOR K=1 TO 3
4060   REM- CHECK IF LEGAL MOVE
4070   IF S(L(1),K)=L THEN 4130
4080   NEXT K
4090  IF L=L(1) THEN 4130
4100  PRINT "NOT POSSIBLE -";
4110  GOTO 4020
4120  REM-CHECK FOR HAZARDS
4130  L(1)=L
4140  REM-WUMPUS
4150  IF L#L(2) THEN 4220
4160  PRINT "... OOPS! BUMPED A WUMPUS!"
4170  REM-MOVE WUMPUS
4180  GOSUB 3380
4190  IF F=0 THEN 4220
4200  RETURN
4210  REM-PIT
4220  IF L#L(3) AND L#L(4) THEN 4270
4230  PRINT "YYYIIIIEEEE . . . FELL IN PIT"
4240  F=-1
4250  RETURN
4260  REM-BATS
4270  IF L#L(5) AND L#L(6) THEN 4310
4280  PRINT "ZAP--SUPER BAT SNATCH! ELSEWHEREVILLE FOR YOU!"
4290  L=FNA(1)
4300  GOTO 4130
4310  RETURN
5000  END
```

Another new game from Creative Computing . . . .

CIVIL WAR

GENERAL DESCRIPTION

The CIVIL program simulates 14 battles of the Civil War. For each battle, it specifies the number of men and amount of money available to each side and the rate of inflation affecting the value of money at that time. The program also determines the state of morale for the Confederate troops and whether they are on offense or defense.

The user of the program is always the Confederate Commander. He decides how much of the available money is to be spent on food, salaries and ammunition, and the strategy to be employed.

The CIVIL simulation makes a 'win or lose' determination based on the user's decisions, calculates the casualties and desertions for each side, and compares the casualties of the simulated battle with those of the actual battle. Whether you win or lose is a function of the simulated casualties for both sides and the morale factor.

FACTOR RELATIONSHIPS

The CIVIL program was developed by L. Cram, L. Goodie, and D. Hibbard, students at Lexington High School, Lexington, Massachusetts. Their description of some of the important features of the game follows:

1. *Men* – These amounts are based on actual figures from each battle in the Civil War. The number of troops the South has can vary greatly according to how well the operator of the computer plays. The two primary factors determining the number of men are morale and the amount of success you had from the last battle. If all of your men either die or desert, and this can happen, your army will be drastically diminished for the next battle.
2. *Money* – These amounts are not based on actual figures. They are probably nowhere near the actual figures but this does not make any difference since everything is relative in the program. Money is to be spent for food, salaries for your men, and ammunition and equipment, and can be saved from one battle for another.
3. *Inflation* – The percent of inflation fluctuates with your success in the previous battles. It determines the present value of your money and is used in calculating the effectiveness of your money.
4. *Morale* – The morale factor is determined from the amount of money spent on food and salaries and varies with your success.
5. *Strategy* – There are two sets of strategies, one for offense and one for defense. The computer will tell you which situation you are in, as determined from the actual Civil War battles. The computer, playing the part of the North, makes a random guess at your strategy and compares its guess to your actual strategy. Your success will depend upon how close the computer comes in its guess. Hence, the same decisions on two different runs of the same battle may have different results.
6. *Casualties* – Casualties are based on actual figures but vary according to how well the Southern army is managed. They tend to be very high if not enough money is spent on ammunition and equipment.
7. *Desertions* – Desertions tend to be very large if the morale factor is low.
8. *Percent of Casualties* – This is a comparison of your casualties and the casualties in the actual Civil War battles. It indicates how well you are waging your part of the war.

SAMPLE RUN

DO YOU WANT DESCRIPTIONS? YES

THIS IS A CIVIL WAR SIMULATION.
TO PLAY, TYPE A RESPONSE WHEN THE COMPUTER ASKS.
REMEMBER THAT ALL FACTORS ARE INTERRELATED AND THAT YOUR RESPONSES COULD CHANGE HISTORY. FACTS AND FIGURES USED ARE BASED ON THE ACTUAL OCCURRENCE. MOST BATTLES TEND TO RESULT AS THEY DID IN THE CIVIL WAR, BUT IT ALL DEPENDS ON YOU!!

THE OBJECT OF THE GAME IS TO WIN AS MANY BATTLES AS POSSIBLE.

YOUR CHOICES FOR DEFENSIVE STRATEGY ARE:

(1) ARTILLERY ATTACK
(2) FORTIFICATION AGAINST FRONTAL ATTACK
(3) FORTIFICATION AGAINST FLANKING MANEUVERS
(4) FALLING BACK

YOUR CHOICES FOR OFFENSIVE STRATEGY ARE:

(1) ARTILLERY ATTACK
(2) FRONTAL ATTACK
(3) FLANKING MANEUVERS
(4) ENCIRCLEMENT

YOU MAY SURRENDER BY TYPING A '5' FOR YOUR STRATEGY.

YOU ARE THE CONFEDERACY. GOOD LUCK!

THIS IS THE BATTLE OF BULL RUN
JULY 21, 1861 GEN. BEAUREGARD COMMANDING THE SOUTH MET THE UNION FORCES WITH GEN. MCDOWELL IN A PREMATURE BATTLE AT BULL RUN. GEN. JACKSON HELPED PUSH BACK THE UNION ATTACK.

	CONFEDERACY	UNION
MEN	18000	18500
MONEY	$81000.	$83300.
INFLATION	25%	10%

What would happen if you spent more on ammunition and less on food and salaries?

HOW MUCH DO YOU WISH TO SPEND FOR
— FOOD...................?25000
— SALARIES...........?25000
— AMMUNITION.....?30000
(NOTE: Do not put commas in numbers.)

MORALE IS FAIR
YOU ARE ON THE DEFENSIVE

YOUR STRATEGY?4

	CONFEDERACY	UNION
CASUALTIES	1744	2596
DESERTIONS	10	46

YOUR CASUALTIES WERE 11% LESS THAN THE ACTUAL CASUALTIES AT BULL RUN

YOU WIN BULL RUN

Do you remember who actually won Bull Run?

PROGRAM LISTING

```
1 LET L=0:LET W=0:LET R1=0:LET P1=0
2 LET Q1=0:LET M3=0:LET M4=0
3 LET P2=0:LET T1=0:LET T2=0
5 REMARKABLE PROGRAM BY L. CRAM , L. GOODIE , AND D. HIBBARD
6 PRINT "DO YOU WANT DESCRIPTIONS (0=YES, 1=NO)";
7 INPUT Z
9 FOR U=1 TO 6
10 PRINT
11 NEXT U
13 IF Z=1 THEN 100
15 PRINT "THIS IS A CIVIL WAR SIMULATION."
20 PRINT "TO PLAY, TYPE A RESPONSE WHEN THE COMPUTER ASKS."
30 PRINT "REMEMBER THAT ALL FACTORS ARE INTERRELATED AND THAT YOUR"
35 PRINT "RESPONSES COULD CHANGE HISTORY. FACTS AND FIGURES USED ARE"
40 PRINT "BASED ON THE ACTUAL OCCURENCE. MOST BATTLES TEND TO RESULT"
45 PRINT "AS THEY DID IN THE CIVIL WAR, BUT IT ALL DEPENDS ON YOU!!"
50 PRINT
51 PRINT "THE OBJECT OF THE GAME IS TO WIN AS MANY BATTLES AS POSSIBLE"
52 PRINT
55 PRINT "YOUR CHOICES FOR DEFENSIVE STRATEGY ARE:"
60 PRINT "        (1) ARTILLERY ATTACK"
65 PRINT "        (2) FORTIFICATION AGAINST FRONTAL ATTACK"
70 PRINT "        (3) FORTIFICATION AGAINST FLANKING MANUEVERS"
75 PRINT "        (4) FALLING BACK"
80 PRINT "YOUR CHOICES FOR OFFENSIVE STRATEGY ARE:"
85 PRINT "        (1) ARTILLERY ATTACK"
90 PRINT "        (2) FRONTAL ATTACK"
95 PRINT "        (3) FLANKING MANUEVERS"
96 PRINT "        (4) ENCIRCLEMENT"
97 PRINT "YOU MAY SURRENDER BY TYPING A '5' FOR YOUR STRATEGY."
98 PRINT
99 PRINT "YOU ARE THE CONFEDERACY.       GOOD LUCK!"
100 READ M1,M2,C1,C2,M,A,U
101 LET I1=10+(L-W)*2
102 LET I2=10+(W-L)*2
103 LET D1=100*INT((M1*(100-I1)/2000)*(1+(R1-Q1)/(R1+1))+0.5)
104 LET D2=100*INT(M2*(100-I2)/2000+0.5)
105 LET F1=5*M1/6
106 LET A1=Z
107 FOR U=1 TO 4
108 PRINT
109 NEXT U
110 PRINT "THIS IS THE BATTLE OF ";
115 GOSUB 800
120 PRINT " ","CONFEDERACY"," UNION"
130 PRINT "MEN"," ";INT(M1*(1+(P1-T1)/(M3+1))),"  ";
131 PRINT INT(M2*(1+(P2-T2)/(M4+1)))
140 PRINT "MONEY","$";D1,"$";D2
150 PRINT "INFLATION"," ";I1+15;"%"," ";I2;"%"
160 PRINT
170 PRINT "HOW MUCH DO YOU WISH TO SPEND FOR FOOD";
180 INPUT F
185 IF F<0 THEN 750
190 PRINT "HOW MUCH DO YOU WISH TO SPEND FOR SALARIES";
200 INPUT S
205 IF S<0 THEN 750
210 PRINT "HOW MUCH DO YOU WISH TO SPEND FOR AMMUNITION";
220 INPUT B
221 IF B<0 THEN 750
222 PRINT
224 IF F+S+B<=D1 THEN 230
226 PRINT "THINK AGAIN!   YOU HAVE ONLY $" D1
228 GOTO 160
230 LET O=((2*F^2+S^2)/F1^2+1)
235 IF O<10 THEN 260
240 PRINT "MORALE IS HIGH"
250 GOTO 300
260 IF O<5 THEN 290
270 PRINT "MORALE IS FAIR"
280 GOTO 300
290 PRINT "MORALE IS POOR"
300 IF M<>3 THEN 330
310 PRINT "YOU ARE ON THE OFFENSIVE"
320 GOTO 370
330 IF M<>1 THEN 360
340 PRINT "YOU ARE ON THE DEFENSIVE"
350 GOTO 370
360 PRINT "BOTH SIDES ARE ON THE OFFENSIVE"
370 PRINT
380 PRINT "YOUR STEGY";
390 INPUT Y
391 IF Y=5 THEN 1487
392 IF ABS(Y-3)<3 THEN 395
393 PRINT "YOU JERK!  USE THE OTHER SET OF STRATEGIES!!"
394 GOTO 370
395 PRINT
400 PRINT " ","CONFEDERACY","UNION"
410 LET C5=(2*C1/5)*(1+1/(2*(ABS(INT(4*RND(1)+1)-Y)+1)))
412 LET C5=INT(C5*(1+1/O)*(1.28+F1/(B+1))+0.5)
414 IF C5+100/O<M1*(1+(P1-T1)/(M3+1)) THEN 424
416 LET C5=INT(13*M1/20*(1+(P1-T1)/(M3+1)))
418 LET E=7*C5/13
420 LET U=1
422 GOTO 426
424 LET E=100/O
426 PRINT "CASUALTIES",C5,INT(17*C2*C1/(C5*20)+0.5)
430 PRINT "DESERTIONS",INT(E),INT(5*O)
432 PRINT
433 IF C5-C1>=0 THEN 439
435 PRINT "YOUR CASUALTIES WERE"INT(100*(C1-C5)/C1+0.5);"% LESS THAN"
437 GOTO 441
439 PRINT "YOUR CASUALTIES WERE"INT(100*(C5-C1)/C1+0.5);"% MORE THAN"
441 PRINT "THE ACTUAL CASUALITIES AT ";
443 LET A1=1
445 GO SUB800
450 IF U=1 THEN 470
460 IF C5+E<17*C2*C1/(C5*20)+5*O THEN 490
470 PRINT "YOU LOSE ";
471 LET L=L+1
480 GOTO 555
490 PRINT "YOU WIN ";
491 LET W=W+1
555 GOSUB 800
556 IF W=8 THEN 1490
580 LET T1=T1+C5+E
590 LET T2=T2+17*C2*C1/(C5*20)+5*O
600 LET P1=P1+C1
610 LET P2=P2+C2
620 LET Q1=Q1+(F+S+B)
630 LET R1=R1+M1*(100-I1)/20
635 LETM3=M3+M1
637 LET M4=M4+M2
650 IF A=14 THEN 1500
660 GOTO 100
670 DATA 18000,18500,1967,2708,1,1,0
672 DATA 40000,44894,10699,13047,3,2,0
674 DATA 95000,115000,20614,15849,3,3,0
676 DATA 54000,63000,10000,14000,2,4,0
678 DATA 40000,50000,10000,12000,3,5,0
680 DATA 75000,120000,5377,12653,1,6,0
682 DATA 38000,45000,11000,12000,1,7,0
684 DATA 32000,90000,13000,17197,2,8,0
686 DATA 50000,70000,12000,19000,1,9,0
688 DATA 72500,85000,20000,23000,3,10,0
690 DATA 66000,60000,18000,16000,2,11,0
692 DATA 37000,60000,6700,5800,2,12,0
694 DATA 62000,110000,17723,18000,2,13,0
696 DATA 65000,100000,8500,3700,1,14,0
750 PRINT "GO TO JAIL."
752 PRINT "GO DIRECTLY TO JAIL."
754 PRINT "DO NOT PASS GO."
756 PRINT "DO NOT COLLECT $200"
758 GOTO 107
800 IF A<>1 THEN 850
810 PRINT "BULL RUN"
820 IF A1=1 THEN 1480
830PRINT"JULY 21,1861     GEN. BEAUREGARD COMMANDING THE SOUTH MET THE"
832PRINT"UNION FORCES WITH GEN MCDOWELL IN A PREMATURE BATTLE AT BULL"
834PRINT"RUN.  GEN. JACKSON HELPED PUSH BACK THE UNION ATTACK."
840 GO TO 1480
850 IF A<>2 THEN 900
860 PRINT "SHILOH"
870 IF A1=1 THEN 1480
880PRINT"APRIL 6-7,1862     THE CONFEDERATE SURPRISE ATTACK AT SHILOH"
882PRINT"FAILED DUE TO POOR ORGANIZATION."
890 GO TO 1480
900 IF A<>3 THEN 950
910 PRINT "SEVEN DAYS"
920 IF A1=1 THEN 1480
930PRINT"JUNE 25-JULY 1,1862     GENERAL LEE (CSA) UPHELD THE OFFENSIVE"
932PRINT"THROUGHOUT THE BATTLE AND FORCED GEN. MCCLELLAN AND THE UNION"
934PRINT"FORCES AWAY FROM RICHMOND."
940 GO TO 1480
950 IF A<>4 THEN 1000
960 PRINT "THE SECOND BULL RUN"
970 IF A1=1 THEN 1480
980PRINT"AUG 29-30,1862     THE COMBINED CONFEDERATE FORCES UNDER LEE AND"
982PRINT"JACKSON DROVE THE UNION FORCES BACK INTO WASHINGTON."
990 GO TO 1480
1000 IF A<>5 THEN 1050
1010 PRINT "ANTIETAM"
1020 IF A1=1 THEN 1480
1030PRINT"SEPT 17,1862     THE SOUTH FAILED TO INCORPORATE MARYLAND INTO"
1032PRINT"THE CONFEDERACY."
1040 GO TO 1480
1050 IF A<>6 THEN 1100
1060 PRINT "FREDERICKSBURG"
1070 IF A1=1 THEN 1480
1080PRINT"DEC 13,1862 THE CONFEDERACY UNDER LEE SUCESSFULLY REPULSED"
1082PRINT"AN ATTACK BY THE UNION UNDER GEN. BURNSIDE."
1090 GOTO 1480
1100 IF A <>7 THEN 1150
1110 PRINT "MURFREESBORO"
1120 IF A1=1 THEN 1480
1130PRINT"DEC 31,1862     THE SOUTH UNDER GEN. BRAGG WON A CLOSE BATTLE"
1140 GOTO 1480
1150 IF A<>8 THEN 1200
1160 PRINT "CHANCELLORSVILLE"
1170 IF A1=1 THEN 1480
1180PRINT"MAY 1-6,1863     THE SOUTH HAD A COSTLY VICTORY AND LOST ONE"
1182PRINT"OF THEIR OUTSTANDING GENERALS, 'STONEWALL' JACKSON."
1190 GOTO 1480
1200 IF A<>9 THEN 1250
1210 PRINT "VICKSBURG"
1220 IF A1=1 THEN 1480
1230PRINT"JULY 4,1863     VICKSBURG WAS A COSTLY DEFEAT FOR THE SOUTH"
1232PRINT"BECAUSE IT GAVE THE UNION ACCESS TO THE MISSISSIPPI."
1240 GOTO 1480
1250 IF A<>10 THEN 1300
1260 PRINT "GETTYSBURG"
1270 IF A1=1 THEN 1480
1280PRINT"JUNE 30,1863     A SOUTHERN MISTAKE BY GEN. LEE AT GETTYSBURG"
1282PRINT"COST THEM ONE OF THE MOST CRUCIAL BATTLES OF THR WAR."
1290 GOTO 1480
1300 IF A<>11 THEN 1350
1310 PRINT "CHICKAMAUGA"
1320 IF A1=1 THEN 1480
1330PRINT"NOV 25,1863     AFTER THE SOUTH HAD SIEGED GEN. ROSENCRANS'"
1332PRINT"ARMY FOR THREE MONTHS, GEN. GRANT BROKE THE SIEGE."
1340 GOTO 1480
1350 IF A<>12 THEN 1400
1360 PRINT "CHATTANOOGA"
1370 IF A1=1 THEN 1480
1380PRINT"SEPT 15,1863     CONFUSION IN A FOREST NEAR CHICKAMAUGA LED"
1382PRINT"TO A COSTLY SOUTHERN VICTORY."
1390 GOTO 1480
1400 IF A<>13 THEN 1450
1410 PRINT "SPOTSYLVANIA"
1420 IF A1=1 THEN 1480
1430PRINT"MAY 5,1864     GRANT'S PLAN TO KEEP LEE ISOLATED BEGAN TO FAIL"
1432PRINT"HERE, AND CONTINUED AT COLD HARBOR AND PETERSBURG."
1440 GOTO 1480
1450 PRINT "ATLANTA"
1460 IF A1=1 THEN 1480
1470PRINT"AUGUST, 1864     SHERMAN AND THREE VETERAN ARMIES CONVERGED ON"
1472PRINT"ATLANTA AND DEALT THE DEATH BLOW TO THE CONFEDERACY."
1480 PRINT
1485 RETURN
1487 PRINT "THE CONFEDERACY HAS SURRENDERED"
1488 GOTO 1500
1490 PRINT "THE UNION HAS SURRENDERED"
1500 PRINT
1510 PRINT "YOU HAVE WON" W; "BATTLES AND LOST" L; "BATTLES."
1515 IF Y=5 THEN 1550
1520 IF W<=L THEN 1550
1530 PRINT "THE CONFEDERACY HAS WON THE WAR"
1540 STOP
1550 PRINT "THE UNION HAS WON THE WAR"
1560 END
```

Another new game from Creative Computing . . . .

REVERSE

987654321

123456789

Description

In the computer game REVERSE the player must arrange a list of numbers in numerical order from left to right. To move, you tell the computer how many numbers in the list (counting from the left) to reverse. For example, if the current list is:

2 3 4 5 1 6 7 8 9

and you reverse four numbers, the result will be:

$\underbrace{5\quad 4\quad 3\quad 2}_{\text{first 4 numbers reversed from above list}}\quad \underbrace{1\quad 6\quad 7\quad 8\quad 9}_{\text{remainder of list stays the same}}$

Now if you reverse five numbers, you win!

$\underbrace{1\quad 2\quad 3\quad 4\quad 5}_{\text{first 5 numbers reversed from above list}}\quad 6\quad 7\quad 8\quad 9$

Playing Strategies

There are many ways to play the game; generally an approach can either be classified as *algorithmic* or *heuristic.* The game thus affords the player an opportunity to explore these concepts in a practical rather than a theoretical context.

An algorithmic approach is one that is described by means of a finite algorithm and guarantees a solution in a predictable number of moves. For example, an algorithmic approach to playing REVERSE would be to order the list from right to left starting with the highest value number and moving down. Using this strategy with a list of nine numbers, your first move would always be to get the 9 into position 1 (leftmost) and the second move would be to reverse nine so the 9 was put into position 9 (rightmost). You would continue moving the 8 to position 1 and then to position 8, the 7, 6, 5 and so on until the list was ordered. This method guarantees a solution in 2N-3 moves (N numbers in the list). One could easily program a computer to play this strategy.

A heuristic approach to solving a problem can be thought of as a rule of thumb. Some rules of thumb are very good and lead to good solutions, others are not as good. Consequently, using a heuristic approach doesn't guarantee the best possible solution but for very complex problems (and even some simple ones) it may be a more efficient approach than a rigorous linear programming or mathematical method which guarantees a perfect solution.

The science of heuristic problem solving using the computer has become very advanced and is widely used for things like locating warehouses, railroad car routing and other problems involving hundreds of variables and many alternative solutions. Consider: a linear programming solution to routing a mixed load boxcar from Boston to receiving points in Hartford, Columbus, Atlanta, and Baton Rouge would take about 0.72 hours to run on a computer. The heuristic solution takes 0.002 seconds to run, yet it generally yields a solution within 5% of the linear programming (perfect) solution. Obviously, with millions of cars to be routed every day, the linear approach is not economically feasible.

The game of REVERSE lends itself very well to a heuristic approach. There are many possible solutions to each game. One is best, but the mathematics to determine this solution are quite complex and would be extremely time-consuming to calculate. (The simpler algorithmic approach above guarantees a solution, but it is far from optimal). A good heuristic approach which takes advantage of "partial orderings" in the list generally yields a solution within 1 or 2 moves of the perfect solution, i.e., within 10% to 20% of perfection.

Using a heuristic approach, your next move is dependent upon the way the list currently appears. No solution is guaranteed in a predictable number of moves, but if you are clever (and lucky?) you should come out ahead of the simple algorithmic approaches. For a list with nine numbers can you describe a heuristic strategy that wins the game in an average of 10 or fewer moves? You may well use more than one rule of thumb (heuristic).

PROGRAM LISTING

```
100 PRINT\PRINT "REVERSE -- A GAME OF SKILL"\PRINT
120 RANDOMIZE
130 DIM A(20)
140 REM *** N=NUMBER OF NUMBERS
150 N=9
160 INPUT "DO YOU WANT THE RULES (YES OR NO)";A$
180 IF A$="NO" THEN 210
190 GOSUB 710
200 REM *** MAKE A RANDOM LIST A(1) TO A(N)
210 A(1)=INT((N-1)*RND)+2
220 FOR K=2 TO N
230 A(K)=INT(N*RND)+1
240 FOR J=1 TO K-1
250 IF A(K)=A(J) THEN 230
260 NEXT J\NEXT K
280 REM *** PRINT ORIGNIAL LIST AND START GAME
290 PRINT\PRINT "HERE WE GO ... THE LIST IS:"
310 T=0
320 GOSUB 610
330 INPUT "HOW MANY SHALL I REVERSE";R
350 IF R=0 THEN 520
360 IF R<=N THEN 390
370 PRINT "OOPS! TOO MANY - I CAN REVERSE AT MOST"N\GOTO 330
390 T=T+1
400 REM *** REVERSE R NUMBERS AND PRINT NEW LIST
410 FOR K=1 TO INT(R/2)
420 Z=A(K)
430 A(K)=A(R-K+1)
440 A(R-K+1)=Z
450 NEXT K
460 GOSUB 610
470 REM *** CHECK FOR A WIN
480 FOR K=1 TO N
490 IF A(K)<>K THEN 330
500 NEXT K
510 PRINT "YOU WON IT IN"T"MOVES !!!"\PRINT
530 INPUT "TRY AGAIN (YES OR NO)";A$
550 IF A$="YES" THEN 210
560 PRINT\PRINT "O.K.  HOPE YOU HAD FUN!!"\GOTO 999
600 REM *** SUBROUTINE TO PRINT LIST
610 PRINT\FOR K=1 TO N\PRINT A(K);\NEXT K
650 PRINT\PRINT\RETURN
700 REM *** SUBROUTINE TO PRINT THE RULES
710 PRINT\PRINT "THIS IS THE GAME OF 'REVERSE'. TO WIN, ALL YOU HAVE"
720 PRINT "TO DO IS ARRANGE A LIST OF NUMBERS (1 THROUGH"N")"
730 PRINT "IN NUMERICAL ORDER FROM LEFT TO RIGHT. TO MOVE, YOU"
740 PRINT "TELL ME HOW MANY NUMBERS (COUNTING FROM THE LEFT) TO"
750 PRINT "REVERSE. FOR EXAMPLE, IF THE CURRENT LIST IS:"
760 PRINT\PRINT "2 3 4 5 1 6 7 8 9"
770 PRINT\PRINT "AND YOU REVERSE 4, THE RESULT WILL BE:"
780 PRINT\PRINT "5 4 3 2 1 6 7 8 9"
790 PRINT\PRINT "NOW, IF YOU REVERSE 5, YOU WIN!"
800 PRINT\PRINT "1 2 3 4 5 6 7 8 9"\PRINT
810 PRINT "NO DOUBT YOU WILL LIKE THIS GAME OF SKILL, BUT"
820 PRINT "IF YOU WANT TO QUIT, REVERSE 0 (ZERO)."\PRINT\RETURN
999 END

READY
```

SAMPLE RUN

```
REVERSE -- A GAME OF SKILL

DO YOU WANT THE RULES (YES OR NO)? YES

THIS IS THE GAME OF 'REVERSE'. TO WIN, ALL YOU HAVE
TO DO IS ARRANGE A LIST OF NUMBERS (1 THROUGH 9 )
IN NUMERICAL ORDER FROM LEFT TO RIGHT. TO MOVE, YOU
TELL ME HOW MANY NUMBERS (COUNTING FROM THE LEFT) TO
REVERSE. FOR EXAMPLE, IF THE CURRENT LIST IS:

2 3 4 5 1 6 7 8 9

AND YOU REVERSE 4, THE RESULT WILL BE:

5 4 3 2 1 6 7 8 9

NOW, IF YOU REVERSE 5, YOU WIN!

1 2 3 4 5 6 7 8 9

NO DOUBT YOU WILL LIKE THIS GAME OF SKILL, BUT
IF YOU WANT TO QUIT, REVERSE 0 (ZERO).

HERE WE GO ... THE LIST IS:

 9  8  6  1  7  3  2  4  5

HOW MANY SHALL I REVERSE? 9

 5  4  2  3  7  1  6  8  9

HOW MANY SHALL I REVERSE? 4

 3  2  4  5  7  1  6  8  9

HOW MANY SHALL I REVERSE? 2

 2  3  4  5  7  1  6  8  9

HOW MANY SHALL I REVERSE? 6

 1  7  5  4  3  2  6  8  9

HOW MANY SHALL I REVERSE? 2

 7  1  5  4  3  2  6  8  9

HOW MANY SHALL I REVERSE? 6

 2  3  4  5  1  7  6  8  9

HOW MANY SHALL I REVERSE? 7

 6  7  1  5  4  3  2  8  9

HOW MANY SHALL I REVERSE? 2

 7  6  1  5  4  3  2  8  9

HOW MANY SHALL I REVERSE? 7

 2  3  4  5  1  6  7  8  9

HOW MANY SHALL I REVERSE? 4

 5  4  3  2  1  6  7  8  9

HOW MANY SHALL I REVERSE? 5

 1  2  3  4  5  6  7  8  9

YOU WON IT IN 11 MOVES !!!
```

Another new game from Creative Computing . . . .

SCHMOO

by Frederick H. Bell
University of Pittsburgh

Computers, Coordinates, and Schmoos

This Module is a computer-based educational (and fun) game with instructions for its use. It is written in elementary BASIC and is compatible with nearly all BASIC interpreters.

Getting Ready

Before teaching this lesson load SPLAT2 into your computer system, debug it, and save it for future access.

Things to Know

You need to know a little bit about grids and angles. Like, (2,-3) means right 2 and down 3, and 237° is in the fourth (Whoops! That's third.) quadrant. Also, you should remember that the distance something travels through the air depends upon the angle at which it is thrown.

Review the Basics

Can you answer these questions? If not, hit the math books!

1. In each of the four quadrants, what are the signs of the x- and y- coordinates?
2. If 0° is the angle coinciding with the positive x- axis, what are the measures of angles whose terminal sides fail in Quadrant I? Quadrant II? Quadrant III, Quadrant IV?

Lines 5 to 70 explain how to play SPLAT2. This is a fun game to play in groups of two or three. If you're pretty good you can "splat the schmoo" in about eight tries; but don't cheat and use the formula. And don't expect me to tell you where it's hidden in the program!

More Things to Do

You might want to make a three dimensional game, SPLAT3 – with flying schmoos. The program shouldn't be too hard and it would be a really neat game. If you want to try something easier, fix SPLAT2 so that it requires initial velocities as well as angles. You could even make a low gravity, moon version of SPLAT2.

Program Listing
Sample Output

REMarks About BASIC
REMark Statements

REMember to REMind yourself when writing BASIC REMark statements to REMain imaginative. If you are not REMiss in this, you can REModel your programs into REMarkable masterpieces with no REMainder of your REMote past before you applied this REMedy and REMoved those old, dull REMark statements. REMit to this REMedial advice and you'll have no REMorse. Before long, you can be REMiniscent about your old programs containing REMinants of ordinary REMark statements.

10 REMARKABLE REMARKS BY DHA

CREATIVE COMPUTING Reviews

Eagle-eyed readers will have noticed on the masthead that our reviews editor is now Peter Kugel. Peter is a professor in the Computer Science Program at Boston College. He is also an inventor of games (Bugs and Loops, On Sets, Queries and Theories, etc.), and was formerly with MIT's Electronics Systems Laboratory.

He is currently seeking additional reviewers for CREATIVE COMPUTING. If you wish to be a reviewer, please contact Peter directly: Prof. Peter Kugel, Computer Science Program, Boston College, Chestnut Hill, MA 02167. (617) 969-0100 X2475.

Electric Media, by Les Brown and Sema Marks, 160 pp paper, $3.30. Harcourt Brace Jovanovich, 757 Third Ave., New York 10017.

Electric Media is one of six books in Harcourt Brace's "Making Contact" series. It is a book that is at once fascinating, educational, and stimulating. A remarkable combination. It is even more remarkable when you realize that the authors are describing on the printed page two media, television and computers, which go far beyond ink on paper. Yet Les Brown manages to plug in the reader with arresting discussions of America's adjustment to television (97% of households watch TV!), equal time/fairness/quality issues, viewing habits, connection into the world, public TV, and cable TV. Also in the TV section are interviews with Dick Cavett and Edwin (Buzz) Aldrin.

If the TV section of the book rates as excellent, Sema Marks' section on computers must be considered superlative! These 50 pages will do more to enhance public understanding, acceptance, and appreciation of computers than all the textbooks in print. Even old hand computer people will have their eyes opened here. You'll read about popular criticism of computers and what's wrong with it. You'll marvel to computers doing once impossible problems. You'll worry when the data bank privacy dragon rears its ugly head. And you'll go bananas when you read about Alan Kay's "Dynabook" project and sample the dialogue of ELIZA and ANIMAL. Closing out the book is a 1972 interview with Marshall McLuhan in which he states, "1984 really happened a long time ago." How true. And what do we do now?

(Although this is a Harcourt Brace book, my experience is that it's very hard to find. You can order it through Creative Computing Library Service – see ad – or from Harcourt Brace directly – ISBN 0-15-318734-4, $3.30 plus shipping. Teacher's Manual – ISBN 0-15-318736-0, $1.20.)

David H. Ahl

Computers and Young Children. Nuffield Foundation. John Wiley & Sons, Somerset, NJ, 1972.

How and what to teach children about computers is the subject of this latest Weaving Guide produced by the Nuffield Mathematics Project. The main parts of this book consist of information about preparing flowcharts and samples of simple flowcharts produced by children; suggestions for a classroom activity in which the children act as human computers; information about using and preparing punched cards to present the program to the computer; and the description of classes actually working with a computer. The elementary classroom teacher will gain much information about computers from reading this text. She also will find many practical suggestions for teaching children about computers.

Computer Poems, collected by Richard W. Bailey. Potagannissing Press, Ann Arbor, Mich., $2.25.

Computer Poems is an anthology of verse written by sixteen poet-programmers; selections range from clever computer-constructed pieces to poems that are amusing despite their origin. Although programming techniques are not discussed, this book will surely interest the computer buff and the layman alike.

Background Math for a Computer World, by Ruth Ashley. John Wiley & Sons, Inc., New York, 1973. $3.95.

A self-teaching guide to the fundamental mathematical knowledge required for further study of computer programming or computer science. Problems from a wide variety of math application areas exercise analytical talents and help to develop logical thought patterns. This is an excellent preview for further work with computers for the nontechnically oriented educator.

Computers, by Jane Jonas Srivastava. Thomas Y. Crowell Co., New York, 1972.

This book is part of the Young Math Books series, and is geared to the very young (primary age) child, with delightful drawings by Ruth and James McCrea.

After beginning with "A computer is a machine for counting," the book goes on to describe various uses of computers, as well as the functions of the five basic parts of digital computers. There is even a super flowchart for "counting giraffes met on the way to school."

The point is made that not only can a computer do a job very quickly, but it will do the same thing over and over, without getting bored and asking, "When's recess?"

Above four reviews by:
Peg Pulliam
Lexington, Mass.

* * *

The Electronic Brain: How it Works, by Joseph J. Cook; 72 pp, $3.69; C. P. Putnam, 1969.

This book is written for students from grades five through eight, discusses the history, operation, and uses of computers. Unfortunately, the chapters are inconsistent in their approach, frequently omitting information necessary for the beginning student of computer science.

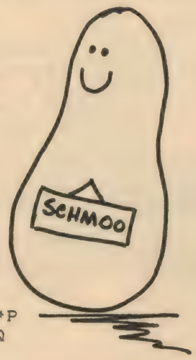
The chapter on the historical development of counting devices is both interesting and informative. It is a brief overview covering only the most important developments up to the MANIC – Mathematical Analyzer, Numerical Integrator, and Computer. Two chapters, "The Arithmetic of Electronic Computers" and "Inside the Electronic Brain" are written for those students who have a mastery of the decimal system and an aptitude for mathematics. All students, however, will understand and enjoy the chapters on present and future uses of the computer.

Although the format is academic and not catchy enough to generate immediate interest in the book, it should be in the school library as reference for those students who have an interest in learning about computers.

Flora Russ
Berkeley, California

Following is a LIST of SPLAT2:

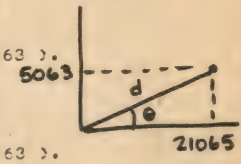
```
5 PRINT"THIS IS A NEW SCHMOO GAME.  SCHMOOS"
10 PRINT"ARE IMAGINARY CREATURES WHO LOVE"
15 PRINT"BEING SPLATTED WITH JUICY MUD BALLS."
20 PRINT"YOU, BEING A SCHMOO LOVER, TRY TO"
25 PRINT"MAKE SCHMOOS HAPPY BY TOSSING MUD"
30 PRINT"BALLS AT THEM.  YOU HAVE A"
35 PRINT"MECHANICAL MUD SLINGER THAT WILL"
40 PRINT"SLING MUD TO A MAXIMUM DISTANCE"
45 PRINT"OF 46,500 INCHES.  YOUR JOB IS TO"
50 PRINT"SET THE MUD SLINGER AT THE CORRECT"
55 PRINT"ELEVATION (0 TO 90) AND THE CORRECT"
60 PRINT"DIRECTIONAL ANGLE (0 TO 360)TO SPLAT THE"
65 PRINT"SCHMOO.  A HIT WITHIN 100 INCHES OF"
70 PRINT"THE SCHMOO WILL SPLATTER HIM."
75 PRINT
90 PRINT
95 RANDOMIZE
100 K1 = 0
110 Z = INT(1+RND*4-1E-8)
120 ON Z GOTO 130,140,150,160
130 P = -1
135 Q = -1
138 GOTO 200
140 P = -1
145 Q = 1
148 GOTO 200
150 P = 1
155 Q = -1
158 GOTO 200
160 P = 1
165 Q = 1
200 X = (INT (26000*RND+5000))*P
210 Y = (INT(26000*RND+5000))*Q
220 S = 0
230 K1 = K1 +1
240 IF K1<2 GOTO 400
250 R = INT(7*RND) + 5
260 GOTO 400
300 PRINT "THE ELEVATION MUST BE BETWEEN 1 AND 90."
310 GOTO 500
320 PRINT"DIRECTIONAL ANGLE MUST BE FROM 0 TO 360."
340 GOTO 500
350 PRINT"*SCHMOO SPLATTED*";S;" MUD BALLS TOSSED."
351 PRINT
352 PRINT"I SEE ANOTHER SCHMOO.  TO SPLAT"
354 PRINT"HIM, TYPE MUD.  TO QUIT, TYPE QUIT."
356 PRINT
358 INPUT C$
360 IF C$ = "MUD" GOTO 110
361 STOP
362 PRINT"YOU MISSED THE SCHMOO AT (";X;",";Y;")."
364 PRINT"YOUR MUD HIT (";INT(X1);",";INT(Y1);")."
366 PRINT
370 IF K1<2 GOTO 500
380 IF S>=R GOTO 800
390 PRINT"SCHMOO MUD HIT ";R2;" INCHES FROM YOU."
395 GOTO 500
400 PRINT
410 PRINT"COORDINATES OF SCHMOO ARE (";X;",";Y;")."
415 IF K1 <2 GOTO 420
417 PRINT"THE SCHMOO IS HAPPY TO BE SPLATTED."
418 PRINT"TO MAKE YOU HAPPY TOO,"
419 PRINT"HE WILL THROW MUD AT YOU."
420 PRINT
500 PRINT"MUD SLINGER ELEVATION";
502 INPUT B
504 PRINT"DIRECTIONAL ANGLE OF MUD SLINGER";
506 INPUT C
520 IF B = 90 GOTO 700
530 IF B >90 GOTO 300
540 IF B<1 GOTO 300
550 IF C<0 GOTO 320
560 IF C>(360-(1E-8)) GOTO 320
570 S = S+1
580 IF K1<2 GOTO 595
590 R2 = INT(ABS(300*RND*(11-2*S))+90)
595 J = 3.14159/180
596 D = ABS(INT(93000*SIN(B*J)*COS(B*J)))
610 X1 = D*COS(C*3.14159/180)
620 Y1 = D*SIN(C*3.14159/180)
630 D1 = SQR((X-X1)^2 + (Y-Y1)^2)
640 IF 100>=D1 GOTO 350
650 GOTO 362
700 PRINT"YOU DOPE!  YOU SPLATTED YOURSELF."
710 GOTO 900
800 PRINT"THE SCHMOO SPLATTED YOU!"
802 PRINT"CLEAN UP AND GOODBY!"
900 END
```

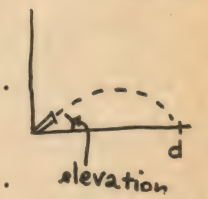


We didn't repeat the game instructions (Lines 5-70) here.

```
COORDINATES OF SCHMOO ARE ( 21065 , 5063 ).

MUD SLINGER ELEVATION ?10
DIRECTIONAL ANGLE OF MUD SLINGER ?23
YOU MISSED THE SCHMOO AT ( 21065 , 5063 ).
YOUR MUD HIT ( 14638 , 6213 ).

MUD SLINGER ELEVATION ?14
DIRECTIONAL ANGLE OF MUD SLINGER ?20
YOU MISSED THE SCHMOO AT ( 21065 , 5063 ).
YOUR MUD HIT ( 20513 , 7466 ).

MUD SLINGER ELEVATION ?13.5
DIRECTIONAL ANGLE OF MUD SLINGER ?18
YOU MISSED THE SCHMOO AT ( 21065 , 5063 ).
YOUR MUD HIT ( 20076 , 6523 ).

MUD SLINGER ELEVATION ?13.5
DIRECTIONAL ANGLE OF MUD SLINGER ?16
YOU MISSED THE SCHMOO AT ( 21065 , 5063 ).
YOUR MUD HIT ( 20292 , 5818 ).

MUD SLINGER ELEVATION ?13.5
DIRECTIONAL ANGLE OF MUD SLINGER ?15
YOU MISSED THE SCHMOO AT ( 21065 , 5063 ).
YOUR MUD HIT ( 20390 , 5463 ).

MUD SLINGER ELEVATION ?13.6
DIRECTIONAL ANGLE OF MUD SLINGER ?14
YOU MISSED THE SCHMOO AT ( 21065 , 5063 ).
YOUR MUD HIT ( 20623 , 5142 ).

MUD SLINGER ELEVATION ?13.7
DIRECTIONAL ANGLE OF MUD SLINGER ?13
YOU MISSED THE SCHMOO AT ( 21065 , 5063 ).
YOUR MUD HIT ( 20850 , 4813 ).

MUD SLINGER ELEVATION ?13.9
DIRECTIONAL ANGLE OF MUD SLINGER ?13
YOU MISSED THE SCHMOO AT ( 21065 , 5063 ).
YOUR MUD HIT ( 21130 , 4878 ).

MUD SLINGER ELEVATION ?13.9
DIRECTIONAL ANGLE OF MUD SLINGER ?13.5
*SCHMOO SPLATTED* 9  MUD BALLS TOSSED.

I SEE ANOTHER SCHMOO.  TO SPLAT
HIM, TYPE MUD.  TO QUIT, TYPE QUIT.

 ?MUD

COORDINATES OF SCHMOO ARE ( 15368 ,-16337 ).
THE SCHMOO IS HAPPY TO BE SPLATTED.
TO MAKE YOU HAPPY TOO,
HE WILL THROW MUD AT YOU.

MUD SLINGER ELEVATION ?15
DIRECTIONAL ANGLE OF MUD SLINGER ?316
YOU MISSED THE SCHMOO AT ( 15368 ,-16337 ).
YOUR MUD HIT ( 16723 ,-16151 ).

SCHMOO MUD HIT  1559  INCHES FROM YOU.
MUD SLINGER ELEVATION ?14.5
DIRECTIONAL ANGLE OF MUD SLINGER ?317
YOU MISSED THE SCHMOO AT ( 15368 ,-16337 ).
YOUR MUD HIT ( 16486 ,-15375 ).

SCHMOO MUD HIT  125  INCHES FROM YOU.
MUD SLINGER ELEVATION ?14.5
DIRECTIONAL ANGLE OF MUD SLINGER ?314
YOU MISSED THE SCHMOO AT ( 15368 ,-16337 ).
YOUR MUD HIT ( 15659 ,-16217 ).

SCHMOO MUD HIT  747  INCHES FROM YOU.
MUD SLINGER ELEVATION ?14.6
DIRECTIONAL ANGLE OF MUD SLINGER ?320
YOU MISSED THE SCHMOO AT ( 15368 ,-16337 ).
YOUR MUD HIT ( 17377 ,-14582 ).

SCHMOO MUD HIT  870  INCHES FROM YOU.
MUD SLINGER ELEVATION ?14.6
DIRECTIONAL ANGLE OF MUD SLINGER ?315
YOU MISSED THE SCHMOO AT ( 15368 ,-16337 ).
YOUR MUD HIT ( 16040 ,-16041 ).

SCHMOO MUD HIT  372  INCHES FROM YOU.
MUD SLINGER ELEVATION ?14.6
DIRECTIONAL ANGLE OF MUD SLINGER ?314.5
YOU MISSED THE SCHMOO AT ( 15368 ,-16337 ).
YOUR MUD HIT ( 15900 ,-16181 ).

THE SCHMOO SPLATTED YOU!
CLEAN UP AND GOODBY!
```

It may be easier with a diagram –





Why does the Schmoo throw mud at you this time??

BOOKS ON COMPUTER LITERACY

Peter Olivieri
Boston College

There are a great many textbooks, journals, articles, and publications that deal, on an instructional level, with the art and science of computing. However, these are often too technical for the casual reader, the individual who is merely seeking a brief introduction to what computers and computing is all about. These people have needs that I feel we all too often overlook. Providing the general public with more information about computers is something that we all should strive for. As computers become more commonplace, it is not unusual to find elementary school children (as early as Grade 3) being exposed to these machines. Materials need to be developed that provide a non-technical, informative, and meaningful introduction to computers for people from age six to sixty. I have recently reviewed several such publications and present them here for addition to your memory banks should you have occasion to recommend something to a relative, friend, or acquaintance. This is by no means a collectively exhaustive list of all materials of this type, but rather a beginning.

THE STORY OF COMPUTERS. Donald Spencer. 64pp. Abacus Computer Corporation, Ormond Beach, Florida 32074. Paper. 1975. A nicely illustrated, clear, non-technical introduction to the world of computers. Geared for children in Grades four through nine. Recommended for your young children. Includes: What are computers? Who invented computers? What kinds of computers are there? How does a computer work? How do I tell a computer what to do? What are some useful computer terms?

COMPUTERS. William R. Corliss. 91pp. United States Atomic Energy Commission. Office of Information Services. Paper. 1973. A very readable book. Quite current (includes a section on optical reading of supermarket products). A little technical in parts (negligible). Includes: Calculating Prodigies. The Birth of Computers. Anatomy of a Computer. The Generation Gaps. Analog Computers. A Reading List. Selected Motion Pictures (too few of them).

THE FIRST BOOK OF INFORMATION SCIENCE. Joseph Becker. 91pp. United States Atomic Energy Commission. Office of Information Services. Paper. 1973. Not too much on computers. Excellent for the information on information sciences. Good photographs. Recommended for those with an interest in information and data. Includes: What is information? Storing and retrieving information. Putting information into a computer. Getting information from a computer. Communicating information. Finding information in microfilm. Future of information sciences. Epilogue. Glossary and a reading list.

COMPUTERS IN ACTION. Donald Spencer. 245pp. Hayden Book Company, New Jersey. Paper. 1974. A guide for the layman, businessman, high school student, teacher, or non-scientific college student. Excellent references at end of each chapter. Easy reading but not quick reading. Somewhat like a textbook. Actually teaches BASIC to the reader. Includes: The Computer Age, Computer Evolution, How Computers Work, Getting Information In and Out of the Computer, Computer Storage, Designing the Computer Program, The Language of the Computer, Introduction to Computer Programming.

UNDERSTANDING COMPUTERS. Thomas H. Crowley. 139pp. McGraw-Hill, New York. Paper. 1967. Written for high school teachers, business managers, and those with limited technical background who are interested in finding out what the computer business is all about. Not quick reading. A learning book rather than an information book. Somewhat outdated though not critically so. Strongly recommended for the serious layman. Includes: Introduction, Basic Computer Functions, What Is Being Processed? Interpretation of Symbols, Memory, Input-Output Operations, Symbol-Processing Operations, The Control Process, Stored Program Computers, Programming, Applications of Computers, Computer "Priests", What Does the Future Hold? 1984, Bibliography.

THE WAY THINGS WORK BOOK OF THE COMPUTER. Simon and Schuster. New York. 245pp. 1974. This is an illustrated encyclopedia of information science, cybernetics, and data processing. Very nicely illustrated. Clear, two-color charts and diagrams. Not light reading. Information packed. Some material a bit technical. Certainly recommended for your bookshelf. Might be geared more for the information or computer scientist, but there is enough here to satisfy many tastes. Only dissappointment was that it didn't go far enough in coverage. Another volume would be well received. Includes: Information and Documentation, Information and Cybernetics, Communication and Documentation, Statistics and Thermal Processes, Control, Self-correction and Automation, Information Theory, Redundancy, Cybernetics and Linguistics, Cybernetics and Physiology, Cybernetics and Psychology, Cybernetics and Biology, Cybernetics in Teaching, Cybernetics in Sociology, Cybernetics in Economics, Documentation, Number Systems, The Binary System, Binary Arithmetic, Logical Operations, Electromagnetic Switches, Electron Tubes, Semiconductor Logic Elements, Photo-electric Logic Elements, Fluidic Switches, Mechanical Storage Devices, Magnetic Storage Devices, Acoustic Data Storage Systems, Thin-Film Storage Devices, Analog Computation Principles, Data Processing Systems, Machine Languages and Symbolic Languages, Programming, Character Recognition, High Speed Printers and Filmsetting Techniques, Time Sharing Systems, Data Transmission, Process Computers, Data Banks, Artificial Intelligence.

* * *

Computers: Tools for Today, by Claude J. DeRossi, $4.75; Children's Press, Chicago, IL, 1972.

Claude DeRossi has done an excellent job in presenting the intricacies of the computer world to the very young in his book entitled, *Computers: Tools For Today*. The book is written at a level understandable by students of the upper grades in elementary schools. The drawings by Margrit Fiddle together with the pictures are especially helpful in illustrating some of the concepts presented.

The objective of the book is to present a general picture of what computers are and what they can and cannot do. Some misconceptions concerning the capabilities of computers are discussed toward the beginning of the book, pointing out to the novice potential computer user that computers cannot solve all problems or answer all questions.

The book is well organized in that many facets of computers are discussed. The readers are introduced to computers by first discussing the history of computers. A couple of chapters discuss the capabilities and limitations of computers. Other chapters include the discussion of the following computer characteristics; input devices (punched cards and magnetic tape), how computers work (add), the numbering system of computers (binary system), and computer language (programs and how they are written including flow charts).

Joseph O. Garcia
Albuquerque, New Mexico

"He who has imagination without learning has wings and no feet."
Joseph Joubert

Computers and You, Kurt R. Stehling; 246 pp; $1.50. A Mentor Book (New American Library), 1301 Avenue of the Americas, New York, NY 10019; 1972.

In an easy-to-read book Kurt R. Stehling outlines the impact of the computer in several fields. The phenomenal growth and impact of computers and technology in education, transportation, medicine, weather, defense and space, business and commerce, the government and social uses is presented in separate chapters. The detail is sufficient to generate a certain amount of appreciation in addition to the acquisition of knowledge about computer uses. A general background and explanation of computers introduces the reader in a quiet way to this computer technology. A Glossary of Computer Terms is also included to help the novice reader with any unfamiliar terms or jargon.

The author expresses his ideas in a very readable fashion with parenthetical dry wit that is entertaining. Unfortunately, a sexist bias is evident in such references that equate a card punching phase with 200,000 woman days rather than worker days and a woman fumbling for change rather than a person. Such stereotyping is certainly unnecessary to accomplish the objective of the book.

The text can be easily read and enjoyed by an adult audience of general readers. Technical explanations are minimal and computer applications are explained in depth. It is an informative and enjoyable book to read.

Jane Donnelly Gawronski
San Diego, Calif.

* * *

Computer-Assisted Instruction Project Final Report, by Alex Dunn and Jean Wastler. 507 pp., $12.50, Montgomery County Public Schools, Rockville, Maryland, 1972.

This is the final report of a three year computer-assisted instruction project in the Montgomery County Public School System, Montgomery County, Maryland. It provides those interested with the account of how a school system planned, developed, and implemented computer-assisted instruction.

The report includes a description of the instructional modules developed and used in the project. The documentation for each unit includes behavioral objectives, description of how the unit can be used, and the mode of presentation to the students. The units range from elementary school subjects such as operations with whole numbers through senior high school subjects such as trigonometry, chemistry, and physics. The text is illustrated with many flow-charts showing the instructional strategy of each unit and with sample computer print-outs showing the students' interaction with the computer.

This well written report also includes a description of the validation and evaluation of the project, a cost analysis of the operation, and a description of the computer system used.

This book should be recommended to administrators and supervisors who might be considering computer-assisted instruction as a means of using today's technology to individualize instruction.

Bruce W. De Young
Oakland, NJ

* * *

Snobol: An Introduction To Programming, by Peter R. Newstead. 160 pp paper. $5.25. Hayden Book Co., 1975.

Every budding programmer deserves to have his mind blown periodically, to extend his view of what can be done with computers and how differently ideas can be expressed within a computer language framework. For anyone whose sole experience is with the usual algorithmic languages, e.g. BASIC, FORTRAN, COBOL, PL/I, or even Assembly languages, an introduction to the power of expression inherent in SNOBOL can be a refreshing experience. I feel that it should be more widely available than it is, even though SNOBOL4 is outnumbered in its implementations by only a few of the above. It is wider in its applicability, especially for students who have not yet crystallized their occupational goals, than many of the others; yet most initial introductions into programming these days appear to be into BASIC.

The present book is designed to aid in teaching SNOBOL as a first language. Its treatment of computing in general is short and elementary and largely that of analogy. The teacher may have his own methods for this phase; chapter 1 is simply one point of view. The introduction to the language itself starts gently in chapter 2 and builds at a good pace, with lots of examples and exercises. Details of coding conventions and the intricacies of pattern matching follow in subsequent chapters. There is a good chapter on debugging and a chapter on user-defined functions which just skirts the edge of comprehensibility for a beginning programmer.

How good is the book? I think it will demand a good teacher with some experience and perhaps affection for the language, but it seems to be a good beginner's text. Any SNOBOL text is bound to have difficulty in competing with those by Griswold (the original SNOBOL author) et al: Griswold happens to be an excellent writer as well as authoritative. But the present book may be more useful as a beginner's text than some of the older books on the subject.

Lynn Yarbrough
Lexington, Mass.

* * *

Business Data Processing I, by Robert Albrecht. Student workbook $2.00, Teacher's guide $1.50. Digital Equipment Corp., Maynard, MA 01754; 1974.

This is probably unlike most of the business data processing books you've seen. This one gets right into BASIC programming. The intent of this is to develop a series of programs to manipulate data. Working with subscripted variables, manipulating lists, and sorting lists are at the heart of this workbook. This treatment of subscripted variables is one of the best I've seen. A fresh writing style, plenty of annotated programs, drawings, and a host of both programming and non-programming exercises combine to make this a winner. The teacher's guide is but a collection of solutions. I would have liked to see teaching suggestions, alternate and additional exercises, more programming problems, etc., in such a guide. If you know this material, you don't really need the solution guide.

Joseph Kmoch
Milwaukee, WI

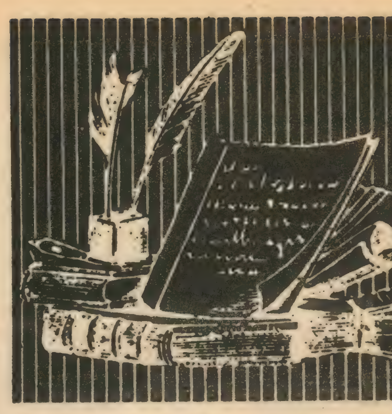
AMERICANS SPEND HALF AS MUCH ON THEIR CHILDREN'S TEXT BOOKS AS THEY DO TO FEED THEIR DOGS AND CATS

DURING THE LAST CENTURY 400,000 TITLES HAVE BEEN PUBLISHED IN THE FIELD OF PSYCHOLOGY

CREATIVE COMPUTING
Feature Review

34 Books on BASIC

Stephen Barrat Gray
Gray Engineering Consultants
260 Noroton Ave.
Darien, Conn. 06820



Installment No. 3. The overall summary and reviews of the first four books appeared in *Creative Computing* Vol. 1, No. 3. Reviews of the second four books appeared in Vol. 1, No. 4.

For a future group review of books on applications of BASIC, I would appreciate information concerning such publications. This would include not only books such as Peckham's *Computers, BASIC and Physics*, but also applications books not oriented toward any particular language, but which could be used with BASIC, such as Gruenberger & Gaffrey's *Problems for Computer Solution*.

Also appreciated is information about books on BASIC in languages other than English.

9. *BASIC For Beginners*, by Wilson Y. Gateley and Gary G. Bitter. Pub. June 25, 1970, by McGraw-Hill, New York, N. Y., 152 pages, 5½ x 8, $5.50 (paperback).

Very good for its limited coverage. Rating: B+

Two striking features of this book are its pleasing design and handsome typeface; this is one of the best looking books. All the programs and program-line examples are in an easily-read sans-serif type.

"This book is a descendant of one written during the Spring of 1968 for the purpose of providing a self-instructional manual for students at Colorado College." It doesn't actually get into BASIC until page 40. First comes a nice introduction consisting of a chapter on The Beginning (programming, computers, languages) and one on Using the Terminal (33 and 35 Teletypes, system commands, paper tapes, correcting errors). The actual text on BASIC runs up to page 117, a total of 78 pages, covering 26 statements. GOSUB and RETURN are not covered "because of our feeling that the subroutine concept, although admittedly of great usefulness in complicated programs, is of little value in most beginner's programs and is more likely to confuse than to help the novice."

There are eight chapters: The Beginning; Using the Terminal; A Start At BASIC; BASIC Control Statements; Loops and Arrays; Library Functions and DEF; Input, Printing and String Data; The End and The Beginning.

There are questions at the end of each chapter, followed by the answers, and then by several exercises, without answers. Each exercise requires writing and running a program to solve the given problem.

The start is slow and conversational, with much "you" and "we," and the coverage is quite thorough. The examples (other than program lines) are not set apart from the text, as in most other books, but are included within the body of the text: "Thus X, B, C1 and N9 are legal variables, but 9N, BX, and X23 are not." This makes the reading flow somewhat more easily than in the books that give a larger number of examples and set them apart, but some may prefer more examples over easier reading.

A unique feature is Chapter 8 on The End and The Beginning, which is four brief pages on various subjects not covered in the previous chapters, such as editing commands, MAT statements, data files, and FORTRAN.

There are three appendixes. The first is a table of the differences between 15 time-sharing systems as to 13 features and limitations, such as maximum number of digits in a constant, whether variable initialization to zero is automatic, etc. The second appendix shows how a simple four-line program that calculates several combinations of two constants will look if run on 14 different systems. Actually, one or two of the runs, within the main text, would suffice. The third appendix consists of additional exercises, providing two problems in each of 13 disciplines, from algebra ("write a program which will carry out synthetic division") to political science ("write a computer program which . . . computes both the Democratic percentage vote and the Republican . . ."); answers are provided for these exercises.

This is one of seven books that give statements and functions on the inside covers, for ready reference.

There are not as many examples as the other texts give, such as for constants, variables, formulas, etc. And there are only nine complete programs in the entire book, although there are 14 more programs in the questions at the end of the chapters.

The first program in the book is a four-liner that prints several combinations of two assigned constants, and which is explained briefly, on page 23 of the chapter, Using the Terminal, which is long before BASIC is actually taught, starting on page 46. The second program is not in the chapter on A Start At BASIC, but actually appears in the questions at the end of that chapter, on page 56.

The language of the text is a little too involved at times, due mainly to the constant use of "you" and "we," which makes for involved phrases and sentences.

On the whole, a nice little book, with its limitations due mainly to its brevity: 78 pages plus wraparound.

* * *

10. *Discovering BASIC: A Problem Solving Approach*, by Robert E. Smith. Pub. Aug. 27, 1970, by Hayden Book Co., New York, N. Y., 203 pages, 5½ x 8¼, $7.95 (hardcover), $5.95 (paperback).

Only for the hardworking and conscientious student. Must be used with a terminal. Rating: As a second or third book, B+; as a first book, C–

This is a bound version of the author's looseleaf, ringbound "BASIC Ideas," published in 1969 by International Timesharing Corp. in Minneapolis, Minn., at $5.95. The idea of the original edition was that the pages could be removed from the binder and easily placed at or on the terminal, which is difficult or impossible with a bound book. The bound edition is identical in content with the looseleaf one.

The book is in two distinct parts. The first 95 pages consist of 41 lessons. The book teaches by asking the reader to run a given program on the computer, and builds the text around such programs, most of which are given with little or no explanation as to how they work, other than the accompanying flowcharts. The reader must figure out a great deal by himself, and is not given adequate preparation in many cases. A program for building magic squares is given as early as page 16, 20 lines long, with a skeletal flowchart and no explanation of what a magic square is, much less some explanation of how the program works.

A time-sharing terminal is essential because most of the lessons present a program and a flowchart, without any explanation or REM statements, only an exhortation to "Try it!" So if the program runs, the reader has an answer, but he probably won't understand how or why. This must be the "problem-solving approach" denoted by the book's subtitle. If so, it's a misnomer, as the computer solves the problems, not the reader. Of course, a bright reader could probably figure out what's going on, but should a programming book be written only for the top five percent (or less) of the population?

There are four review tests within the first 95 pages, and they are scored by running given programs, with one's answers put into the DATA statements. There are no explanations whatsoever as to how these scoring programs work; this too is left as an exercise for the student.

The writing style is not at all smooth, with such cute phrases as "by tags we mean the cute little messages . . ."

Many statements are presented without explanation of any kind, such as RND in the magic-square program. The REM statement gets only 26 words of explanation. The few times a reader is asked to write a program, in the 41 lessons, he usually has not been given enough information beforehand to be able to do so, and since most of the programs are given in the text, he has had very little experience in writing programs – he's been running the author's programs nearly all the time.

In a program that determines the highest common factor, there is no explanation that program lines 45 to 55 exchange the values of X and Y, one of the crucial parts of the program.

Page 34 gives an uncompleted program, the reader being expected to fill in the missing seven lines. Some readers may learn from this type of teaching, but not many.

A program on page 66 contains a command that no other author mentions: PAUSE.

The second part of the book consists of 50 review problems, each with a short description of the problem, and a flowchart. The reader is expected to write programs that solve a great variety of problems, such as radix sorting, simulating a dice game, rank correlation, annuity interest, linear correlation, etc. A couple of the problems have complete or partial programs. The reader could learn much more if the programs required were simpler to write, and if he had to write his own flowcharts. As it is, many of the 50 review problems leave too much to the imagination. Review problem 6 is on a chi-square test, without bothering to explain what a chi-square test is. But a full 50-line program is given.

There is no index.

The last 39 pages contain program solutions to problems in the text, photographically reduced to the eye-straining dimension of 22 characters per horizontal inch.

A reader *can* learn from this book if he will conscientiously dig his way through the lessons and programs. The question is, will he do this, or get bored early, and start to skip pages? For the very bright, with highly inquisitive minds, there is something to be mined here – the hard way. For the rest, meaning the majority, the book as a first text is much too difficult.

The back cover notes that the author's books on computer programming "are distinguished by his originality of presentation and his ability to clarify computer languages." That latter claim may well be true of the author's FORTRAN and COBOL books, but not of this one.

* * *

11. *Basic BASIC: An Introduction to Computer Programming in BASIC Language*, by James S. Coan. Pub. Sept. 11, 1970, by Hayden Book Co., New York, N. Y., 256 pages, 6 x 9, $8.25 (hardcover), $6.50 (paperback).

Despite some drawbacks, a useful and helpful book. Rating: B+

There are many good points to this book, some of them unique. The statements, with brief explanations, are presented in boxes, and so stand out loud and clear. Some other authors intimidate the reader by presenting long and complex programs much too early; early in his book, Coan gets the reader used to the sight of long programs, but they are relatively simple ones. The 50-line program on page 36 is easy to understand; it has many explanatory PRINT lines and concerns various ways of printing the items in a list. This is in the chapter on Loops And Lists, which starts out, like the other chapters, with short programs (6 lines here) and builds up to larger ones.

There are 13 chapters: Introduction to BASIC, Writing a Program, Loops And Lists, Computer Functions, Elementary Data Processing, Introduction to INPUT and RESTORE, Specific Applications (Euclidian algorithm, change of base, looking at integers digit by digit), The Quadratic Function, Trigonometry, Complex Numbers, Polynomials, MAT Instructions, Elementary Probability. There are seven appendixes: Storing Programs on Paper Tape, Error Diagnosis, Special Formatting Functions (TAB, IMAGE), Summary of Flowchart Shapes, Summary of Statements in BASIC, Index of Programs in Text, Answers to Even-Numbered Problems.

There are problems for each section within a chapter. Some problems are check-marked to indicate that they are the more difficult ones.

Some fairly complex programs are presented, such as the one on page 85 on questionnaire analysis, which very few authors get into. A few pages later, the author mixes two programs (two-way temperature conversion) and calls one or the other with 0 or 1, a unique program in these books. Not only is Coan the only author to go into complex numbers, he has an eight-page chapter on the subject. He is also the only one to give programs for synthetic division, integral zeros, real zeros, complex zeros, and the Descartes rule of signs, in a chapter on polynomials. And there is a fine seven-page appendix on Error Diagnosis, explaining the three types of errors in detail.

The items on the minus side of the ledger may not bother every reader, but can be annoying to some. The type is small and uncomfortable to read; the back-of-the-book answers to problems are tiny photoreductions of Teletype output, 27 miniscule characters to the horizontal inch.

The writing style is odd, as though it were a transcription of classroom lectures, quite prosaic and showing little imagination. For example the definition of IF-THEN is "XXX IF YYYYYY THEN ZZZ. If YYYYYY is true, transfer to line ZZZ. If YYYYYY is false, pass to the next line after XXX." Not very helpful.

Some features of BASIC are given very short shrift. The explanation of the E format for exponentiation takes all of one sentence, and there is none at all for negative E.

Some parts of the book, such as the portion on testing integers for divisibility, on page 102, contain mathematical manipulations that would be hard for many to follow without a teacher for guidance.

The final chapter, on Elementary Probability, is one that only the top students may understand. The chapter contains a teaser, in showing a RUN that gives all the four-letter combinations of the word FLAG, but not the program itself. The excuse given is that "the techniques required for this vary so greatly from system to system that we will not present the program, but only the RUN."

Most of these drawbacks can easily be ignored by the reader who is after the many excellent parts of this book, which in a future edition may become an outstanding one.

* * *

ALL 4 OF THESE GREAT HISTORY BOOKS FREE

WITH YOUR SUBSCRIPTION TO MANKIND MAGAZINE

1 **THE HUMAN SIDE OF HISTORY**
The fascinating history of man's human side—manners, morals, and games—is not only informative but also fun. Included in this book are a survey of the first one hundred years of college football and histories of cosmetics, glass, ice cream, and (would you believe?) home laundry. On the serious side there is an interesting new look at the Victorians and studies of juvenile delinquency, prehistoric diseases, and the black plague. Illustrated with forty-seven drawings and paintings. 256 pages.

2 **THE AMERICAN INDIAN**
Only within the last few years have white historians begun to look at the history of the American Indian objectively. This unique collection covers the highlights of Indian history primarily from the Indian Point of view. Included are a history of the Pueblo Indians of New Mexico, the tragic story of the removal of the Cherokees from their ancient homeland, the story of the man who almost organized the Indians into a unified fighting force, and a study of American Indian humor. Illustrated with fifty-four photographs, drawings, paintings and maps. 256 pages.

3 **GREAT MILITARY CAMPAIGNS**
"War is hell," said William Tecumseh Sherman and war is particularly hell for the losers. This collection of articles on great military campaigns concentrates, to a great extent, on the unsung heroes of losing causes such as Xavier Mina who led a force of Spanish guerrillas against Napoleon during the Peninsular Campaign of 1808–1814; Tyrone of Ulster who fought against the British for Irish independence; Boabdil of Granada who lost Moorish Spain to the Christians, and Finland's Winter War of 1939. Illustrated with forty-six photographs, drawings, paintings and maps. 256 pages.

4 **THE ANCIENT WORLD**
Among the scholars represented in this volume are the renowned Dr. Michael Grant, former president of the Queen's University, Belfast, Ireland, Dr. Marvin Berry of the University of Southern California, and Dr. Paul I. Maier of Western Michigan University. The articles range from Dr. Berry's prize-winning "The Age of Solomon," in which he questions the myth of King Solomon the Just, to Dr. Grant's study of the early centuries of Christianity; from a comprehensive profile of Alexander the Great to Dr. Maier's unforgettable study of Pontius Pilate. Illustrated with thirty-four photographs, drawings, paintings and maps. 256 pages.

Experience history alive with verve and color to stimulate your understanding and sharpen your vision of man and his adventure on earth. Share Lord Byron's vision of Greece. Travel the western badlands with Jesse James. Open an Egyptian tomb. Charge up San Juan hill. Walk the streets of Colonial America or have dinner with an Indian tribe. Each issue of *Mankind* magazine brings you the delight of fresh, bold, unexpected ideas, beautifully illustrated and superbly written. Share with us the pleasure of *Mankind*. Send in your subscription today.

ALL 4 BOOKS FEATURED ABOVE YOURS FREE IF YOU SUBSCRIBE NOW!

MANKIND MAGAZINE, Dept. 10 MD06
8060 MELROSE AVE./LOS ANGELES, CALIF. 90046

Enclosed is $12.00 ☐ cash, ☐ check, ☐ money order payment in full for my 12-issue subscription to Mankind magazine—a savings of $3.00 off regular single copy price. Also send me the four featured books free as my subscription bonus.

Name ______________________

Address ______________________

City ______________________

State ______________ Zip ______________

creative computing

BACK ISSUES

NOV-DEC 1974 – CHARTER ISSUE (Out of print)

JAN-FEB 1975 – GENERAL TOPICS (Out of print)

MAR-APR 1975 – COMPUTER CAREERS
A Computer Career for You?
Profile of the Computer Industry – Job opportunities.
Shorthand Instruction via Computer.
Computer Cartoons – a complete analysis.
Games: Rabbit Chase, Not One, Binary.
Review of BASIC books – Summary and first four reviews.

MAY-JUN 1975 – SPACE EXPLORATION
Games: Super Star Trek, Lunar Landing, Parachute Jump on Any Planet, ICBM Missile Intercept, Seawar, Geowar.
Escape – A simulation about escaping the earth's gravitation.
Mr. Spock's 7th Sense – real computer sensors.
Trends in Mathematics Curriculum Research.
Computer Simulation of the Atmosphere.
Structured Programming.
The Automobile and Air Pollution – Writing your own computer simulation.
Mr. Spock centerfold poster.

Back issues are $1.50 each paid in advance. A billing charge of $1.00 is added to all orders without payment enclosed.

BULK SUBSCRIPTIONS

Bulk subscriptions are now available to *Creative Computing.* All copies must be sent to the same address; minimum term of the subscription is one year.

Quantity	*Price*
10-24	$7.00 each
25-49	6.00
50-99	5.00
100+	4.00

COMING SOON

NOV-DEC 1975 – THE COMPUTER IMPACT ON SOCIETY
Isaac Asimov in a new story "The Life and Times of Multivac" looks at what happens after a nuclear holocaust when all the computers on earth link up to support the remaining human survivors.
Surveys, the Census, and Privacy – a new classroom activity.
Point of Sale Computer Systems.
Computer Crime and the White Collar Criminal.
Computer Security – A mini course.
Building Your Own Computer – A survey of sources, kits, plans, and amateur computer clubs.

JAN-FEB 1976 – GAMES AND PUZZLES
Re-inventing the Wheel – A collection of commonly-used routines and algorithms.
Chess programs.
Games, games, games.
The Complete Computer Catalog (3rd installment).

MAR-APR 1976 – ARTIFICIAL INTELLIGENCE AND FUTURE COMPUTERS
Smaller, faster, cheaper computers.
Access to computers in museums, libraries and storefronts.
At what point does a computer start to think?
Review of every BASIC book in print (6th and last installment).

MAY-JUN 1976 – COMPUTER ART
This fantastic double-size, full color, quality paper, large format issue will feature the works of 40 leading computer artists from around the world. Guest editor for this special issue is Ruth Leavitt, an outstanding computer artist.

Closing date for articles or other material for issues is 3½ months before the cover date. If you want an acknowledgement of your submittal, please include a self-addressed stamped envelope.

CHANGE OF ADDRESS

Moving? Changing jobs? If so, to insure you get every issue of *Creative Computing,* please send us both your OLD and NEW address. Your "GC" code number which appears above your name on the address label would help too. And by all means, don't forget ZIP code. (We heard recently that the USPS automatically incinerates all magazines without zip codes.)

CREATIVE COMPUTING SUBSCRIPTION FORM

Published bi-monthly – 6 issues per year

All subscribers will receive a $5.95 computer art book FREE!

1\.

Quan.	*Type*	*Term*	*Price Each*
_____	Institutional	1-Year	$15.00
_____	Institutional	3-Year	$40.00
_____	Individual	1-Year	$ 8.00
_____	Individual	3-Year	$21.00
_____	Student	1-Year	$ 6.00
_____	Student	3-Year	$16.00
_____	Bulk (10-24 copies)	1-Year	$ 7.00
_____	Bulk (25-49 copies)	1-Year	$ 6.00
_____	Bulk (50-99 copies)	1-Year	$ 5.00
_____	Bulk (100+ copies)	1-Year	$ 4.00
_____	Sample Copy		$ 1.00
_____	Foreign Postage	$1.50 per year	

2\. *New or Renewal*

☐ New
☐ Renewal
☐ Gift

3\. *Billing Instructions*

☐ Payment enclosed
☐ Please bill me ($1.00 billing charge will be added except on institutional and bulk subscriptions.

4\. *Name and Address*

Name ______________________ Title/Dept. __________

School or Company ______________________

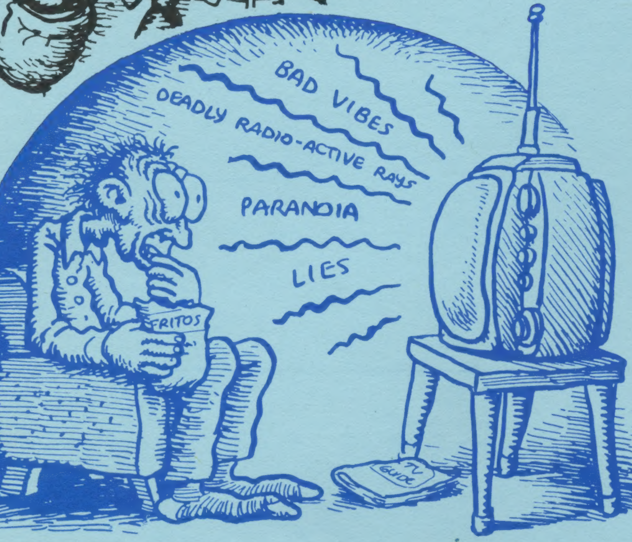
Street Address ______________________

City ______________ State _____ Zip _______

Return form to: CREATIVE COMPUTING, P.O. Box 789-M, Morristown, New Jersey 07960

DON'T YOU THINK IT'S TIME TO Stop Watching T.V.?

HERE'S WHY!



TV makes people SICK!! TV ENSLAVES you and SAPS ALL YOUR CREATIVE ENERGY! TV HOOKS YOU like DOPE! TV is obviously a "VAST WASTELAND." This is common knowledge. TV is just plain BAD for you physically, mentally, and spiritually. Watching it will cause you great unhappiness in life and you're more than likely to get CANCER!!

IF YOUR CREATIVE INSANITY IS SEEKING A REAL OUTLET, THEN READ

creative computing

It's the magazine that answers to NO ONE!! CREATIVE COMPUTING is the Last outspoken Bastion of TRUTH left in America today. Read it and you shall be free! Contained within these pages are mind-blowing glimpses of REALITY as it REALLY IS! This magazine BREAKS THROUGH the TV-induced stupor, for it is ANTI-MEDIA! It has things for you to ACTUALLY DO! It's got the medicine for the BLUES, and has been known to turn MENTALLY ILL persons into HEALTHY, GOOD-HUMORED FREE-THINKERS. You too will benefit from reading CREATIVE COMPUTING!!

Only \$8 per year (\$6 for students)
CREATIVE COMPUTING, Dept. C,
P.O. Box 789-M, Morristown, NJ 07960.