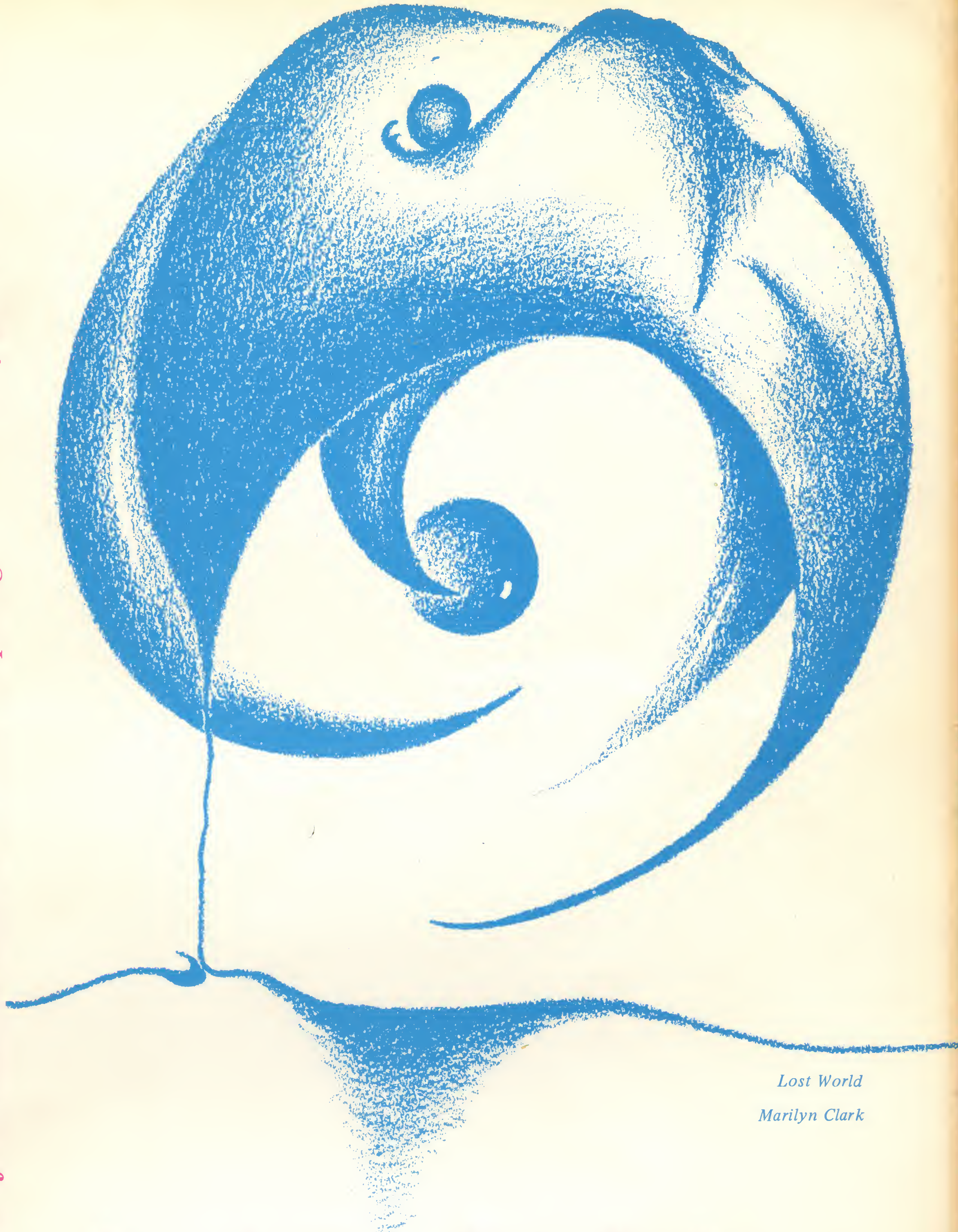
creative computing

the magazine of recreational and educational computing

May-Jun 1975 $1.50

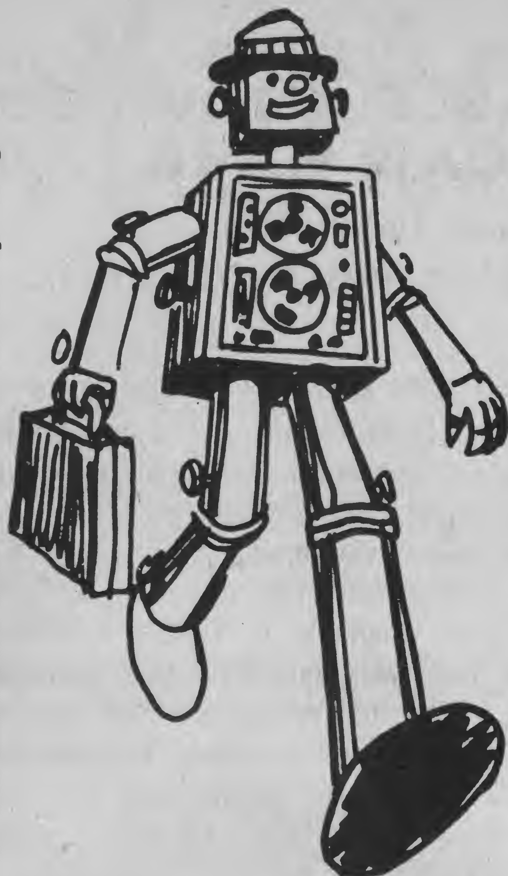


Lost World
Marilyn Clark

Atmosphere Smog, Weather
Space Exploration, Super Star Trek Game
Book Reviews, Puzzles, Future City Comics

creative computing

★ a non-profit magazine of educational and recreational computing ★



Volume 1 Number 4

In This Issue of CREATIVE COMPUTING

ARTICLES

PERSONALITY

THINGS TO DO

COMPUTER GAMES

FOOLISHNESS

DEPARTMENTS

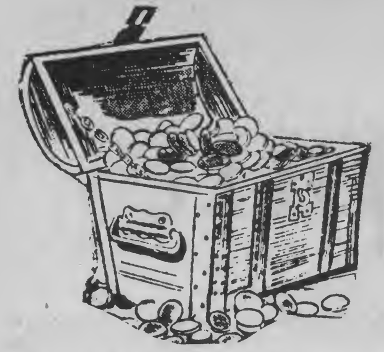
THE COVER

The cover of this issue is by Marilyn Clark and is titled "Lost World". The original medium is charcoal and our reproduction is approximately full size.

After graduating from Penn State with a BA in English, Marilyn worked for McGraw-Hill and SRI. For the last 10 years she has been a consultant/programmer in the Computer Center User Services group at the University of California, San Francisco. She is active with Friends of the Earth, the Sierra Club, Zero Population Growth and, in her own words, reads a "wide variety of left-wing and liberal magazines and newspapers."

She would like to do more art and writing, but she finds consulting with student programmers mind stretching and fulfilling. She eats nothing out of a can, is serious about the world, but has a sparkle in her eye, and enjoys life.

CREATIVE COMPUTING is published bi-monthly by Ideametrics, P. O. Box 789-M, Morristown, NJ 07960. Third 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. Printed in USA. Second class permit applied for at Morristown, NJ.



Earn Extra Money in Your Spare Time

READERS! We need more subscribers so we can financially break even, and then increase the size of *Creative Computing*. We can advertise but that costs more than it brings in. We can buy mailing lists and send out flyers, but good lists are hard to come by. There are other approaches, but the cost is generally higher than the payoff.

So we're coming to you readers, because we feel that your friends are the most likely new subscribers. We know you love us and you'd be happy to work for us for nothing, however, we're not asking that. We're going to pay you for your efforts. Here's the deal:

1. We'll pay you a 15% commission on the subscriptions you bring in.
2. Minimum of five subscriptions per order.
3. Payment must be enclosed with order (we can't possibly afford to send out bills on this deal.)

For example, say you get the following six subscriptions:

Type	*Cost*	*15% Commission*
1 Year Student	$ 6.00	$ .90
1 Year Individual	8.00	1.20
1 Year Library	15.00	2.25
3 Year Student	16.00	2.40
3 Year Individual	21.00	3.15
3 Year Library	40.00	6.00
TOTAL	$106.00	$15.90

You would send us six completed subscription blanks (or reasonable facsimilies), $106 in cash, check, or money order and your name and address. By return mail, you'll receive our check for $15.90.

Remember: don't send in fewer than 5 subscriptions, do enclose payment with the order, and do include your name and address. (One of the five subscriptions may be your own — new, renewal, or extension.)

CHANGE OF ADDRESS

Because of our new-fangled computerized system for keeping names and addresses of subscribers, we must have your OLD and NEW address in case of any change. Your CG number which appears above your name on the address label would help enormously too.

Zip Code. It is VITAL. All our files are kep in zip code order. If it's not on the label, your magazine could wind up destroyed (yes, destroyed, not returned to us) rather than delivered to you. Uncle Sam only has an obligation to deliver 3rd class mail if it has a zip code on the address. Don't forget ZIP!

☆☆

CREATIVE COMPUTING SUBSCRIPTION FORM

Published bi-monthly —— 6 issues per year

INSTITUTIONAL
- ☐ 1-year $15
- ☐ 3-year $40

INDIVIDUAL REGULAR
- ☐ 1-year $ 8
- ☐ 3-year $21

INDIVIDUAL STUDENT
- ☐ 1-year $ 6
- ☐ 3-year $16

FOREIGN POSTAGE
- ☐ $1.50 per year

BILLING INSTRUCTIONS
- ☐ Payment enclosed
- ☐ Please bill me (Receive 1 issue less)

Name ______________________ Title or Dept. ______________________

School or Company ____

Street Address ____

City ______________________ State ______________ Zip ____________

Return form to: CREATIVE COMPUTING, P.O. Box 789-M, Morristown, New Jersey 07960

Input/Output

Dear Editor:

I think you should stick to "standard" BASIC in programs that are included in *CREATIVE COMPUTING*. In volume 1, number 2, there were programs on pages 12, 13, and 19 that use the backslash for multiple statements on a line. The one on page 19 also has some construction that looks like Fortran implied do loops in a print line and if-then-else with statements allowed as arguments. It is honestly not BASIC and will probably only run on the machine that originated it. The use of an output string in an input statement (e.g. INPUT "YOUR MESSAGE PLEASE" A$) is also nonstandard. Sorry to push the point so hard, particularly on one of your own programs, but I think that programming style is pretty important, especially in publications that lots of people are going to see. The language you choose is an important part of style, and encouraging weird extensions that don't conform to the spirit of a language is poor style.

Christopher G. Hoogendyk
Dartmouth College

I agree with you in spirit; however, when a significant or interesting program is submitted to us (for example, SUPER STAR TREK in this issue), should we not publish it because it is not in standard BASIC? Or should we require the submitter to convert it to "standard" BASIC (to which most contributors would reply, "Why should I bother?"). Or should we convert it to standard BASIC (at which request, most of our volunteer editors would find other things to do). Or should we publish it and leave it as an exercise for readers to convert?

READERS: What do you think?

Parting note: to my knowledge, the BASIC Standards Committee has not yet defined "standard" BASIC. – DHA.

Dear Editor:

The Summer 1975 Conference of ADCIS, the Association for the Development of Computer-Based Instructional Systems, will take place on August 5, 6, and 7. The host institution will be the University of Maine Portland-Gorham. Further and more detailed information about the Program and the Call for Participation may be obtained from:

Dr. Martin Kamp, Info. Systems
San Francisco Medical Center
Unviersity of California
San Francisco, CA 94143

Alan G. Smith

Dear Editor:

As the former Systems Programmer for the Long Island Regional Instructional Computer Services (LIRICS) I am well aware of the problem of students compromising software security. (Editorial, Jan-Feb 1975) At LIRICS we had to deal with all three types of security breachers described by Mr. Tagg in over 60 school districts.

Students who discovered ways to breach system security and reported it to me without using it were thanked. Such students saved many man-hours of blind searching which would have been required had a malicious user discovered the problem.

Students that discovered but did not use or report problems were ignored since we could not track them down anyway.

Students who used holes in the software security to disrupt our operation in any manner were attacked from two directions. Management attacked with a seek and destroy type inquiry while software people attacked with rigged controls and monitoring.

We believe our method of dealing with these students was successful. I would suggest that educational institutions encourage experimentation but attack malicious students with a determined and sneaky software specialist.

Harold R. Berenson
Syosset, New York

Dear Editor:

Thanks again for a superb issue of *Creative Computing*. I'm sure that you can't keep up the pace on improvement because there just isn't that much room to go. This is the best journal for my purposes that I have ever seen.

There is one thing that I'd like to point out in relation to that historical reprint from IBM [Digital Calculators – Then and Now. Jan-Feb 1975]. In an article called "Will the Inventor of the First Digital Computer Please Stand Up?" W. David Gardner reports on the work of Dr. John Vincent Atanasoff for *Datamation* (Feb., 1974, pp. 84-90). The article gives the decision of Federal District Court Judge Earl R. Larson which "defrocked Dr. J. Presper Eckert and Dr. John W. Mauchly as the high priests of electronic digital computer invention." It goes on to explain how the decision arose in a case involving Sperry Rand and Honeywell over the patent of ENIAC. After carefully considering the evidence, Judge Larson decided that the patent was invalid because the basic ideas were taken from a machine which Atanasoff developed between 1935 and 1942 at the University of Iowa. Atanasoff has gone without proper credit long enough (and besides too many people have the idea that nothing important but agriculture happens out here on the plains).

Paul J. Emmerich
Dana College

T-Shirt Design Contest
Win BIG $!

Just imagine the startled looks on the faces of your friends when you show up at MacDonald's wearing your *Creative Computing* sweatshirt. Wow! Ducking into a nearby phone booth you remove it to reveal your *Creative Computing* T-shirt underneath. Double Wow!!

Just imagine our problem trying to find a sweatshirt/T-shirt design that reflects our light-hearted approach to computers. What should it be? A computer monster? A monster computer? A hopped-up computer hot rod? A computer on a high? A computer playing games (heh, heh, heh)? A creative computer creating a creative computer? A creating computer creating computer create – bleep-bleep-bleep – [The editor creating this announcement just blew an IC so I'll finish – The Other Editor]

Creative Computing will pay $25 (yes, twenty-five dollars) for the best T-shirt design submitted to us by July 18, 1975. Runners up will get $10 prizes! Designs should be black and white, ink or paint (not pencil or charcoal), and approximately square. The final design will be reduced to about 10" x 10" so if you do it larger, remember that detail will be lost in the reduction. Entries cannot be returned. Entries will be acknowledged only if accompanied by a self-addressed stamped envelope. If you want your entry kept flat, mark the outside envelope with LARGE letters "DO NOT FOLD OR BEND". Otherwise the Morristown Post Office will manage to put in our 3" x 5" box.

DON'T WAIT. DO IT TODAY. And then send your design or designs to:

T-Shirt Contest
Creative Computing
P.O. Box 789-M
Morristown, New Jersey 07960

CREATIVE COMPUTING Editorial

Where Are We Going?

One evening last July, I was lying on my back on the esplanade next to the Charles River. The strains of a Mozart piano concerto by the Boston Pops filled the air. The night was particularly clear and the stars in the moonless sky were glittering like diamonds on a piece of black velvet. Counting the stars that could be seen with the naked eye in even one small sector of the sky would have been an absolute impossibility.

The entire vista recalled to mind a statement from the Project Cyclops* report. It stated that scores of other intelligent races are radiating communication signals that can be received on Earth. This should come as no surprise. Given the incredibly vast numbers of galaxy clusters, galaxies, stars and solar systems, the probability of other races of superior intelligence or in a considerably advanced evolutionary state to humans is literally astronomical.

Entering the realm of speculation, it is quite possible that we on Earth have been observed by another race and ignored much as you or I might view a slug or an ant. The number of cells in the brain of a slug indicate that it has virtually no intelligence whatsoever and who in their right mind would want to communicate with such a hideous, slimy creature anyway? Is there any reason to believe that we humans haven't been examined by some extraterrestrial race (the Bermuda Triangle disappearances?) and then been discarded as too barbaric and dumb to communicate with and too revolting and ugly to worry about anyway?

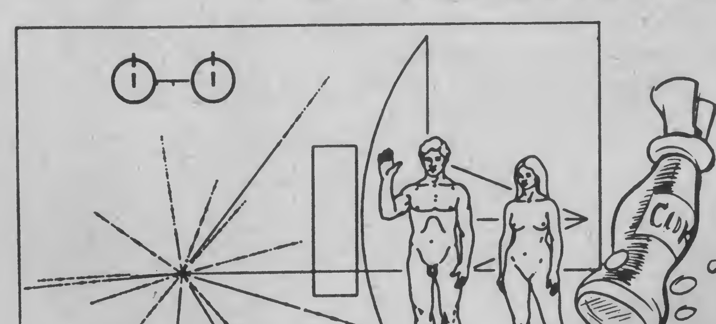
Consider the human species for a moment. Over the some odd thousand years of recorded history, the human race has made enormous gains in technology. No question about that. It's curious, however, that there have been virtually no corresponding gains in interpersonal relationships. Nations still war with one another as do states, neighbors, and husbands and wives.

To an extraterrestrial, our advanced technology probably looks like cute little toys for the human ants to play with. But fundamentally, all the technology of the past 2000 years hasn't altered the native intelligence and conduct of the species one iota. We still war with one another and lash out at those closest to us. Will computers help? What can we expect from our association with these logical companions? I have to believe that computers give us more potential to extend our intellect than we've ever had before. However, if we take history into account, I can't be overly sanguine about the outcome.

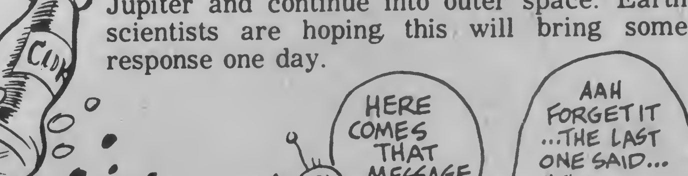
If there's a message in all of this I guess it's got to be: "STOP and THINK for a minute. What are the goals of the human race? Where should we be going as a species? Are you, you personally, contributing to that goal? In the 24 hours just past, what percentage of the time were you really using your intelligence to the fullest and what percentage were you a human ant responding to the myriad pressures of job, family, school, or society? Why not take 10 minutes a week (one-tenth of one percent of the time in a week) to ponder humankind on a galactic scale? You might be surprised at the outcome!"

David Ahl

*Project Cyclops was an intensive 3-month study at Stanford to recommend an approach to search for extraterrestrial life.



An aluminum plate was placed aboard the spacecraft "Pioneer 10" launched in March 1972. On it is etched a message designed to tell any interstellar finder that there is intelligent life on this planet. "Pioneer 10" is destined to fly-by Jupiter and continue into outer space. Earth scientists are hoping this will bring some response one day.



Quite Challenging
Mike T.

Another new game from Creative Computing . . .

SPLAT

OPEN A PARACHUTE AT THE LAST MOMENT

by John F. Yegge
Oak Ridge Associated Universities

SPLAT simulates a parachute jump in which you try to open your parachute at the last possible moment without going splat! You may select your own terminal velocity or let the computer do it for you. You may also select the acceleration due to gravity or, again, let the computer do it in which case you might wind up on any of the eight planets (out to Neptune), the moon, or sun.

The computer then tells you the height you're jumping from and asks for the seconds of free fall. It then divides your free fall time into eight intervals and gives you progress reports on your way down. The computer also keeps track of all prior jumps in the file PARACHUTE and lets you know how you compared with previous successful jumps. You can easily convert SPLAT to your version of BASIC by eliminating the file which keeps track of previous jumps although if you have file capabilities, use them — the game is that much more fun!

```
WELCOME TO 'SPLAT' -- THE GAME THAT SIMULATES A PARACHUTE
JUMP.  TRY TO OPEN YOUR CHUTE AT THE LAST POSSIBLE
MOMENT WITHOUT GOING SPLAT.


 SELECT YOUR OWN TERMINAL VELOCITY (YES OR NO)? NO
OK.  TERMINAL VELOCITY = 796 MI/HR
WANT TO SELECT ACCELERATION DUE TO GRAVITY (YES OR NO)? NO
FINE. YOU'RE ON MERCURY. ACCELERATION=12.2FT/SEC/SEC

    ALTITUDE          = 9297 FT
    TERM.VELOCITY     = 1167.47 FT/SEC +-5%
    ACCELERATION      = 12.2 FT/SEC/SEC +-5%
SET THE TIMER FOR YOUR FREEFALL.
HOW MANY SECONDS? 8
HERE WE GO.

TIME (SEC)     DIST TO FALL (FT)
==========     =================
 0              9297
 1              9290.88
 2              9272.51
 3              9241.89
 4              9199.02
 5              9143.91
 6              9076.55
 7              8996.94
 8              8905.09
CHUTE OPEN
CONSERVATIVE AREN'T YOU?  YOU RANKED ONLY 9 IN THE
 14 SUCCESSFUL JUMPS BEFORE YOURS.
DO YOU WANT TO PLAY AGAIN? YES


 SELECT YOUR OWN TERMINAL VELOCITY (YES OR NO)? NO
OK.  TERMINAL VELOCITY = 740 MI/HR
WANT TO SELECT ACCELERATION DUE TO GRAVITY (YES OR NO)? NO
THEN YOU'RE ON NEPTUNE. ACCELERATION=39.6FT/SEC/SEC

    ALTITUDE          = 5189 FT
    TERM.VELOCITY     = 1085.33 FT/SEC +-5%
    ACCELERATION      = 39.6 FT/SEC/SEC +-5%
SET THE TIMER FOR YOUR FREEFALL.
HOW MANY SECONDS? 18
HERE WE GO.

TIME (SEC)     DIST TO FALL (FT)
==========     =================
 0              5189
 2.25           5089.74
 4.5            4791.95
 6.75           4295.63
 9              3600.78
 11.25          2707.4
 13.5           1615.5
 15.75          325.072
 16.2678        SPLAT
MAY THE ANGEL OF HEAVEN LEAD YOU INTO PARADISE
I'LL GIVE YOU ANOTHER CHANCE.
DO YOU WANT TO PLAY AGAIN? YES
```

```
30 OPEN "PARACH,UTE" AS FILE 1%
40 DIM #1%,A(4000)
55 RANDOMIZE
95 PRINT "WELCOME TO 'SPLAT' -- THE GAME THAT SIMULATES A PARACHUTE"
96 PRINT "JUMP.  TRY TO OPEN YOUR CHUTE AT THE LAST POSSIBLE"
97 PRINT "MOMENT WITHOUT GOING SPLAT."
118 PRINT\PRINT\D1=0\V=0\A=0\N=0\M=0\D1=INT(9001*RND(1)+1000)
119 PRINT " SELECT YOUR OWN TERMINAL VELOCITY (YES OR NO)";\INPUT A1$
120 IF A1$="NO" THEN 128 ELSE IF A1$="YES" THEN 123
121 PRINT "'YES' OR 'NO' PLEASE";\INPUT A1$\GOTO 120
123 PRINT "WHAT TERMINAL VELOCITY (MI/HR)";\INPUT V1
125 V1=V1*(5280/3600)\V=V1+((V1*RND(0))/20)-((V1*RND(0))/20)\GOTO 135
128 V1=INT(1000*RND(0))
130 PRINT "OK,  TERMINAL VELOCITY ="V1"MI/HR"
131 V1=V1*(5280/3600)\V=V1+((V1*RND(0))/20)-((V1*RND(0))/20)
135  PRINT "WANT TO SELECT ACCELERATION DUE TO GRAVITY (YES OR NO)";
136 INPUT B1$
140 IF B1$="NO" THEN 150 ELSE IF B1$="YES" THEN 143
141 PRINT "'YES' OR 'NO' PLEASE";\INPUT B1$\GOTO 140
143 PRINT "WHAT ACCELERATION (FT/SEC/SEC)";\INPUT A2
145 A=A2+((A2*RND(0))/20)-((A2*RND(0))/20)\GOTO 205
150 ON INT(1+(10*RND(0)))GOTO151,152,153,154,155,156,157,158,159,160
151 PRINT"FINE. YOU'RE ON MERCURY. ACCELERATION=12.2FT/SEC/SEC"\GOTO161
152 PRINT"ALRIGHT. YOU'RE ON VENUS. ACCELERATION=28.3 FT/SEC/SEC"\GOTO162
153 PRINT "THEN YOU'RE ON EARTH. ACCELERATION=32.16 FT/SEC/SEC"\GOTO 163
154 PRINT"FINE. YOU'RE ON THE MOON. ACCELERATION=5.15FT/SEC/SEC"\GOTO 164
155 PRINT"ALRIGHT. YOU'RE ON MARS. ACCELERATION=12.5FT/SEC/SEC"\GOTO 165
156 PRINT"THEN YOU'RE ON JUPITER. ACCELERATION=85.2FT/SEC/SEC"\GOTO 166
157 PRINT"FINE. YOU'RE ON SATURN. ACCELERATION=37.6FT/SEC/SEC"\GOTO 167
158 PRINT"ALRIGHT. YOU'RE ON URANUS. ACCELERATION=33.8FT/SEC/SEC"\GOTO 168
159 PRINT"THEN YOU'RE ON NEPTUNE. ACCELERATION=39.6FT/SEC/SEC"\GOTO 169
160 PRINT"FINE. YOU'RE ON THE SUN. ACCELERATION=896FT/SEC/SEC"\GOTO 170
161 A2=12.2\GOTO 145
162 A2=28.3\GOTO 145
163 A2=32.16\GOTO 145
164 A2=5.15\GOTO 145
165 A2=12.5\GOTO 145
166 A2=85.2\GOTO 145
167 A2=37.6\GOTO 145
168 A2=33.8 \GOTO 145
169 A2=39.6\GOTO 145
170 A2=896\GOTO 145
205 PRINT
206 PRINT "    ALTITUDE          ="D1"FT"
207 PRINT "    TERM.VELOCITY     ="V1"FT/SEC +-5%"
208 PRINT "    ACCELERATION      ="A2"FT/SEC/SEC +-5%"
210 PRINT "SET THE TIMER FOR YOUR FREEFALL."
211 PRINT "HOW MANY SECONDS";\INPUT T
215 PRINT "HERE WE GO."
217 PRINT
218 PRINT "TIME (SEC)","DIST TO FALL (FT)"
219 PRINT "==========","================="
300 FOR I=0 TO T STEP (T/8)
310 IF I>V/A GOTO 400
320 D=D1-((A/2)*I^2)
330 IF D<=0 GOTO 1000
340 PRINT I,D
350 NEXT I
360 GOTO 500
400 PRINT "TERMINAL VELOCITY REACHED AT T PLUS"V/A"SECONDS"
405 FOR I=I TO T STEP (T/8)
410 D=D1-((V^2/(2*A))+(V*(I-(V/A))))
420 IF D<=0 GOTO 1010
430 PRINT I,D
440 NEXT I
450 GOTO 500
500 PRINT "CHUTE OPEN"
510 K=0\K1=0
530 FOR I=1 TO 4000
550 IF A(I)=0 GOTO 620
560 K=K+1
570 IF D>=A(I) GOTO 600
580 K1=K1+1
600 NEXT I
620 A(I)=D
630 I=I-1
650 IF K-K1<=.1*K GOTO 700
660 IF K-K1<=.25*K GOTO 710
670 IF K-K1<=.5*K GOTO 720
680 IF K-K1<=.75*K GOTO 730
690 IF K-K1<=.9*K GOTO 740
695 GOTO 750
700 PRINT "WOW!  THAT'S SOME JUMPING.  OF THE"K"SUCCESSFUL JUMPS"
701 PRINT "BEFORE YOURS, ONLY"K-K1"OPENED THEIR CHUTES LOWER THAN"
702 PRINT "YOU DID."
703 GOTO 2000
710 PRINT "PRETTY GOOD! " K"SUCCESSFUL JUMPS PRECEDED YOURS AND ONLY"
711 PRINT K-K1" OF THEM GOT LOWER THAN YOU DID BEFORE THEIR CHUTES"
712 PRINT "OPENED." \GOTO 2000
720 PRINT "NOT BAD.  THERE HAVE BEEN"K"SUCCESSFUL JUMPS BEFORE YOURS."
721 PRINT"YOU WERE BEATEN OUT BY"K-K1"OF THEM."\GOTO 2000
730 PRINT "CONSERVATIVE AREN'T YOU?  YOU RANKED ONLY"K-K1"IN THE"
731 PRINT K"SUCCESSFUL JUMPS BEFORE YOURS."\GOTO 2000
740 PRINT "HUMPH!  DON'T YOU HAVE ANY SPORTING BLOOD?  THERE WERE"
741 PRINT K"SUCCESSFUL JUMPS BEFORE YOURS AND YOU CAME IN"K1"JUMPS"
742 PRINT "BETTER THAN THE WORST.  SHAPE UP!!!"\GOTO 2000
750 PRINT "HEY!  YOU PULLED THE RIP CORD MUCH TOO SOON.  "K"SUCCESSFUL"
751 PRINT "JUMPS BEFORE YOURS AND YOU CAME IN NUMBER"K-K1"!  GET WITH IT!"
752 GOTO 2000
800 PRINT "REQUIESCAT IN PACE."\GOTO 1950
801 PRINT "MAY THE ANGEL OF HEAVEN LEAD YOU INTO PARADISE"\GOTO 1950
802 PRINT "REST IN PEACE"\GOTO 1950
803 PRINT "SON-OF-A-GUN"\GOTO 1950
804 PRINT "#$%&&%!$"\GOTO 1950
805 PRINT "A KICK IN THE PANTS IS A BOOST IF YOU'RE HEADED RIGHT"\GOTO 1950
806 PRINT "HMMM. SHOULD HAVE PICKED A SHORTER TIME."\GOTO 1950
807 PRINT "MUTTER. MUTTER. MUTTER."\GOTO 1950
808 PRINT "PUSHING UP DAISIES."\GOTO1950
809 PRINT "EASY COME, EASY GO."\GOTO 1950
1000 PRINT SQR(2*D1/A),"SPLAT"
1005 ON INT(1+(10*RND(0)))GOTO 800,801,802,803,804,805,806,807,808,809
1010 PRINT (V/A)+((D1-(V^2/(2*A)))/V),"SPLAT"
1020 GOTO 1005
1950 PRINT "I'LL GIVE YOU ANOTHER CHANCE."\GOTO 2000
2000 PRINT "DO YOU WANT TO PLAY AGAIN";\INPUT Z$
2001 IF Z$="YES" GOTO 118
2002 IF Z$="NO" GOTO 2005
2003 PRINT "YES OR NO"\GOTO 2000
2005 PRINT "PLEASE";\INPUT Z$\IF Z$="YES" THEN 118 ELSE 2007
2007 PRINT "SSSSSSSSSS."\GOTO 2046
2046 CLOSE 1%
9999 END
```

Another new game from Creative Computing . . .

GEOWAR

No writeup was included with this game except, of course, the instructions and remarks in the listing. The idea and approach came from Kenneth Janowiak, a teacher at St. Patrick HS, Chicago. Programming was done by Gary Lorenc, a former student. (See the review of GEOWAR on the facing page.)

PROGRAM LISTING

```
1 REM *** GEOWAR
2 REM *** WRITTEN BY GARY LORENC, IDEA BY KENNETH JANOWIAK
3 REM *** ST. PATRICK HS, 5900 W. BELMONT AVE., CHICAGO, ILL
4 REM *** UPDATED TO BASIC-PLUS BY DAVE AHL
5 PRINT"DO YOU WANT A DESCRIPTION OF THE GAME? (1=YES,0=NO) ";
6 INPUT I
7 IF I=0 THEN 46
8 PRINT
9 PRINT"     THE FIRST QUADRANT OF A REGULAR COORDINATE GRAPH WILL";
10 PRINT" SERVE AS"
11 PRINT"THE BATTLEFIELD.  FIVE ENEMY INSTALLATIONS ARE LOCATED ";
12 PRINT"WITHIN A"
13 PRINT"30 BY 30 UNIT AREA.  NO TARGET IS INSIDE THE 10 BY 10 ";
14 PRINT"UNIT AREA"
15 PRINT"ADJACENT TO THE ORIGIN, AS THIS IS THE LOCATION OF OUR ";
16 PRINT"BASE.  WHEN"
17 PRINT"THE MACHINE ASKS FOR THE DEGREE OF THE SHOT, RESPOND ";
18 PRINT"WITH A NUMBER"
19 PRINT"BETWEEN 1 AND 90."
20 PRINT
21 PRINT TAB(51),"SCARE**********"
22 PRINT"     1. A DIRECT HIT IS A HIT WITHIN 1 DEGREE OF";
23 PRINT TAB(51),"*            *"
24 PRINT"          THE TARGET.",TAB(51),"*  HIT******  *"
25 PRINT"     2. A HIT MUST PASS BETWEEN THE FIRST SET OF";
26 PRINT TAB(51),"*  *       *  *"
27 PRINT"          INTEGRAL POINTS NW AND SE OF THE TARGET.";
28 PRINT TAB(51),"*  *   D   *  *"
29 PRINT"     3. A SCARE MUST PASS BETWEEN THE NEXT SET OF";
30 PRINT TAB(51),"*  *       *  *"
31 PRINT"          INTEGRAL POINTS NW AND SE OF THE TARGET,";
32 PRINT TAB(51),"*  ******HIT  *"
33 PRINT"          AND CAUSES THE ENEMY TO RELOCATE A ";
34 PRINT TAB(51),"*            *"
35 PRINT"          MAXIMUM OF 1 UNIT IN ANY DIRECTION.";
36 PRINT TAB(51),"**********SCARE"
37 PRINT
38 PRINT
39 PRINT"     MISSILES HAVE INFINITE RANGE AND MAY HIT MORE THAN ";
40 PRINT"ONE TARGET."
41 PRINT"A MISSILE THAT NEARLY MISSES AN INSTALLATION (A SCARE) ";
42 PRINT"WILL BE"
43 PRINT"IMMEDIATELY SHOT DOWN.  ANY HITS BEFORE THIS TIME WILL ";
44 PRINT"NOT BE COUNTED"
45 PRINT"UNLESS A DIRECT HIT WAS MADE."
46 PRINT
47 PRINT
48 PRINT"READY TO GO? (1=YES,0=NO) ";
49 INPUT R2
50 IF R2=0 THEN 192
51 PRINT"GOOD LUCK!"
52 PRINT
53 DIM C(10),H(20),D(10),S(20),F(5)
54 DEF FNV(V1)=INT((180/3.14159)*ATN(V1)+.5)
55 X=250
55 X1=RND(X)
57 G2=0
58 S2=0
59 D2=0
60 H2=0
61 FOR K=1 TO 10
62 GO SUB 154
63 IF INT(K/2)<>K/2 THEN 70
64 IF C(K-1)>10 THEN 70
65 IF C(K)>10 THEN 70
66 FOR L=K-1 TO K
67 GOSUB 154
68 NEXT L
69 GO TO 63
70 NEXT K
71 S=0
72 FOR L=1 TO 5
73 D(L)=FNV(C(2*L)/C(2*L-1))
74 NEXT L
75 A=2
76 L1=10
77 T5=5
78 D5=0
79 H5=0
80 GO SUB 147
81 PRINT
82 PRINT"ENTER DEGREE OF SHOT ";
83 D1=0
84 H1=0
85 FOR Q=1 TO 5
86 F(Q)=20
87 NEXT Q
88 INPUT D
89 IF D>=90 THEN 81
90 IF D<0 THEN 177 ELSE IF D=0 THEN 192
91 S=S+1
92 FOR A=2 TO 10 STEP 2
93 IF D>S(A) THEN 103
94 IF D<S(A-1) THEN 103
95 IF D>H(A) THEN 105
96 IF D<H(A-1) THEN 105
97 IF D>D(A/2)+1 THEN 101
98 IF D<D(A/2)-1 THEN 101
99 D1=D1+1
100 GO TO 102
101 H1=H1+1
102 F(D1+H1)=A
103 NEXT A
104 GO TO 108
105 IF D1>0 THEN 118
106 GO SUB 138
107 GO TO 81
108 IF D1+H1<>0 THEN 112
109 IF T5=1 THEN 159
110 PRINT"NO LUCK -- TRY AGAIN."
111 GO TO 81
112 IF D1>0 THEN 118
113 IF H1>1 THEN 116
114 PRINT"**CONGRATULATIONS**   A HIT."
115 GO TO 124
116 PRINT"**CONGRATULATIONS**";H1;" HITS."
117 GO TO 124
118 PRINT"****BULLS EYE**** ";
119 IF D1>1 THEN 123
120 IF H1>0 THEN 123
121 PRINT" A DIRECT HIT!"
122 GO TO 124
123 PRINT D1+H1;" HITS -- A DIRECT HIT ON";D1;" OF THEM!"
124 T5=T5-(D1+H1)
125 D5=D5+D1
126 H5=H5+H1
127 IF T5=0 THEN 167
128 FOR J= 1 TO H1+D1
129 Z=F(J)
130 D(Z/2)=0
131 H(Z)=0
132 H(Z-1)=0
133 S(Z)=0
134 S(Z-1)=0
135 NEXT J
136 PRINT 5-T5;" DOWN --";T5;" TO GO."
137 GO TO 81
138 PRINT"A NEAR HIT.  ENEMY HAS RELOCATED."
139 FOR R=1 TO 2
140 X2=INT(RND(0)*100)
141 IF ABS(C(A-(R-1))-X2)>1 THEN 140
142 IF C(A-(R-1))<=2 THEN 140
143 C(A-(R-1))=X2
144 NEXT R
145 D(A/2)=FNV(C(A)/C(A-1))
146 L1=A
147 FOR I=A TO L1 STEP 2
148 H(I-1)=FNV((C(I)-1)/(C(I-1)+1))
149 H(I)=FNV((C(I)+1)/(C(I-1)-1))
150 S(I-1)=FNV((C(I)-2)/(C(I-1)+2))
151 S(I)=FNV((C(I)+2)/(C(I-1)-2))
152 NEXT I
153 RETURN
154 R=INT(RND(0)*100)
155 IF R>30 THEN 154
156 IF R<3 THEN 154
157 C(K)=R
158 RETURN
159 FOR Z1=1 TO 5
160 IF D(Z1)>1 THEN 162
161 NEXT Z1
162 IF D<D(Z1) THEN 165
163 PRINT"TOO HIGH -- TRY AGAIN."
164 GO TO 81
165 PRINT"TOO LOW -- TRY AGAIN."
166 GO TO 81
167 PRINT
168 PRINT"GAME TOTALS:";H5;" HITS AND";D5;" DIRECT HITS ON";S;"SHOTS."
169 PRINT
170 PRINT"READY FOR A NEW GAME? (1=YES,0=NO) ";
171 G2=G2+1
172 S2=S2+S
173 D2=D2+D5
174 H2=H2+H5
175 INPUT G
176 IF G=0 THEN 184
177 PRINT
178 PRINT
179 PRINT
180 PRINT"FIVE NEW INSTALLATIONS HAVE BEEN BUILT AT DIFFERENT ";
181 PRINT"LOCATIONS."
182 PRINT"GOOD LUCK!"
183 GO TO 61
184 PRINT
185 PRINT
186 PRINT "TOTALS FOR"G2"GAMES;"H2"HITS AND"D2
187 PRINT "DIRECT HITS ON"S2"SHOTS."
188 PRINT "AN AVERAGE OF"S2/(D2+H2)"SHOTS PER TARGET."
192 END
```

SAMPLE RUN

```
RUN

DO YOU WANT A DESCRIPTION OF THE GAME? (1-YES,0-NO) !1

     THE FIRST QUADRANT OF A REGULAR COORDINATE GRAPH WILL SERVE AS
THE BATTLEFIELD.  FIVE ENEMY INSTALLATIONS ARE LOCATED WITHIN A
30 BY 30 UNIT AREA.  NO TARGET IS INSIDE THE 10 BY 10 UNIT AREA
ADJACENT TO THE ORIGIN, AS THIS IS THE LOCATION OF OUR BASE.  WHEN
THE MACHINE ASKS FOR THE DEGREE OF THE SHOT, RESPOND WITH A NUMBER
BETWEEN 1 AND 90.

                                                  SCARE**********
     1. A DIRECT HIT IS A HIT WITHIN 1 DEGREE OF  *            *
          THE TARGET.                             *  HIT******  *
     2. A HIT MUST PASS BETWEEN THE FIRST SET OF  *  *       *  *
          INTEGRAL POINTS NW AND SE OF THE TARGET.*  *   D   *  *
     3. A SCARE MUST PASS BETWEEN THE NEXT SET OF *  *       *  *
          INTEGRAL POINTS NW AND SE OF THE TARGET,*  ******HIT  *
          AND CAUSES THE ENEMY TO RELOCATE A      *            *
          MAXIMUM OF 1 UNIT IN ANY DIRECTION.     **********SCARE

     MISSILES HAVE INFINITE RANGE AND MAY HIT MORE THAN ONE TARGET.
A MISSILE THAT NEARLY MISSES AN INSTALLATION (A SCARE) WILL BE
IMMEDIATELY SHOT DOWN.  ANY HITS BEFORE THIS TIME WILL NOT BE COUNTED
UNLESS A DIRECT HIT WAS MADE.
```

```
READY TO GO? (1-YES,0-NØ) !1
GOOD LUCK!

ENTER DEGREE ØF SHØT !20
NØ LUCK -- TRY AGAIN.

ENTER DEGREE ØF SHØT !30
A NEAR HIT.  ENEMY HAS RELØCATED.

ENTER DEGREE ØF SHØT !31
****BULLS EYE****      3  HITS -- A DIRECT HIT ØN      2  ØF THEM!
     3  DØWN --      2  TØ GØ.
                                   MULTIPLE HITS
ENTER DEGREE ØF SHØT !40            ARE POSSIBLE
NØ LUCK -- TRY AGAIN.

ENTER DEGREE ØF SHØT !50
NØ LUCK -- TRY AGAIN.

ENTER DEGREE ØF SHØT !60
****BULLS EYE****      2  HITS -- A DIRECT HIT ØN      1  ØF THEM!

GAME TØTALS:      2  HITS AND      3  DIRECT HITS ØN      6 SHØTS.
                                    TOTALS FOR THE GAME
READY FØR A NEW GAME? (1-YES,0-NØ) !1  (HARD TO BEAT IT!)


FIVE NEW INSTALLATIØNS HAVE BEEN BUILT AT DIFFERENT LØCATIØNS.
GØØD LUCK!

ENTER DEGREE ØF SHØT !45
A NEAR HIT.  ENEMY HAS RELØCATED.

ENTER DEGREE ØF SHØT !44
A NEAR HIT.  ENEMY HAS RELØCATED.

ENTER DEGREE ØF SHØT !31
A NEAR HIT.  ENEMY HAS RELØCATED.

ENTER DEGREE ØF SHØT !30
****BULLS EYE****  A DIRECT HIT!
     2  DØWN --      3  TØ GØ.

ENTER DEGREE ØF SHØT !25
A NEAR HIT.  ENEMY HAS RELØCATED.

ENTER DEGREE ØF SHØT !26
A NEAR HIT.  ENEMY HAS RELØCATED.

ENTER DEGREE ØF SHØT !27
**CØNGRATULATIØNS**   A HIT.
     3  DØWN --      2  TØ GØ.

ENTER DEGREE ØF SHØT !65
**CØNGRATULATIØNS**   A HIT.
     4  DØWN --      1  TØ GØ.

ENTER DEGREE ØF SHØT !50     HELP GIVEN ON LAST SHOTS
TØØ LØW -- TRY AGAIN.         "TOO HIGH" OR "TOO LOW"

ENTER DEGREE ØF SHØT !70
TØØ LØW -- TRY AGAIN.

ENTER DEGREE ØF SHØT !80
**CØNGRATULATIØNS**   A HIT.

GAME TØTALS:      4  HITS AND      1  DIRECT HITS ØN     28 SHØTS.

READY FØR A NEW GAME? (1-YES,0-NØ) !0
                             TOTALS FOR ALL GAMES & AVERAGE
TØTALS FØR  2 GAMES:   6 HITS AND   4 DIRECT HITS ØN   34 SHØTS
                         AN AVERAGE ØF  3.40 SHØTS PER TARGET
```

REVIEW OF GEOWAR

by Gregory Yob

The editor of any publication has a dilemma. There's lots of material, but most of it is of low quality or presents the wrong viewpoint for his magazine. I am sure this is true of computer games as evidenced by the game of GEOWAR which was given to me by the editor of *Creative Computing*. As a dedicated games-lover, these comments are offered in the hopes for better games. In fact, please correspond with me if you share (or reject) my views.

Let's get down to business. GEOWAR is another of those "shoot the enemy with missiles (phasers, lasers, zap-beams, MIRVS, etc.)" games. In some ways I liked it; in most I didn't.

THE TECHNICAL LEVEL. Programming GEOWAR or an equivalent game requires a good knowledge of BASIC in many ways. Noted in the program were the uses of arrays, pointers, subroutines, library functions and a defined function. As a problem for a final exam (do the flowchart) GEOWAR is excellent. Writing and debugging GEOWAR is a fine term project for second-semester programming.

THE LEVEL OF CLARITY. The instructions for GEOWAR are muddled a bit. It took me two readings to understand the first quadrant instead of the full 360 degrees was the playing area. I offer an improved diagram of the playing area (there wasn't any, a mortal sin for tactical games) and a new version of the HIT-SCARE diagram (see figures). If you are to use a grid, show the points clearly!!! If your range of fire is limited, MAKE IT CLEAR!!! This is a usual case of pictures vs. kilowords.

```
30 UNITS- I   90 DEGREES
          I     E
          I  E
          I                  E
          I
          I               E
10 UNITS- I-------
          I      .
          I CLEAR'          E
          I ZONE '            0 DEGREES
     YOU---*===========================
               10                  30
              UNITS               UNITS
```

Field of Play

```
+SCARE+--+--+
.  .  .  .  .
+--HIT+==+--+
.  !  !  !  .
+--+--D--+--+
.  !  !  !  .
+--+==+==+--+
.  .  .  .  .
+--+--+--+--+
```

Scare/Hit Diagram

TUTORIAL LEVEL. What does GEOWAR teach? Possibly about angles . . . Mostly it is a game of guessing. Guessing strategies are very clearly done in STARS. In a sense, GEOWAR is a five-number version of STARS. Regarding angles, STAR TREK is much more effective and lots of fun! I notice the games authors live in Chicago. A visit to Urbana and a tour of the games on PLATO is well worth the effort. I particularly suggest MOONWAR, CONQUEST, NOVA and ROSE. MOONWAR is the most effective angle-teacher I have ever met. (All other games lovers should also try PLATO. Try DOGFIGHT!)

ESTHETIC AND PHILOSOPHICAL LEVEL. This is where I am most annoyed with GEOWAR. It's another hunt and kill game in an era where mutual co-operation in complex systems is a vital need. Missiles and cartesian grids are very common in computer games, and in writer's words, "the theme is a bit overdone". If we must teach of war, think about these situations:

a) an Army Artillery unit
b) a destroyer at sea
c) a jet in a dogfight
d) ICBMs (Minuteman, Polaris)

In each situation, the techniques and objectives differ. Hitting the target is only a small part of the game. Many neat games ideas can come of these situations viewed as part of a larger system, i.e., the artillery unit as part of supporting a commando unit.

WHAT I'D LIKE TO SEE:

1) Games using several players in different and mutually dependent roles.
2) Social, Economic and Ecological themes vs. War
3) An interesting field of play (as in HUNT THE WUMPUS) with variations and topography
4) The player's advantages to be the results of their actions. (the RND function is much over-used and often destroys the skill-learning aspects of a game)

Another new game from Creative Computing . . .

ICBM

by Paul Calter
Vermont Technical College

Your radar station picks up an enemy ICBM heading your way, telling you its coordinates (in miles north and miles east of your location). You launch a surface-to-air missile (SAM) to intercept it.

Your only control over the SAM is that you can aim it in any direction, both at launch, and in mid-air. Using the coordinates of the ICBM as a guide, you INPUT the direction (measured CCW from North) in which you want the SAM to travel.

At the next radar scan one minute later, you are given the new coordinates of the ICBM, the coordinates of your SAM, and the distance between the two. You can now make corrections in the course of your SAM by entering a new direction.

You have no control over the altitude of your SAM, as it is assumed that it will seek the same altitude as the ICBM.

As the two missiles draw closer, you make adjustments in the direction of the SAM so as to intercept the ICBM. It's not easy to hit, because the ICBM is programmed to make evasive maneuvers, by taking random deviations from the straight line course to your location. Also, its speed is not known, although it does not vary after being randomly selected at the start of the run.

You can destroy the ICBM by coming within 5 miles of it, at which time your SAM's heat-seeking sensors will come into action and direct it to its target. If you overshoot the ICBM it's possible to turn the SAM around and chase the ICBM back towards your location. But be careful; you may get both missiles in your lap.

There is also some element of chance involved, as several accidents have been programmed to occur randomly. These can work for you or against you.

Some ways to improve and expand the program are:

1. Operator control over SAM speed: In the present version the speed of the SAM is randomly selected by the computer at the start of the run, and remains constant thereafter. This often results in overshooting the ICBM. Modify the program so that you can input a new speed (within limits) at the same time you input the new direction.

2. Three dimensional version: Have the computer print the *altitude* of the ICBM, as well as its coordinates. The operator will then have to INPUT the angle his SAM is to make with the horizontal, when entering the other quantities.

3. Extend to all Quadrants. In the present version, the ICBM approaches only from the Northeast. You can expand this to include approach from any compass direction.

This game is derived from a program submitted by Chris Falco, of Glen Ridge High School, NJ.

PROGRAM LISTING

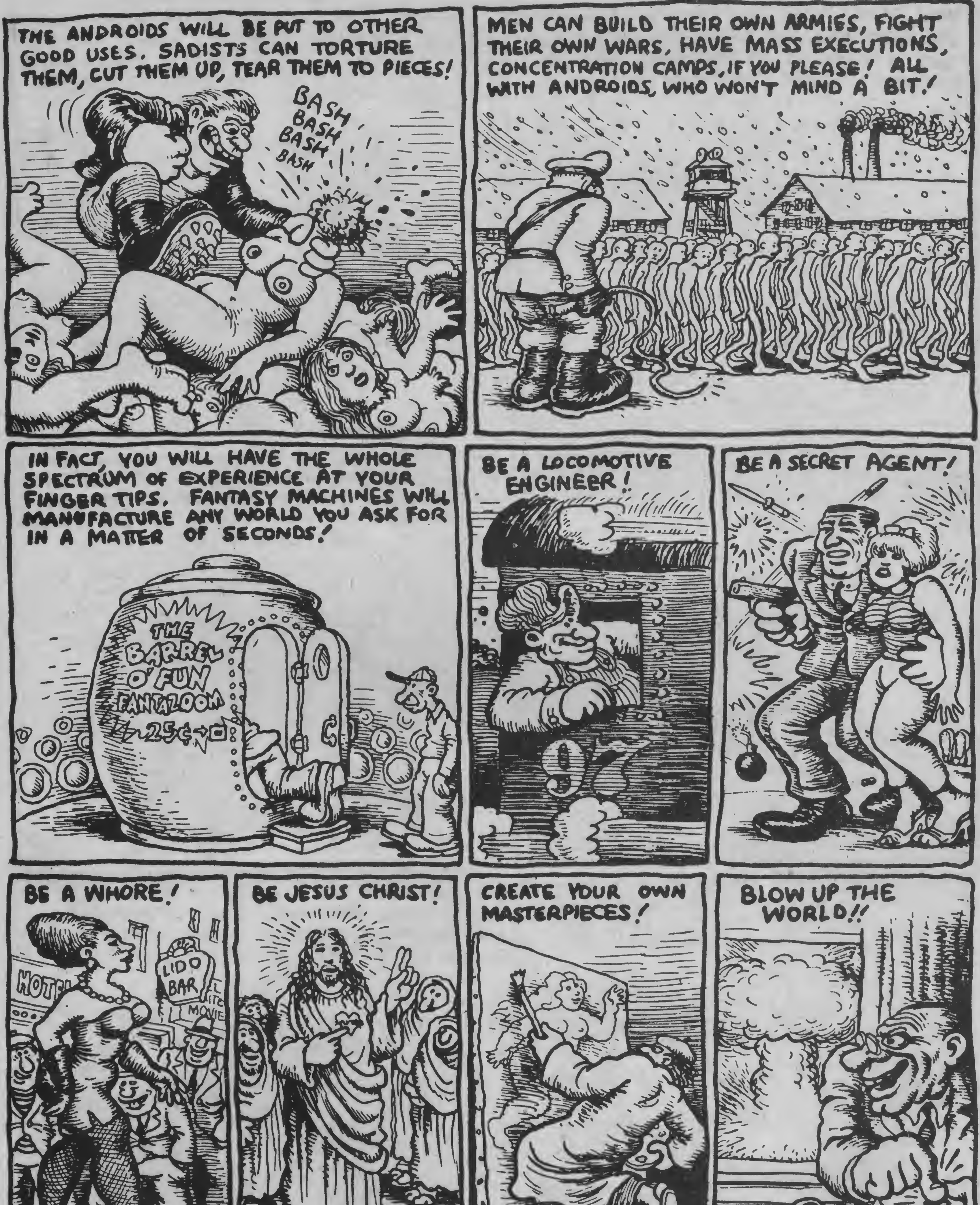
```
100 RANDOMIZE
110 LET X1=0
120 LET Y1=0
130 LET X= INT (RND*300)+200
140 LET Y=INT(RND*300)+200
150 LET S=INT(RND*20+50)
160 LET S1=INT(RND*20+50)
170 PRINT"----------MISSLE----------         -----------------SAM-------"
180 PRINT "MILES", "MILES", "MILES","MILES","HEADING"
190 PRINT "NORTH","EAST","NORTH","EAST","?"
200 PRINT "-------------------------------------------------------------"
210 FOR N=1 TO 50
220 PRINT Y, X, Y1, X1,
230 IF X=0 THEN 550
240 INPUT T1
250 LET T1=T1/57.296
260 LET H=INT(RND*200+1)
270 IF H>4 THEN 290
280 ON H GO TO 470,490,510,530
290 LET X1=INT(X1+S1*SIN(T1))
300 LET Y1=INT(Y1+S1*COS(T1))
310 IF SQR(X↑2+Y↑2)>S THEN 350
320 LET X=0
330 LET Y=0
340 GO TO 430
350 LET R=SQR(X↑2+Y↑2)/1000
360 LET T=ATN(Y/X)
370 LET X=INT(X-S*COS(T)+RND*20*R)
380 LET Y=INT(Y-S*SIN(T)+RND*20*R)
390 LET D=SQR((X-X1)↑2+(Y-Y1)↑2)
400 IF D=<5 THEN 440
410 LET D=INT(D)
420 PRINT "ICBM & SAM NOW"; D; " MILES APART"
430 NEXT N
440 PRINT "CONGRATULATIONS!  YOUR SAM CAME WITHIN";D;"MILES OF"
450 PRINT "THE ICBM AND DESTROYED IT."
460 GO TO 560
470 PRINT "TOO BAD.  YOUR SAM FELL TO THE GROUND"
480 GO TO 560
490 PRINT"YOUR SAM EXPLODED IN MIDAIR"
500 GO TO 560
510 PRINT "GOOD LUCK-THE ICBM EXPLODED HARMLESSLY IN MID-AIR"
520 GO TO 560
530 PRINT "GOOD LUCK-THE ICBM TURNED OUT TO BE A FRIENDLY AIRCRAFT"
540 GO TO 560
550 PRINT "TOO BAD! THE ICBM JUST HIT YOUR LOCATION"
560 PRINT"DO YOU WANT TO PLAY MORE?  (Y OR N)"
570 INPUT A$
580 IF A$="Y" THEN 130
590 END
READY
```

SAMPLE RUN

```
ICBM      08 JAN 75

----------MISSLE----------        -----------------SAM---------
MILES            MILES           MILES           MILES          HEADING
NORTH            EAST            NORTH           EAST           ?
-------------------------------------------------------------------
 587              868             0               0              ? 60
ICBM & SAM NOW 945  MILES APART
 565              832             29              50             ? 60
ICBM & SAM NOW 859  MILES APART
 554              832             58              100            ? 55
ICBM & SAM NOW 761  MILES APART
 525              773             91              147            ? 60
ICBM & SAM NOW 667  MILES APART
 512              737             120             197            ? 60
ICBM & SAM NOW 569  MILES APART
 493              701             149             247            ? 55
ICBM & SAM NOW 470  MILES APART
 467              668             152             294            ? 55
ICBM & SAM NOW 367  MILES APART
 436              635             215             341            ? X55
ICBM & SAM NOW 271  MILES APART
 420              598             248             388            ? 57
ICBM & SAM NOW 171  MILES APART
 392              565             279             436            ? 57
ICBM & SAM NOW 76  MILES APART
 374              526             310             484            ? 30
ICBM & SAM NOW 32  MILES APART
 346              483             360             512            ? 225
ICBM & SAM NOW 21  MILES APART
 320              449             318             470            ? 235
ICBM & SAM NOW 17  MILES APART
 295              408             284             422            ? 245
ICBM & SAM NOW 12  MILES APART
 271              369             259             369            ? 240
ICBM & SAM NOW 21  MILES APART
 242              335             229             318            ? 60
ICBM & SAM NOW 82  MILES APART
 213              299             258             368            ? 240
ICBM & SAM NOW 71  MILES APART
 182              262             228             317            ? 235
ICBM & SAM NOW 64  MILES APART
 155              218             194             269            ? 240
ICBM & SAM NOW 57  MILES APART
 126              175             164             218            ? 240
ICBM & SAM NOW 49  MILES APART
 96               135             134             167            ? 240
ICBM & SAM NOW 44  MILES APART
 67               92              104             116            ? 220
ICBM & SAM NOW 37  MILES APART
 35               49              59              78             ? 230
GOOD LUCK-THE ICBM TURNED OUT TO BE A FRIENDLY AIRCRAFT
DO YOU WANT TO PLAY MORE?  (Y OR N)
? N
```


CITY OF THE FUTURE
THERE'S A BIG NEW WONDERFUL TOMORROW FULL OF MONUMENTAL ACHIEVEMENTS!! UNHEARD OF VISTAS OF PROGRESS AWAIT! BETTER WORLDS ARE BEING BUILT! EVERYONE WILL HAVE ALL OF EVERYTHING! LET'S LOOK AT SOME OF THE THINGS WE'RE GOING TO GET IN THE YEARS TO COME!
LEVEL 52
346829
R.CRUMB


YOU WON'T HAVE TO SHIT ANYMORE! BOWELS WILL BE REMOVED AT BIRTH AND A SANITIZING DISPOSAL UNIT INSTALLED. NEEDS EMPTYING ONLY ONCE A MONTH. NO MORE WORRY ABOUT SMELLY EXCREMENT! GOOD-BYE TOILET!!
INCINERATOR
BUILDINGS, CARS WILL BE SOFT PLASTIC. STREETS WILL BE SOFT PLASTIC. ACCIDENTS WILL BE A THING OF THE PAST. NOBODY WILL GET HURT ANYMORE!
FOOF!
HAR
NO MORE HEAT AND COLD, NIGHT AND DAY. CITIES WILL HAVE ROOM TEMPERATURE ALL THE TIME. LIGHTING WILL BE SOFT, DIFFUSED. WARM SNOW FOR CHRISTMAS!!
EVERYONE WILL BE TUNED IN TO EVERYTHING THAT'S HAPPENING ALL THE TIME! NO-ONE WILL BE LEFT OUT. WE'LL ALL BE NORMAL!
FACTS FANCIES
ECONOMY
POLITICS
SCIENCE
ART
OPINIONS
IDEAS
TODAY'S MENU
GET FOOD HERE
NOBODY WILL WORK! ALL PRODUCTION, DISTRIBUTION AND MAINTENANCE WILL BE DONE BY COMPUTERIZED ROBOTS. PEOPLE CAN SPEND ALL OF THEIR TIME PLAYING, EATING, OR WATCHING TV!
...OR, THEY CAN *beep*!! SPECIAL *beep*-ING ANDROIDS WILL BE AVAILABLE TO EVERYONE! SOCIAL PROBLEMS WILL DISSAPPEAR. RISK OF INVOLVEMENT WITH THE OPPOSITE SEX WILL BE ELIMINATED!


THE ANDROIDS WILL BE PUT TO OTHER GOOD USES. SADISTS CAN TORTURE THEM, CUT THEM UP, TEAR THEM TO PIECES!
BASH BASH BASH BASH
MEN CAN BUILD THEIR OWN ARMIES, FIGHT THEIR OWN WARS, HAVE MASS EXECUTIONS, CONCENTRATION CAMPS, IF YOU PLEASE! ALL WITH ANDROIDS, WHO WON'T MIND A BIT!
IN FACT, YOU WILL HAVE THE WHOLE SPECTRUM OF EXPERIENCE AT YOUR FINGER TIPS. FANTASY MACHINES WILL MANUFACTURE ANY WORLD YOU ASK FOR IN A MATTER OF SECONDS!
THE BARREL O' FUN FANTAZOOM
25¢
BE A LOCOMOTIVE ENGINEER!
9
BE A SECRET AGENT!
BE A WHORE!
HOTEL
LIDO BAR
MOVIE
BE JESUS CHRIST!
CREATE YOUR OWN MASTERPIECES!
BLOW UP THE WORLD!!
PRESIDENT OF THE WORLD

JUST TO KEEP US ON OUR TOES, VAST ENTERTAINMENT NETWORKS WILL BE ORGANIZED THAT SPECIALIZE IN SURPRISE PRANKS! PEOPLE WILL GET TRIPPED!
SPLAT
CLOWNS WILL APPEAR OUT OF NOWHERE WITH SELTZER BOTTLES!
HAW
HOUSES WILL GET UP AND WALK AROUND! TREES WILL MAKE FACES! PEEL AN ORANGE AND SOCKO!
SOME OTHER ADVANCES: CLOCKS THAT YOU CAN HAVE PUT INSIDE YOUR HEAD SO THAT YOU'LL ALWAYS KNOW EXACTLY WHAT TIME IT IS!
THE TIME IS NOW 9:30!
BABY SITTING WILL BE DONE BY ROBOTS WITH TV HEADS THAT PLAY VIDEO TAPES OF MOM AND DAD. PARENTS WILL NO LONGER BE TIED TO THEIR CHILDREN!
SUCH CUTE LITTLE DARLINGS!
HEY GIMME A NICKEL MOM!
ME SOME TOO WOW
MANY NEW SPORTS WILL BE INVENTED! KIDS WILL RISK THEIR LIVES IN DANGEROUS ROCKET DRAG RACES. SOME OF THESE JOBS WILL GO AS FAST AS 25,000 MPH!!


THERE'LL ALWAYS BE THE SEARCH FOR THE BIGGER KICK! GUYS WILL PLAY "CHICKEN" WITH SUICIDAL SOUPED-UP BODY ROCKETS!
OTHERS WILL JUST SIT AROUND ALL DAY PLAYING MIND GAMES!
ONCE A YEAR ALL THE OLD STUFF WILL BE GATHERED UP AND PUT INTO HUGE MACHINES WHICH WILL GRIND IT UP AND MAKE IT INTO NEW STUFF!
THE BED AS WE KNOW IT WILL BE REPLACED BY A SOFT, WARM, MOIST FOAM PLASTIC BLOB THAT YOU JUST DIVE INTO AND FALL ASLEEP WHILE IT UNDULATES SLOWLY IN AND OUT AND SOOTHING, SWEET MUSIC PLAYS.
ZZZZ
YES, EVERYTHING WILL BE BEAUTIFUL, BUT WE'LL STILL HAVE TO REGULATE POPULATION GROWTH. SO WHEN YOU'RE 65 THEY'LL COME LOOKING FOR YOU WITH A PIE...NOT JUST AN ORDINARY PIE!!
A CYANIDE PIE!! WHAT A WAY TO GO!!
HONK HONK
THAT'S ALL BROTHER

Follow-up on Palindromes

Remember in the Jan.-Feb. issue on page 12 we asked if readers could improve on Tom Karzes' programs to turn any number into a palindrome by successive reversals and adding. Here's an example:

```
   76
   67
  ---
  143
  341
  ---
  484 a palindrome
```

In the Jan.-Feb. issue we said Tom Karzes' palindrome program "fails with greater than a 7-digit number." What we really should have said is that it will not accept input numbers of over seven digits. In retrospect, this is not much of a limitation at all since palindromic seed numbers are generally much less than 7 digits. His program, in fact, spews out a number as long as a Teletype line. Sorry, Tom!

In any event, Gregory Yob of Menlo Park, CA put some additional sophistication into a palindromic reversal program. Here's the program, a couple of sample runs and then a portion of the run using 196 as a starter (remember — that's the one that doesn't seem to ever become palindromic). Gregory interrupted that run after 101 additions.

He then wrote a souped up version that doesn't print out the calculations but rather just the final palindrome and number of steps to reach it. If a palindrome isn't formed by 254 digits the program quits and prints the last number. Here's a listing of the second program and a couple of sample runs including the run of 196 as a starter.

Want to carry on? Why not modify the program to try all the numbers between 100 and 200 in sequence? Or extend it, if your BASIC compiler permits, to handle a longer number?

```
0070  DIM A$[254],B$[254],C$[254]
0080  PRINT "STARTING #";
0090  INPUT A$
0100  S=0
0110  REM- PRINT LAST SUM
0120  PRINT TAB(10);A$
0130  REM: MAKE REVERSED #
0140   FOR J=LEN(A$) TO 1 STEP -1
0150   K=LEN(A$)-J+1
0160   B$[K:K]=A$[J:J]
0170   NEXT J
0180  REM: PRINT REVERSED # & STEP COUNT
0190  S=S+1
0200  PRINT S;TAB(11);B$
0210  REM- DO THE ADDITION
0220  C$=A$+B$
0230  REM: RESET AND DO AGAIIN
0240  A$=C$
0250  B$=" "
0260  PRINT
0270  GOTO 120
0280  END
```

LISTING

SAMPLE RUNS

```
STARTING #?176
             176
 1           671

             847
 2           748

             1595
 3           5951

             7546
 4           6457

             14003
 5           30041

             44044
 6           44044
```

```
STARTING #?2344
           2344
 1          4432

            6776
 2          6776
```

```
STARTING #?776
           776
 1         677

           1453
 2         3541

           4994
```

AND HERE'S THE BIG ONE (196)

```
STARTING #?196
          196
 1        691

          887
 2        788

          1675
 3        5761

          7436
 4        6347

          7074449215608204801780891870975118165118444806
99        6084448115618115790781980871084028065129444707

          13158897331226320592562872742059146230247889513
100       31598874203264195024727826529502362213379885131

          44757771534490515617290699271561508443627774644
101       44647772634480516517299609271651509443517775744
```

WE DIDN'T HAVE SPACE FOR ALL THE INTERMEDIATE PRINT-OUT (YOU RUN IT!). GREGORY ABORTED THE RUN AFTER 101 ADDITIONS.

1675-Ever a Palindrome?

Two students at Highland Park High School, NJ, Tony Skaltsiotis and Andrew Glassner, decided to test the hypothesis that 1675 does not ever become palindromic, even after infinite reversals. They wrote a program which could reverse and add the number tens of thousands of times, and store up to 7167 digits.

The PDP8/e at the high school could only reverse the number 60 times, so they went to the IBM 360 at Rutgers University.

They write us, "We tested the number for palindromacy while reversing it *ten-thousand times*. We achieved no palindrome, but we did get a 3798-digit number! Since we ran the program using a CRT terminal, we had no hard-copy printout, but we will send you the results after we attempt even more reversals."

[Ed note: I have discussed this problem with Walt Koetke, our Problems Editor, and Greame Levin, Publisher of *Games & Puzzles*. I would speculate that the numbers being formed, probably after something like 100 digits, are essentially *random* and, by definition, sooner or later the reversal will become a palindrome *solely because it is random*. Do you agree? Let's hear more from readers on this. — DHA]

```
0010  REM: FIND A PALINDROMIC NUMBER BY REVERSALS AND ADDITIONS
0020  REM: BY GREGORY YOB  (415) 326-4039
0030  REM: PO BOX 310, MENLO PARK, CALIFORNIA 94025
0040  REM: --- INSPIRED BY ARTICLE IN CREATIVE COMPUTING
0050  REM: --- RUN ON BASIC TIMESHARING SYSTEM 3000
0060  REM: --- WHICH HAS LIMITED STRING ARITHMETIC
0070  DIM A$(254),B$(254),C$(254)
0080  PRINT "STARTING #";
0090  INPUT A$
0100  S=0
0110  REM- PRINT LAST SUM
0120  REM: REMOVED TO ELIMINATE PRINTOUT
0130  REM: MAKE REVERSED #
0140  B$="  "
0150   FOR J=LEN(A$) TO 1 STEP -1
0160   REM: RESULT STRINT BEGINS WITH A BLANK, SO:
0170   K=LEN(A$)-J+2
0180   IF A$(J:J)=" " THEN 200
0190   B$(K:K)=A$(J:J)
0200   NEXT J
0210  REM: PRINT REVERSED # & STEP COUNT
0220  S=S+1
0230  REM: REMOVED & REPLACED BY A TEST
0240  IF A$=B$ THEN 340
0250  IF LEN(A$)=254 THEN 320
0260  REM- DO THE ADDITION
0270  C$=A$+B$
0280  REM: RESET AND DO AGAIIN
0290  A$=C$
0300  B$=" "
0310  GOTO 140
0320  PRINT "IS NOT ";
0330  GOTO 350
0340  PRINT
0350  PRINT "PALINDROMIC AT STEP "S-1
0360  PRINT " THE NUMBER IS:"
0370  PRINT A$(1:70)
0380  IF LEN(A$)<71 THEN 450
0390  PRINT A$(71:140)
0400  IF LEN(A$)<141 THEN 450
0410  PRINT A$(141:210)
0420  IF LEN(A$)<211 THEN 450
0430  PRINT A$(211:254)
0440  GOTO 450
0450  END
```

HERE'S THE SOUPED UP VERSION OF THE PROGRAM

AND HERE'S THE RUN USING 196 AS A STARTER.

```
STARTING #?196

STOPPED AT 200
XI?PRINT S;LEN(A$)
 425   188
```

USE INTERRUPT TO FIND OUT WHERE WE ARE

```
XI?PRINT A$
 60239554322389886717750708169963186816112861056083174321936723132176220
622203953811987008220121132800680208448313126923572231218729124371389660
157321618670468962907057618589874322246582305

XI?GOTO 200

IS NOT PALINDROMIC AT STEP  584
 THE NUMBER IS:
 179783220484416006808962247253580610457420253934734211705030836407528
4812966783124693965464737561164092999951577064625923388740047774319526
4717751599982013522557373744703865302976682185924704639030507112338426
46201385302697635373226089760062438403328896
```

PROGRAM HALTED AT STEP #584 WHEN NUMBER FORMED 254 DIGITS, THE MAXIMUM STRING SIZE OF SYSTEM.

With trembling pseudopods, Rork Glanf tore away the Earth-Girl's space-suit.

SCIENCE FICTION BY DOODLES WEAVER

Time and again, *and overtime*

SEATED IN THE TIME-BINDING HARNESS, and with a smile that could only mean sinister self-assurance, Rork Glanf, inter-stellar spy from Ganymede II posing as Professor Aych Gentry of the Cybernetics-Semantics Laboratory, Earthian Division, pressed the nuclear stub which would release the tensor force fields surrounding him and propel him into another space-time phase. Even as his ducleum-covered waldo made contact with the magnetic knob, Glanf-Gentry reviewed quickly what he must do in the next few moments.

Immediately on arriving in the year he was born, he would enter the place of his birth, disguise himself as an intern, steal into the maternity ward, and surreptitiously exchange two infants in their cribs—himself and his twin brother—thus causing a rupture in the past that would enable him to return to the present as King of the Galaxy,

continued

Time and again, *and overtime* continued

which position his brother now held. Simple, fast, effective, and foolproof!

Gentry-Glanf's pseudopod, disguised as a human finger, released the activating distorter! In the laboratory rose a loud whine and a light flashed reminiscent of a super-nova as Glanf and the time-binding machine disappeared.

° ° °

Shading his eyes with one hand, Professor Karloff looked up from his vivisection of a Syrian aqua-aardvark and said to Adam Rink, the Android: "Bless my garters, what won't that young fool think of next?"

° ° °

Transported instantaneously to the day of his birth, Glanf materialized on the front steps of the hospital, slipped inside, overpowered a lone intern with his portable thalamic-paralyzer, donned the white uniform, walked boldly into the baby ward, went directly over to himself-the infant, removed himself from one crib and substituted his twin brother for himself, putting himself in his brother's bed. Then, cackling mirthfully to himself-as-adult in Saturnian pidgeon-Martian, he reactivated the nuclear stud and before you could say "Wow" he reappeared in the laboratory just as Professor Karloff was ending the speech: "that young fool think of next?"

At that instant his smile of confidence froze. Why had he returned to the laboratory, if he were the King of the Galaxy?

How come he was still an insignificant interstellar spy?
Why was he not in the Uranium Chair of the King?
What had gone wrong?
Where was the mixup? Why . . .

"I'll tell you why!" shouted Adam the Android (who was also a telepath), tearing off his human face and revealing the lizard-like features of the terrible man-eating Plutonian Quaggle-beast: "Because I am really your twin brother and I perfected the time-binding machine just twenty minutes before you did, and I went to our birthplace and switched the babies first, so you actually put yourself back to where you were in the first place!"

Then laughing like a moon-mad space pirate, the android-human-Quaggle-beast slid across the floor on his nineteen appendages into a teleportation booth, appearing immediately in his Uranium Chair at the meeting of the Galactic Council, where he quickly signed a document recommending death for his twin brother, Professor Aych Gentry, who was really Rork Glanf.

But Glanf the Ganymedian was not so easily defeated. Instantly he pressed the stub of the tensor force-field machine and returned to the past two hours before the preceeding conversation occurred. There he perfected the Time Machine, and returned to the present one-half hour before his twin brother had perfected it, then rapidly returned to the day of his birth, overpowered the intern, switched the babies, and reappeared in the laboratory just as Professor Karloff was ending the speech: "that young fool think of next?"

° ° °

At that instant his smile of confidence froze. Why had he returned to the laboratory, if he were King of the Galaxy?.

How come he was still an insignificant inter-stellar spy?
Why was he not in the Uranium Chair of the King?
What had gone wrong?
Where was the mixup? Why . . .

"I'll tell you why!" howled Adam the Android (also a telepath), tearing off his human face and revealing the ugly features of the truculent flesh-eating Plutonian Quaggle-beast.

"Because I, Adam, I am really your twin brother and I knew you were going to go back in time two hours before I perfected the Time Machine so I went back **four** hours before you went back the second time and then I returned to our birthplace and switched the babies before you switched the babies the second time after you had switched the babies the first time, so you actually put yourself back to where you were the time before you switched the babies the second time!"

Then shrieking like an insane spider the human-android-Quaggle-beast slushed across the floor on his nineteen tentacles into a teleportation booth, appearing immediately in the Uranium Chair at the meeting of the Galactic Council, where he signed a death warrant for his twin brother, Professor Gentry, who was really Rork Glanf.

But the spy-Professor-Ganymedian was not so easily defeated. Instantly he pressed the stub of the time binding machine and returned to the past, six hours before the two preceding occurrences, perfected the Time Machine, returned to the present one-half hour before his twin brother had perfected it the second time, went back to the day of his birth, overpowered the intern, switched the babies, and reappeared in the laboratory just as Professor Karloff was ending his speech: "that young fool think of next?"

° ° °

But his smile of confidence froze. Why had he returned to the laboratory, if he were King of the Galaxy? What had gone wrong? Where was the mixup? Why . . .

"I'll tell you why!" screamed Adam the Android (who could also read minds), tearing off his face and so on: Because I am really your twin brother and I knew you were going to go and so on and on . . ."

° ° °

Those two little babies really got around, hey?

° ° °

P. S. You rascals looking for the part about Earth-Girl shown in opening illustration—never mind! It was all a hoax by us foxy editors to make you read story.

Problems for Creative Computing . . . .

AEDI, MUTAB, NEDA and SOGAL

by Walter Koetke
Lexington High School

Your non-terrestrial thoughts should not remain free of problems that require creative solutions. Toward that end, here are two situations that you might find interesting. After solving either one or both of these problems, please send your solution to Walter Koetke at the *Creative Computing* address. The best solutions received will be acknowledged in a future column.

If you think you've seen the first problem before, you may be correct. It's really an old problem in a new disguise.

The civilizations of the three planets Neda, Mutab and Sogal have agreed to begin a war in the year 2431. Although these societies have not eliminated such irrational actions as war, they have at least formalized the process. There are, for instance, no guerilla activities and wars are usually very brief and always decisive. Wars are fought with inter-planetary rockets each of which is powerful enough to completely destroy an entire planet. With such powerful weapons at their disposal, Neda, Mutab and Sogal have agreed to the following set of rules, for only in this way can they be assured of a single victor.

Rule 1: The fight will continue until only one civilization remains.

Rule 2: The rather primitive technique of drawing lots will be used to determine which planet may launch the first rocket, which the second and which the third.

Rule 3: After the launching rotation is established, rocket launching begins and continues in order until only one planet remains.

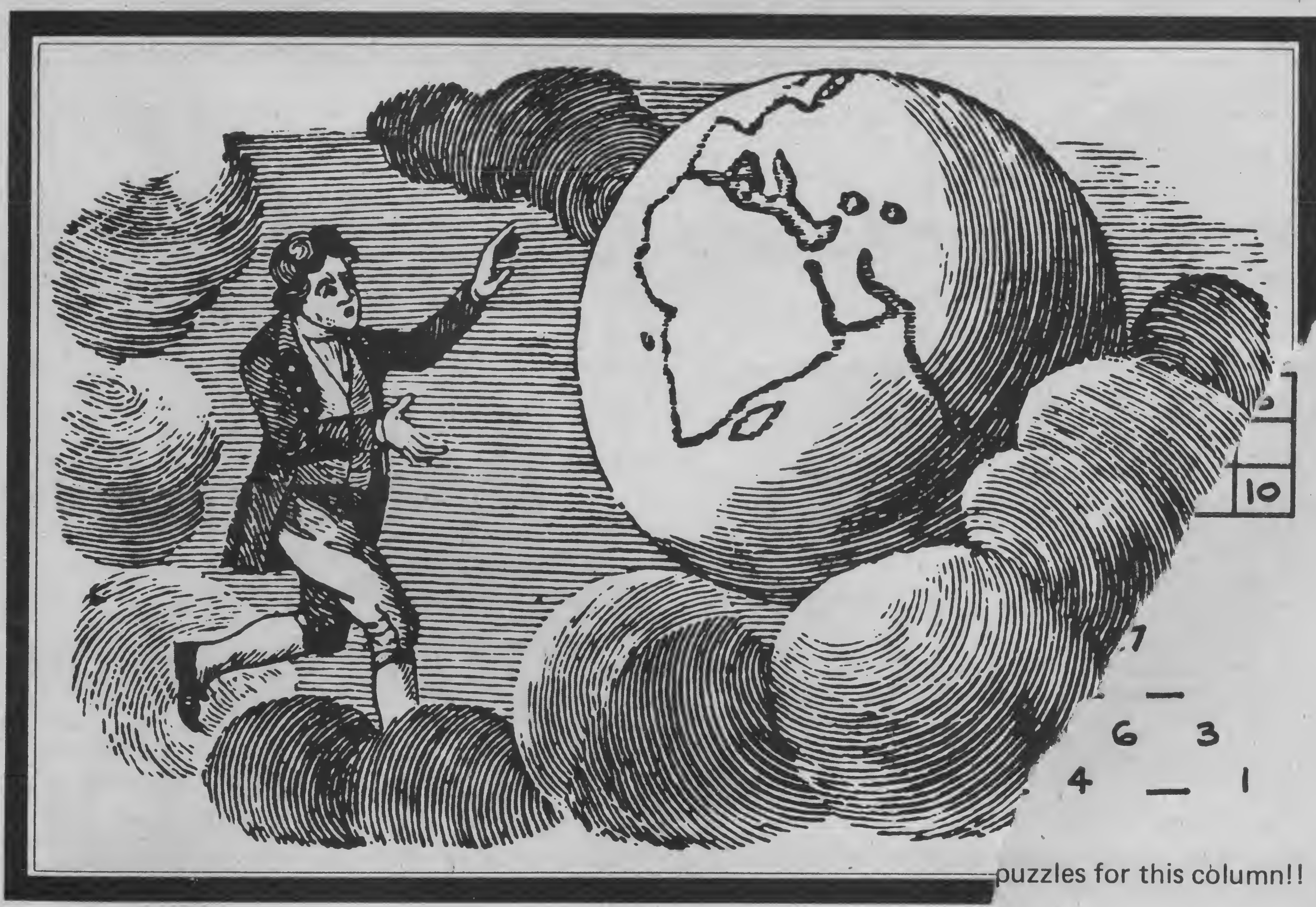
When contemplating the outcome of this war, the three civilizations have full knowledge of the background of their adversaries.

Mutab is clearly the technologically superior civilization. Once launched, their rockets always strike with perfect accuracy — thus disproving a modern theory that nothing is perfect. Before this war begins, both of the other civilizations are aware of the terrifying fact that if a Mutab rocket is fired at them, the probability of their being completely destroyed is 1.

Neda is the oldest civilization and long ago had the superior technology. However, the complacency of a self-centered, unchallenged mind has been eroding this superiority for many years. As a result, the technology of Neda has not advanced in over 40 years. If a Nedian rocket is fired at another planet, the probability of hitting that planet is 0.8, just as it was 40 years ago.

Sogal is by far the newest of the three civilizations. Being dedicated to producing its own technology on its own terms has resulted in a proud and purposeful civilization, but one that is technologically four to five hundred years behind its present adversaries. A missile launched by Sogal has only a 50-50 chance of reaching its intended target.

Your role in this future war is to determine each civilization's probability of winning.



puzzles for this column!!

AEDI continued . .

The second problem is based upon an idea presented by C. Stanley Ogilvy in the text *Tomorrow's Math*, Oxford University Press, 1972. However, you should attempt your own solution before seeking Ogilvy's support.

Although the civilization on the planet of Aedi is generally considered rather advanced, its political system no longer attracts the imagination and support of the majority of citizens. In an effort to attract more capable leaders at the highest level, a new plan was formulated for selecting the president. The originators of the plan also hoped that their new idea would result in a younger president and a change of presidents at least every 10 years.

Essentially, the new plan is as follows. Once a president is selected, he holds office for at least five years. At that time he may or may not be replaced by a newly selected person. The selection process is new and is the key to this new plan. The selection process is a problem — one problem known by all citizens at all times. When a president has served for five years, all citizens of Aedi are invited to submit their solution to the problem. If a solution submitted is better than that previously submitted by the current president, then the submitter becomes the new president. If a solution submitted is equal to that previously submitted by the current president, then the submitter becomes the new president only if his solution is also different from that previously submitted by the current president.

The problem used by the Aedians to select their leader can be attacked on many different levels. The problem involves three sets of three instructions each and a board on which play is recorded. Three blank instruction sets and the playing tablet appear as:

B	
0	
1	

INSTRUCTION SET 1

B	
0	
1	

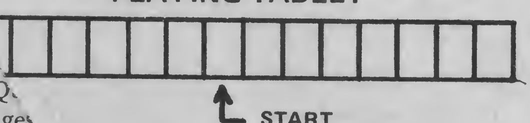
INSTRUCTION SET 2

B	
0	
1	

INSTRUCTION SET 3

t...
pla...
yours...

PLAYING TABLET



Then
human-Q...
appendages
diately in his
Council, where
ing death for hi...
was really Rork

...e of the form STOP (self-explanatory) or ...ements:

...tion of what to record on the playing ... only possibilities are 1, 0 or B (blank).

But Glanf the Ga...
Instantly he pressed t...
chine and returned to ...
ceeding conversation occ...
Time Machine, and returne...
before his twin brother had...
returned to the day of his bir...

... of which direction to move on the ... One square left or right are the only ...h instruction board contains ... be followed.

... record a 1 on the playing ... go to board 3 for the next

President of Aedi

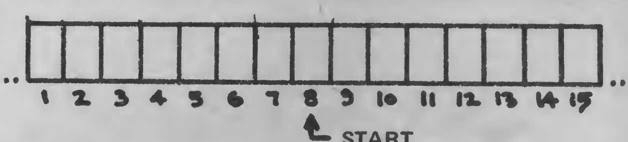
The combination of the contents of your place on the playing tablet and the instruction board you are following dictate your next insturction. The left column of the instruction board indicates B (blank), 0 or 1. If your current place on the playing tablet is blank, you follow instruction B; if it is a 0, you follow instruction 0; and if it is a 1, you follow instruction 1. The play always begins with board 1.

Consider the following complete set of instructions. If you think you understand the rules, try following the instructions before reading further.

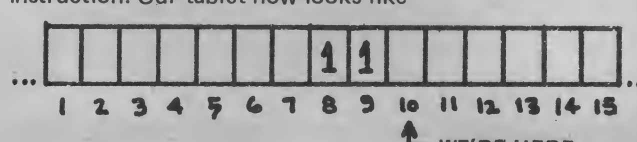
B	1 - R - 2
0	B - R - 3
1	0 - L - 2

B	1 - R - 3
0	1 - L - 1
1	B - L - 3

B	1 - L - 1
0	STOP
1	1 - R - 1



The infinite tablet has been partially numbered for the convenience of this discussion. Play begins on board 1 and, since square 8 is blank, our move is 1-R-2. Thus we write a 1 in square 8, move 1 square to the right (square 9) and go to board 2 for the next instruction. Since square 9 is blank, the second instruction is 1-R-3. Once again we write a 1, move to the right, and this time go to board 3 for the next instruction. Our tablet now looks like



The next instruction is 1-L-1, which records a 1 in square 10, returns us to square 9 and indicates that the next instruction is on board 1. Because square 9 contains a 1, our instruction is 0-L-2 so we replace the 1 with a 0, move to square 8 and proceed to board 2 for the next instruction. The tablet now appears as:

...	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...
...								1	0	1						...

↑ WE'RE HERE (square 8)

And on we go. If you continue following these instructions until you reach STOP, the tablet will finally appear as:

...	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...
...						1	1	1	0	1						...

When STOP is reached, the success of the effort is measured by the longest string of consecutive ones that appear on the tablet. In the example, the longest string contained but three ones.

The Aedians' problem was not to follow a particular instruction set, but to create one. Specifically, their leader would be the person who could write the series of instructions that would produce the longest *finite* sequence of consecutive ones. Since you've just seen the example used to introduce the problem to the young Aedians, you'll have to beat three consecutive ones before you're their new leader. If you generate an impressive series, be sure to send the instructions to *Creative Computing.* All worlds seem desperately in need of leaders and we'll gladly publish your name as a likely candidate.

* * *

Never underestimate the importance of just fooling around.

Kenneth Boulding

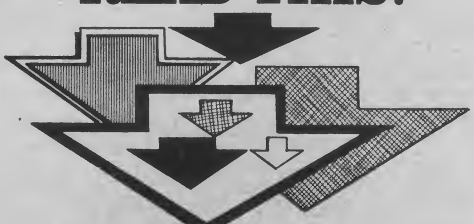
* * *

"The only time my education was interrupted was when I went to school."

George Bernard Shaw

* * *

READ THIS!

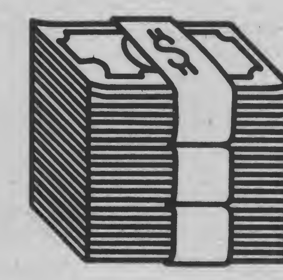


We are skipping the July-August issue and resuming with the Sept.-Oct. issue. All subscriptions will be EXTENDED one issue to compensate. This is not a regular occurence; we are doing it in 1975 only to bring Volume 1 of six issues in line with the calendar year, i.e., No. 6 will now be the Nov.-Dec. '75 issue and Volume 2 will start with Jan.-Feb. '76.

Puzzles and Problems For Fun

▶ The number $153 = 1^3 + 5^3 + 3^3$ Find all other 3-digit numbers that have the same property. How about 4-digit numbers? To the 4^{th}?

Bill Morrison
Sudbury, Mass.



▶ Mr. Karbunkle went to the bank to cash his weekly paycheck. In handing over the money, the cashier, by mistake, gave him dollars for cents and cents for dollars.

He pocketed the money without examining it and spent a nickel on candy for his little boy. He then discovered the error and found he possessed exactly twice the amount of the check.

If he had no money in his pocket before cashing the check, what was the exact amount of the check? One clue: Mr. Karbunkle earns less than $50 a week.

▶ Can you find the missing number for each diagram? You first have to figure the pattern which may be horizontal or vertical with a relationship between every number, every second or third number. You may have to add, subtract, multiply, divide, invert or do a combination of these things. Have fun!

A.

3	5		17
7		25	49
	37	73	

B.

	13	6
7	9	
12		10

C.

4	20	5
	8	8
6	54	9
7	49	7

D.

27

— —

12 6 3

— 4 — 1

Send us your favorite puzzles for this column!!

PILOT 73 Information Exchange

A little while ago, I was wondering how PILOT 73 was getting along. As I asked my friends what's up, I noticed that . . . Lots of folk are using PILOT, but there's nobody who knows much about who's doing what with which machine, et cetera.

A typical situation is: There's an experienced FORTRAN programmer who is eager to develop a PILOT interpreter in ANSII FORTRAN IV. However, he doesn't know of anyone who wants such a system. As I was looking through another friend's correspondence, I discovered several requests for a PILOT written in FORTRAN.

As a dedicated PILOT – person, (having written several versions from time to time) I came up with a neat idea . . .

A FREE OR LOW-COST CO-ORDINATING SERVICE FOR PILOT USERS!!!

Rather than waiting for somebody else to do this, I undertake this task (as fate allows me to have the requisite time). Now to flesh out the idea:

WHAT YOU CAN DO FOR ME

At the moment, there are lists of names with cryptic notes piled upon my desk. These are persons interested in PILOT in some way. However, there's very little knowledge beyond the names. Spend a few minutes, answer the questions below, and send to me. If you do this, you become a member of the Exchange.

WHAT I CAN DO FOR YOU

1) *Summary Sheet* I will assemble a catalog of PILOT people and send it to you. A short summary of each person or group's interests will be included under general areas of interest (ie, all B5500 users). From time to time, updated catalogs will be sent out. This is free until costs become excessive.
2) *Connections* Send me your need or request and I shall pass it on to those with the resource you need. Both you and they will get a card indicating your area of mutual interest.
3) *Resource Center* I shall maintain a library of PILOT resources, including listings of PILOT programs, PILOT interpreters, translators, and other implementations, manuals, technical tricks, and so forth. An index of the library will be attached to the Summary Sheet. Help the library grow by contributing a copy of your aspect of PILOT. If a particular item in the Library is of interest to you, I will make copies for you at cost.

Remember, the more I know, the better the service is for you.

Gregory Yob
PO Box 310
Menlo Park, Calif. 94025

PILOT 73 RESOURCE QUESTIONNAIRE

NAME:
ADDRESS:
PHONE:

1) Do you have a working version of PILOT?
2) If yes, on which machine(s)?
   2.1) Host language?
   2.2) Core Memory required?
   2.3) Configuration of Peripherals?
   2.4) How well is it debugged?
   2.5) Did you write it yourself?
   2.6) Compiler/Interpreter/Translator?
   2.7) Performance/response time?
   2.8) Do you have a user's Manual? (Send a copy)
   2.9) Do you have a listing and paper tape? (Send a copy)
3) If you are writing PILOT programs, are there some available for others? (send copies please)
4) Are you looking for a version of PILOT? (If yes, 2.1 – 2.9 above)
5) Please state your needs and interests:

6) Names and addresses of other persons you feel may be interested in the Exchange:

Nolan Bushnell- Father of PONG

by Trish Todd

You spot a large metal machine in the corner; it has two knobs, a coin slot, and a television screen which shows a dot of light lazily bouncing off the sides of the screen at irregular angles. Immediately, you are curious and begin to read the instructions to Pong, one of the computerized games from Atari, Inc.

The company was started two years ago by Nolan Bushnell, who was managing an amusement park to finance an electrical engineering major. He built the first game prototype, Computer Space, in his garage, and then he met "Moose," who built games from other designs of Bushnell's for a percentage. The first was Pong; it was tested in a bar called Andy Capp's – and in 24 hours you couldn't get near it. The "company" then expanded to include twelve other people who worked together and produced ten Pongs a day. Each Pong brings in about $200 a week.

The "company" has now developed into Atari, Inc., worth over $20,000,000. Located in Los Gatos, California, Atari manufactures Pong, Gotcha, Rebound, Space Race, Super Pong, Pong Doubles, Quadrapong, and Grantrak 10. Bushnell now owns Atari, and Moose is vice-president of research and development; both are confronted with lawyers, patents, security, and labor problems which they never foresaw in Bushnell's garage.

As Atari has grown, Nolan has tried to retain the "Atarian philosophy," which is based on dignity, trust, freedom, and loyalty. This philosophy is intended to produce a comfortable working environment; for example, labor and management share the same medical-insurance program (which also covers unwed pregnancies). The company tries to promote an informal atmosphere, both on the assembly line and between labor and management. Both men and women work on the assembly line, where judging from the hair and attire, sex is hard to distinguish. There are both men and women administrators too. Nolan's management philosophy is occasionally revealed in the form of a surprise party for all the employees – and he buys the liquor!

However, the "Atarian philosophy" has also had to cope with several problems. Workers have been uneasy about the absence of a union; the assembly line is a hazardous place, and an exploding television screen can permanently ruin a limb. Workers have complained of low wages. Atari has also had to hire security guards to protect itself from theft by employees who have used the philosophy to their advantage.

Atari's success is a result of its product's popularity. Their "computer" games are found in bars, lounges, hotel lobbies, banks, and country clubs; in Hawaii, Pongs may be found on the sidewalks chained to parking meters. Its popularity lies in its sophistication; like tennis, it involves coordination and brain power, and the more one practices, the less is left to chance. These games addict their players because the final result is either frustration or reward. The games also easily lend themselves to socialization through light-hearted competition or, as in Pong Doubles, through teamwork. Atari has gone one step beyond novelty and developed a true participatory sport.

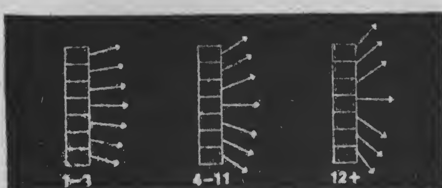
As the blip lazily glides into your goal and your opponent scores again, it is hard to realize that the game is based on the algorithms that are built into a computer's circuitry system. So quarter after quarter is deposited in the slot, Atari's profits zoom upward, and the computer becomes increasingly important in America's leisure time.

Playing PONG to *Win*

by David Ahl

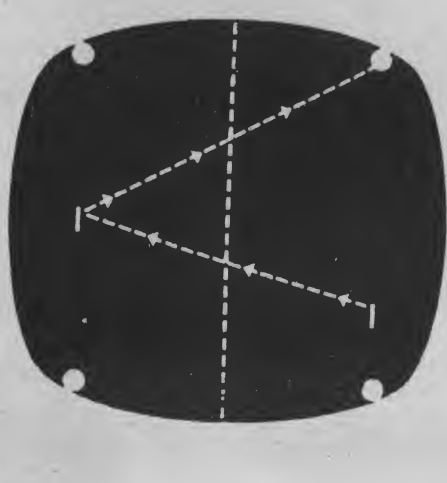
First you should understand that Pong is merely a miniature "computer" attached to a TV screen. The behavior of the ball and paddles is permanently programmed into the "computer" or PC board. If X happens, Y will result. Simple. No luck involved. No body English. No spins on the ball. Understand the algorithms and you can win the game.

The Paddle
The Pong paddle seems to be a single unbroken surface. Many players believe it can impart a spin to the ball as in actual ping-pong or tennis. Wrong. The paddle actually consists of seven sections. Each section returns the ball at a predetermined angle, *no matter what the angle of incidence.* The middle section returns it horizontally, the end sections at the greatest angles. The others are in between. Try to set your paddle for a return as soon as possible and fine tune it on the final approach of the ball.

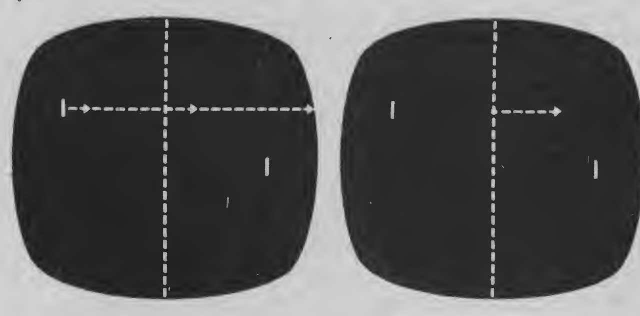


The Volley
The ball may seem to speed up with every volley. It doesn't. But it does speed up on the fourth and twelfth volleys. And, devilishly, the return angles of the paddles increase on these volleys too.

Gotcha Zones
A shot hit to the four corners of the screen cannot be returned no matter what the position of the paddle. You can most easily hit the ball to your opponent's corners from the center of the court; it's more difficult as you get to either of your corners. Hence, try to aim the ball at your opponent's corners to either score on him or at least prevent him getting it to your corners.



The Serve
The person who misses a point always receives the next serve. (On some earlier Pongs this algorithm was reversed, i.e., person who scores a point receives the next serve). You can predict where the serve will come from by simply imagining the screen wrapped around a cylinder with a second screen in back that you can't see. If the ball went off the screen fairly straight, it will appear from a continuation of the same path it was tracing. If it went off at an angle, it will bounce against the edge of the invisible screen on the other side of the cylinder and reappear at the new (opposite) angle. If you miss an angular serve, the next one will approach slightly higher or lower (unless the angle was exactly 45° in which case it will come from the same point).



Now go hustle your friends. And if they want to know how you got so good all of a sudden, tell them to subscribe to *Creative Computing*.

Supplement to the Whole Earth Catalog
The CoEVOLUTION Quarterly

CoEvolution Quarterly is the latest in an extraordinary series of publications from Stewart Brand, leader and spokesman for the Alternative Press movement.

I visited briefly with Stewart last November in his warehouse-on-the-water in Sausalito, windowless unfortunately. The place is piled floor-to-ceiling with magazines, newsletters, and information about virtually every project, organization, store or source in the U. S. It brings to mind NBC News Central although here the information is the written word rather than tapes and TTY prinout. Here in this dilapidated building is both the nerve center and central clearinghouse for the alternative press of the nation.

You have all seen *Whole Earth Catalog, Last Updated Whole Earth Catalog,* and *Whole Earth Epilog* full of those perceptive, mind-expanding notations followed by the now famous "--SB" (Stewart Brand). *CoEvolution Quarterly* brings you updates of material from the catalogs, new listings, as well as some longer articles and stories. Packed with information and sources about land use, shelter, food, community, learning, communications, and the alternative life. Handled by many local bookstores or $6.00/year from 558 Santa Cruz, Menlo Park, CA 94025.

History shows that new ideas in science most often come from brash youngsters, mavericks, or rank outsiders.

Fortune, May 1974

Another new game from Creative Computing . . .

LUNAR

by David Ahl

LUNAR, also known as ROCKET, APOLLO, LEM, etc. is, next to STAR TREK and SPACE WAR, the most popular computer game. It is certainly the most popular on smaller machines. (I remember a milestone of sorts when I managed to compress LUNAR to run on 4K PDP-8 BASIC while retaining full instructions and landing messages. I used every single character available.)

The version of LUNAR presented here was originally written in FOCAL by Jim Storer, a student at Lexington (Mass.) High School in the mid 60's. While everyone claims to be the original program author of LUNAR, I'm reasonably sure that Jim predates the others and therefore qualifies as the original, original author. I converted the program to BASIC in early 1970. It's a straight-forward version without side stabilization rockets or other goodies but, nevertheless, is quite a challenge to land successfully.

PLAYING THE GAME

Your mission is to achieve a soft landing of your LEM on the moon. You separate from the command ship 200 miles above the surface of the moon and, every 10 seconds, set the burn rate of your retro rockets to slow your craft. You may free fall (0 lbs./sec.) or burn at any rate between 8 lbs./sec. and 200 lbs./sec. Since ignition occurs at 8 lbs./sec., burn rates between 1 and 7 lbs./sec. may not be used. A negative burn rate automatically aborts your mission.

There are three popular ways to land:

1. Constant burn rate all the way down.
2. Free fall for a while, then maximum burn rate tapering off as you get close.
3. Gradually increase burn rate to a maximum, then taper off as you get close.

Recall from physics that Newton found the force of attraction (gravity) between two bodies varies directly with the mass of the bodies and inversely with the square of the distance between their centers. This may help you land successfully. Then again, it may not.

COMPUTER NOTES

Convert the program to your version of BASIC. Multiple statements on one line are separated by a colon (:). Everything else is standard.

Some computers produce an error calculating the expansions (Statements 910 and 920) when you get close to the moon and the numbers get very small. If yours does, substitute the expanded form. Here it is for Statement 910:

```
-Q*(1+Q*(1/2+Q*(1/3+Q*(1/4+Q/5))))
```

You should be able to figure out the other one yourself.

Would you like us to print the other versions of LUNAR in *Creative Computing*? If so, write and let me know—DHA.

MAPPING THE MOON

This photograph pictures Mare Crisium, the large "flat" area near the eastern edge of the moon as seen from Earth. In the foreground is the mountainous terrain that forms the southern rim of Mare Crisium. Visible near the horizon, 285 miles across the mare, is its northern rim. Prominent at above right in the mare is the 24-mile-wide crater Picard. Photo was made from Apollo 10, the last flight before the lunar landing. (Photo Kodak)

SAMPLE RUN

```
RUNNH
LUNAR LANDING SIMULATION


CONTROL CALLING LUNAR MODULE..

YOU MAY SET THE FUEL RATE (K) TO ZERO OR ANY VALUE
BETWEEN 8 AND 200 LBS PER SECOND.  A NEGATIVE FUEL
RATE WILL ABORT THE MISSION.

YOU HAVE 16000 LBS OF FUEL.
ESTIMATED FREE FALL IMPACT TIME IS 120 SECONDS.
CAPSULE WEIGHT IS 32,500 LBS.

FIRST RADAR CHECK COMING UP...
BEGIN LANDING PROCEDURE


TIME(SECS)    HEIGHT(MI)     VELOCITY(MPH) FUEL(LBS)      FUEL RATE
 0             120            3600          16000          K? 0
 10            109.95         3636          16000          K? 0
 20            99.8           3672          16000          K? 0
 30            89.55          3708          16000          K? 0
 40            79.2           3744          16000          K? 0
 50            68.75          3780          16000          K? 0
 60            58.2           3816          16000          K? 0
 70            47.55          3852          16000          K? 200
 80            37.3656        3476.43       14000          K? 200
 90            28.2623        3072.94       12000          K? 200
 100           20.3232        2637.46       10000          K? 200
 110           13.644         2164.97       8000           K? 200
 120           8.33572        1649.14       6000           K? 200
 130           4.52958        1081.92       4000           K? 180
 140           2.2887         522.398       2200           K? 90
 150           1.22786        238.772       1300           K? 40
 160           .718389        127.493       900            K? 20
 170           .418089        88.5788       700            K? 22
 180           .237646        41.1605       480            K? 14
 190           .147722        23.5114       340            K? 10
 200           .859622E-1     20.9169       240            K? 10
 210           .317304E-1     18.0912       140            K? 12
ON THE MOON AT  218.451 SECS.
IMPACT VELOCITY OF  8.90172 M.P.H.
FUEL LEFT  38.5879 LBS.
VERY GOOD LANDING, NOT PERFECT YET.


TRY AGAIN (1 FOR YES, 0 FOR NO)? 0

CONTROL OUT

READY
```

PROGRAM LISTING

```
LISTNH
1 REM *** WRITTEN BY JIM STORER, LEXINGTON HS
2 REM *** CONVERTED FROM FOCAL TO BASIC BY  DAVID AHL, DIGITAL
10 PRINT "LUNAR LANDING SIMULATION":PRINT
20 PRINT:PRINT:PRINT "CONTROL CALLING LUNAR MODULE.  ":PRINT
35PRINT"YOU MAY SET THE FUEL RATE (K) TO ZERO OR ANY VALUE"
40PRINT"BETWEEN 8 AND 200 LBS PER SECOND.   A NEGATIVE FUEL"
50 PRINT "RATE WILL ABORT THE MISSION.":PRINT
60PRINT"YOU HAVE 16000 LBS OF FUEL."
70PRINT"ESTIMATED FREE FALL IMPACT TIME IS 120 SECONDS."
80PRINT"CAPSULE WEIGHT IS 32,500 LBS."
90 PRINT:PRINT "FIRST RADAR CHECK COMING UP..."
100 PRINT "BEGIN LANDING PROCEDURE":PRINT:PRINT
110PRINT"TIME(SECS)","HEIGHT(MI)","VELOCITY(MPH)","FUEL(LBS)","FUEL RATE"
120 LET L=0:LET A=120:LET V=1:LET M=32500:LET N=16500
170 LET G=.001:LET Z=1.8
210 PRINT INT(L+.5),A,V*3600,M-N,"K";
220 INPUT K
225 LET T=10
230 IF K<0 GO TO 590
235 IF K=0 GOTO 310
240 IF K<8 THEN 260
250 IF K<=200 GO TO 310
260 PRINT "NOT POSSIBLE",,,,"K";
270 INPUT K : GOTO 230
310 IF M-N-.001 <=0 GOTO410
320 IF T<.001 GOTO 210
330 LET S=T:IF N+S*K<=M GO TO 350
340 LET S=(M-N)/K
350 LET I0=1 : GOTO 900
360 IF I <= 0 GOT O 710
370 IF V<=0 GO TO 380
375 IF J<0 GOTO 810
380 LET I0=1:GOTO600
410 PRINT"FUEL OUT AT ";L;"SECS."
420 LET S=(-V+SQR(V*V+2*A*G))/G
430 LET V = V+G*S
440 LET L=L+S
510 PRINT"ON THE MOON AT ";L;"SECS."
511 LET W = 3600*V
514 PRINT "IMPACT VELOCITY OF ";W;"M.P.H."
520 PRINT "FUEL LEFT ";M-N;"LBS."
530 IF W>=1 GOTO 550
540 PRINT "PERFECT LANDING!  CONGRATULATIONS!!": GOTO 590
550 IF W >=10 THEN 560
552 PRINT "VERY GOOD LANDING, NOT PERFECT YET.":GOTO590
560 IF W >= 25 THEN 570
562 PRINT "A FAIR LANDING, NO CRAFT DAMAGE.":GOTO590
570 IF W >= 60 THEN580
572 PRINT "CRAFT DAMAGE.  HOPE YOUR OXYGEN HOLDS OUT UNTIL A"
574 PRINT "RESCUE MISSION ARRIVES!":GOTO 590
580 PRINT "SORRY, BUT THERE WERE NO SURVIVORS."
585 PRINT "IN FACT YOU BLASTED A NEW LUNAR CRATER ";W*.277777;" FEET DEEP.
590 PRINT:PRINT:PRINT "TRY AGAIN (1 FOR YES, 0 FOR NO)";
592 INPUT R:IF R=1 THEN 90
595 PRINT: PRINT "CONTROL OUT" : GOTO 1800
600 LET L=L+S
610 LET T = T-S
620 LET M=M-S*K
630 LET A=I
640 LET V=J
650 IF I0=1 GO TO 310
660 IF I0=3 GO TO 850
710 IF S< .005 GO TO 510
720 LET S= 2*A/(V+SQR(V*V+2*A*(G-Z*K/M)))
730 LET I0=2 :GOTO 900
810 LET W=(1-M*G/(Z*K))/2
820 LET S=M*V/(Z*K*(W+SQR(W*W+V/Z)))+.05
825 LET I0=3 :GOTO 900
830 IF I<=0 THEN 710
840 GOTO 600
850 IF J>=0 THEN 310
860 IF V<=0 GO TO 310
870 GOTO 810
900 LET Q=S*K/M
905 IF Q<=0 THEN 1000
910 LET J=V+G*S+Z*(-Q*(1+Q*(1/2+Q*(1/3+Q*(1/4+Q*(1/5))))))
920 LET I=A-G*S*S/2-V*S+Z*S*(Q*(1/2+Q*(1/6+Q*(1/12+Q*(1/20+Q*(1/30))))))
930 IF I0=1 GOTO 360
940 IF I0=2 GOTO 600
950 IF I0=3 GOTO 830
1000 LET J=V+G*S
1010 LET I=A-G*S*S/2-V*S
1020 GOTO930
1800 END

READY
```

Appollo 14 Launch Control Center, Cape Kennedy, Florida

WOULD YOU LIKE TO EXPRESS YOURSELF IN A PERSONAL VISIT TO AN EDITOR?

All the *Creative Computing* editors are randomly available. However, most of us have other jobs which puts bread on the table so it isn't the greatest idea to phone us during the day. Probably the best time to see one of us is face-to-face at a conference. Here is where we'll be (or will have been) over the next few months.

Mar.	20-24	NSTA, Convention Center, Los Angeles
Apr.	23-26	NCTM, Curригan Hall, Denver, Co.
Apr.	28-May	AEDS, Cavalier Hotel, Virginia Beach
May	19-23	NCC, Convention Center, Anaheim, Ca.
May	26-28	NY Book Fair, (watch local newspaper for location), New York, N. Y.
June	16-18	CCUC/6, Texas Christian Univ., Ft. Worth, Tx.
Sept.	1-5	IFIP, Marseilles, France
Oct.	20-22	ACM, Radisson Hotel, Minneapolis, Mn.

We are skipping the July-August issue and resuming with the Sept.-Oct. issue. All subscriptions will be EXTENDED one issue to compensate. This is not a regular occurence; we are doing it in 1975 only to bring Volume 1 of six issues in line with the calendar year, i.e., No. 6 will now be the Nov.-Dec. '75 issue and Volume 2 will start with Jan.-Feb. '76.

Nicholas Copernicus

Copernicus was born on February 19, 1473, in Torun, Polish Prussia, the son of a merchant. Though today he is remembered as the father of modern astronomy, in his own time he made his livelihood not as an astronomer, but as an ecclesiastic. His position as canon of the cathedral of Frombork gave him both financial security and freedom to pursue his own astronomical studies. It was at Frombork that Copernicus worked out his great book, the De Revolutionibus.

The accepted model of the structure of the universe in Copernicus' time was earth-centered. The sun, the moon, the five known planets, and the stars were thought to revolve about the earth in endless, perfect circles.

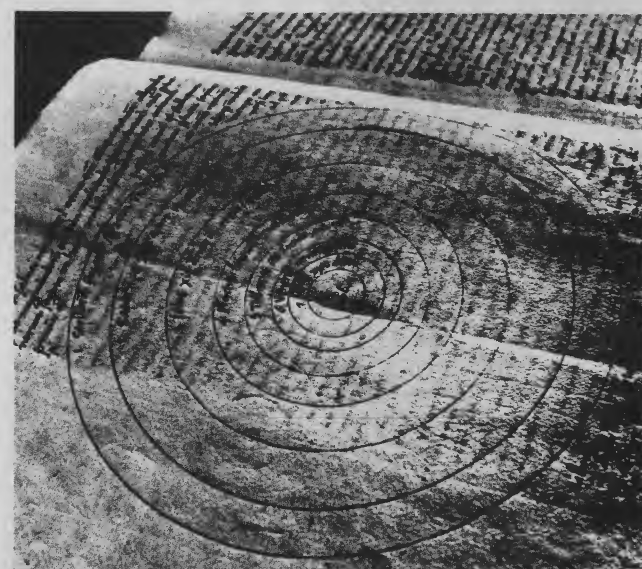
The model was developed by Aristotle around 350 B.C., and elaborated by Claudius Ptolemy of Alexandria around 150 A.D. Ptolemy outlined his system in a treatise which has come to be known as the Almagest, *meaning "the greatest." What Ptolemy established for the first time was a working mathematical model by which the positions of the planets could be predicted accurately.*

In the Ptolemaic system, each planet moved in a small circle (or epicycle) whose center was carried round the earth in a larger orbit (or deferent). For fourteen centuries astronomers computed planetary positions from tables based on this analysis.

Copernicus described an unfamiliar universe, with the sun, not the earth, at its center; he treated the earth as a planet among the other planets, with a yearly orbit around the sun, a daily rotation on its axis, and a conical precession.

His great breakthrough is the recognition that the complex paths which we see traced by the planets could be explained by a combination of their own motion and that of the earth from which we observe them.

His new model gave astronomical inquiry the direction it still follows today. But he arrived at his innovations using the traditional assumption, shared by Aristotle and Ptolemy, that the motion of heavenly bodies must be a compound of circles—an idea which was soon to be overthrown by Johannes Kepler, as he worked to build a foundation for the Copernican hypothesis.



Copernicus' cosmology drawing, from his manuscript of the De Revolutionibus.

Copernicus' book, the De Revolutionibus, *had an immediate impact on astronomical theory. "Astronomy is written for astronomers," he wrote in his preface. But in the coming century his re-examination of the structure of the universe would permanently alter the way people thought of themselves and their world.*

The De Revolutionibus—*published only in the year of its author's death—might never have appeared in print if Georg Joachim Rheticus, a young professor of mathematics, had not persuaded Copernicus to entrust him with the manuscript for publication.*

A new simulation . . .

Escape

by Dr. J. Harris
Chelsea Centre for Science Education
University of London, England

INTRODUCTION

In this module you are going to investigate how an object travels if it is launched vertically upwards from the earth's surface. You will be able to find out how far away it can travel, what initial velocity it must have to reach a certain height, and so on.

Of course the computer doesn't actually do the experiments. It might — but in this case doesn't — simply tell you the results of trials which have been made. What it does do is to work out the results, based on particular physical laws, which we have good reason to have faith in. The calculations are also based on some simplifying assumptions. All this is explained in the body of the text.

MOTION IN A GRAVITATIONAL FIELD

Imagine that you are trying to throw a ball straight upwards, as high as you can. Obviously the harder you throw it the higher it will get before reaching its highest point and starting to fall back to earth.

Q1 Suppose that for your strongest throw the stone gets to a height of 5m. How high would the stone get if you could give it twice the initial speed?

	A	*7.1m*
	B	*10m*
	C	*14m*
	D	*20m*
or	*E*	*25m*

If you could answer that correctly you should know the height which the stone reaches depends on how much kinetic energy it has to start with, and that kinetic energy depends on $(\text{speed})^2$. So twice the velocity means four times the kinetic energy. As the stone goes up its kinetic energy is transformed to potential energy. The potential energy depends on how high above the ground the stone is, and when the stone is 20m above ground, the potential energy is four times that for 5m above ground.

So we can calculate how far the stone will go for a particular initial speed, however great. We can also calculate what initial speed it must be given to reach a particular distance.

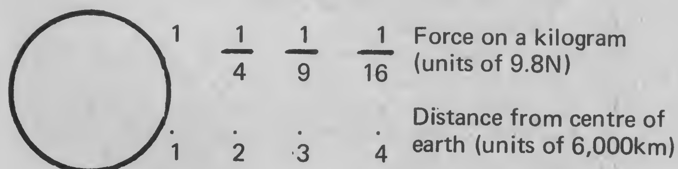
The Chelsea Science Simulation Project is the British counterpart to the Huntington Computer Project in the United States. In other words, it is a project to produce high-quality computer simulation modules consisting of a program, student's workbook, and teacher's guide. Most of the modules produced to date are in the areas of physics and biology and are in various states of test, revision, and final forms. A test version of the physics module, ESCAPE, is presented here. Future issues of Creative Computing will carry a complete article about the Chelsea Science Simulation Project and, hopefully, additional modules if enough readers want them. Write and let us know your interests.

Q2 So far we have made at least two assumptions in this discussion. Do you know what they are?

Suppose that a space probe were fired directly from the earth's surface, and fired fast enough to travel a great distance. Would the kind of argument used above still work? In particular, would the potential energy go on increasing uniformly with distance from the earth?

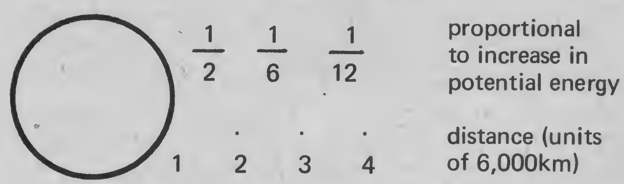
You probably know that as one travels away from the earth the earth's pull, gravity, gets weaker. The *force* of gravity on an object follows the so-called inverse-square law.

The radius of the earth is about 6,000km. The inverse-square law says that at twice this distance (12,000km) from the earth's centre the force is only $(\frac{1}{2})^2 = \frac{1}{4}$ of what it is at the surface of the earth; at three times the distance the force is $(1/3)^2 = 1/9$th, and so on.



The force pulling the object back towards earth steadily decreases as it gets further and further away. What about the potential energy? It goes on increasing — but does it still increase steadily, in proportion to the distance from the earth's centre?

Theory says it wouldn't. It says that potential energy will increase more and more slowly as the probe gets further from the earth's centre and the gravity force gets weaker. The increase in potential energy in going from a distance 6,000km above the earth's surface to a distance 12,000km is less than the increase in going from the surface to a distance 6,000km. And again, the potential energy increase in going from 12,000 to 18,000 is less than the change in going from 6,000km to 12,000km, and so on.



And so the kinetic energy is "used up" less and less quickly as the probe gets further away from earth.

Now this raises a question. Would it be possible to give the probe so much kinetic energy that it would never be completely used up (transformed to potential energy)? If so the probe would just go on indefinitely moving farther and farther away from earth. Perhaps it seems a reasonable idea that since the kinetic energy is transformed less and less quickly there will always be some left (if you eat half a cake today, then tomorrow eat half the remaining half to leave you a quarter, then eat half the quarter . . . you will, in theory anyway, always have *some* cake left).

On the other hand however far from the earth the probe is (assume the earth is the only body in the Universe) there is always *some* gravitational force on it. And so you might argue that no matter how far away the probe is, nor how fast it is still moving, it will never be able to escape completely, never to return, because the earth will always be pulling it back, however weakly.

Perhaps both these arguments sound reasonable to you. But they can't both be right! And one can't decide between them except by doing calculations to find out. (Or by doing an experiment — can one fire something off into space so that it never comes back?)

This is where the computer can help. It has been programmed to answer different questions to do with launch velocity and distance travelled. Of course, it has to assume a theoretical basis on which to make these calculations. It has been programmed to assume that the gravitational force falls off according to the inverse-square law, and that there is no air resistance. Other simplifying assumptions have been made, such as:

- ▶ the earth is the only body in the Universe
- ▶ the probe is always launched vertically upwards

The computer can tell you:

A What launch velocity is needed to fire a probe to a particular chosen distance.

B What velocity the probe has left at certain distances, for a particular launch velocity.

C At what distance does the probe stop and turn around and begin returning to earth, for a chosen launch velocity.

Your job is to use the computer to help answer the problem posed earlier: "Is it possible to fire something so fast that it will never return?"

Decide which of the three questions (A, B or C above) would be the most helpful.

For each question you will have to give some information. For example, if you ask for question C to be answered you will have to choose values for the launch velocity of the probe, and its mass.

To help you find the answer to the problem use the computer to answer these specific questions:

Q3 Does the mass of the probe affect how far it will travel for a given launch velocity, or what launch velocity it must be given if it is to reach a certain distance; If so, how?

Q4 Suppose the last question were asked about energy instead of velocity — would the answer be the same? (Does the mass of the probe affect how far it will travel for a given initial kinetic energy, or what energy it must be given if it is to reach a certain distance?)

Q5 The earth's radius, R is about 6,000km (6×10^6m). What launch velocity is needed to carry a probe from the earth's surface to a distance R from it?

The next question refers to the distance from earth to the moon, the sun, etc., but in answering it you should assume, as before, that the earth is the only body in the Universe.

Q6 What launch velocity is needed to get a probe as far as the moon (earth — moon distance is about 380×10^6m)?

— as far away from the earth as the sun is (150×10^9m)?

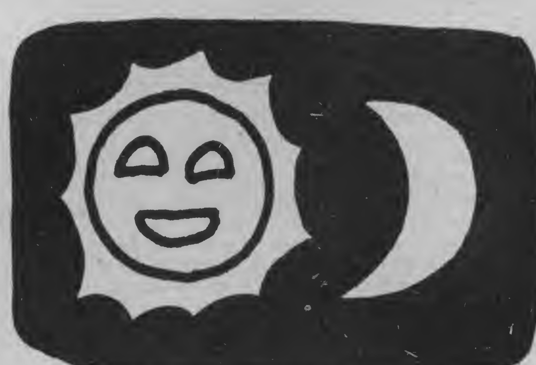
— as far as Pluto, the furthest known planet (about 60×10^{12}m)?

— as far as the nearest star (about 40×10^{15}m)?

— as far as you like.

Sample output is shown only for Question A. Run the program yourself to see how it works for Questions B and C.

```
WHAT OUTPUT DO YOU WANT?
TYPE 1 FOR LAUNCH VELOCITY TO REACH A CHOSEN HEIGHT
 OR  2 FOR VELOCITY AT DIFFERENT HEIGHTS ,FOR A CHOSEN
       LAUNCH VELOCITY
 OR  3 FOR HEIGHT REACHED FOR A CHOSEN LAUNCH VELOCITY
?1


TO INTERRUPT THE PROGRAM TO GET DIFFERENT OUTPUT
TYPE  0  WHEN YOU ARE ASKED TO INPUT MASS, OR HEIGHT

MASS OF PROBE (KG)   HEIGHT (M)   LAUNCH VELOCITY (M/S)

?1000
                     ?150000
                                       1700.22

?0


TO GET ANOTHER OUTPUT AS LISTED ABOVE TYPE 1,2 OR 3

TYPE 4  IF YOU WOULD LIKE A TABLE OR GRAPH SHOWING
THE LAUNCH VELOCITY NEEDED TO REACH CHOSEN HEIGHTS

TYPE 0 TO END THE PROGRAM
?4


TYPE  (1)  FOR TABLE
 OR   (2)  FOR GRAPH
?2

TOTAL HEIGHT  (METRES)=?400000



 HT.                 LAUNCH VELOCITY(M/S)------->
(M)            0                                                2724.71
               *---------+---------+---------+---------+---------+
               +
               +
               +
               +
40000.         -         *
               +
               +
               +
               +
80000.         -               *
               +
               +
               +
               +
120000.        -                    *
               +
               +
               +
               +
160000.        -                         *
               +
               +
               +
               +
200000.        -                            *
               +
               +
               +
               +
240000.        -                                *
               +
               +
               +
               +
280000.        -                                   *
               +
               +
               +
               +
320000.        -                                    *
               +
               +
               +
               +
360000.        -                                       *
               +
               +
               +
               +
400000.        -                                          *

TYPE 1 IF YOU WANT TO RE-RUN THIS PART OF THE PROGRAM
?0
```



Another new game from Creative Computing . . .

SEAWAR

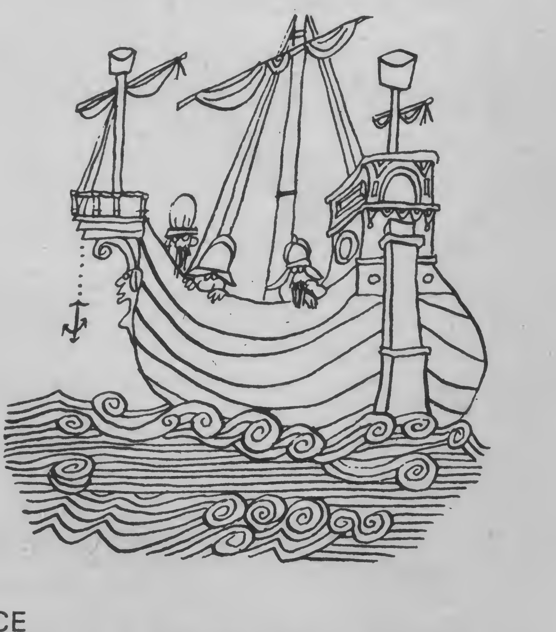
DESCRIPTION

You are the commander of a fleet of ships operating in enemy territory. Your task force consists of 9 ships, and the enemy has 9 ships. Whoever sinks all of the opponent's ships first wins the campaign.

You, as the commander, must provide the angle of elevation at which the guns will be fired, neglecting air resistance. Your instruments will read the range to the target, and the initial velocity is held constant at about 675 meters per second. Since there is a 7 second time limit for entering the angle of elevation, you will have to act quickly!

PROGRAMMING NOTES

1. The program as listed will run on a Hewlett-Packard 2000F system, but it can be adapted to other computer systems using BASIC.
2. Statement 550 allows 7 seconds to input the angle of elevation. When the game is initially introduced, you may prefer to extend this time to 15-20 seconds.
3. Lines 210 and 1100 have the bell enclosed in the quotation marks.
4. The initial velocity may be varied by changing line 700.



SOURCE

The origin of SEAWAR is unknown. It was revised and submitted to us by David S. Paxton, Fairfax, Virginia. It was further revised and the writeup prepared by Mary T. Dobbs, Mathematics and Science Center, Glen Allen, Virginia.

USING THE PROGRAM

SEAWAR will help you learn about the paths of projectiles and what happens as the angle of elevation varies.

1. First, what do you think the path of the projectile looks like. Make a sketch. (If you're still not sure, do some research in the library — it will help you win the battle, commander!)
   a. What angle of elevation do you think will give the maximum range?
   b. What will happen if you fire the guns at 0°?
   c. What will happen to the projectile if you fire it straight up?
2. After becoming proficient at winning the battle, change the initial velocity of the projectile. How does this affect the range?
3. For a more sophisticated look at projectiles, check out these programs.
   **"The Paris Gun," as listed and described in the Hewlett-Packard Users Group Newsletter, Nov-Dec 1974.
   **PRJTL, Huntington I Simulation Programs — PHYSICS, published by Digital Equipment Corporation.

SAMPLE RUN

```
RUN
SEAWAR

YOU COMMAND A FLEET OF SHIPS OPERATING IN ENEMY TERRITORY!!!
DO YOU NEED INSTRUCTIONS?YES
YOU TELL YOUR GUN CREWS THE ELEVATION TO SET THEIR GUNS.
ELEVATION IS IN DEGREES FROM 0 TO 360.
YOUR TASK FORCE CONSISTS OF 3 DESTROYERS, 2 CRUISERS,
2 BATTLESHIPS, AND 2 HEAVY AIRCRAFT CARRIERS.
THE ENEMY HAS 9 SHIPS FOR HIS DEFENSE.
IF YOU SUCCEED IN SINKING ALL HIS SHIPS BEFORE HE SINKS
YOURS, YOU HAVE WON.  HOWEVER, IF HE SINKS ALL YOUR SHIPS
BEFORE YOU HAVE DEFEATED HIM, YOU HAVE LOST!!
LET US BEGIN!!!

YOUR FLAGSHIP HAS DETECTED A U-BOAT APPROACHING AT 5 FATHOMS.
YOUR SUBMARINE DETECTION EQUIMENT READS THE RANGE TO THE TARGET
AS 23175 METERS.
THE U-BOAT HAS COMMENCED FIRING TORPEDOES AT YOUR SHIPS.
HIS FIRST TORPEDO EXPLODED  65 METERS BEHIND YOUR SHIP.
WHAT ELEVATION ** 13
-----FIRE!!!
DEPTH CHARGE EXPLODED 2792 METERS SHORT OF TARGET.
THE ENEMY TORPEDO EXPLODED  57 METERS IN FRONT OF YOUR
SHIP.
WHAT ELEVATION ** 15
-----FIRE!!!
DEPTH CHARGE EXPLODED RIGHT ON TOP OF THAT BABY!!!

TARGET DESTROYED!!!      *2 **  ROUNDS EXPENDED.
YOU HAVE LOST 0 SHIPS, AND THE ENEMY HAS LOST 1.

YOUR FLAGSHIP REPORTS THE SIGHTING OF AN ENEMY 210 MM SHORE GUN
YOUR INSTRUMENTS READ THE RANGE TO THE TARGET AS 22539
METERS.
THE ENEMY 210 MM SHORE GUN IS FIRING ON YOUR SHIPS!
HIS FIRST ROUND FELL 425 METERS SHORT.
WHAT ELEVATION **
          ADMIRAL !!  YOU HAVE TO BE FAST IN THIS GAME!!
THE ENEMY 210 MM SHORE GUN SANK ONE OF YOUR DESTROYERS!!
WHAT ELEVATION ** 14
-----FIRE!!!
SHOT FELL   710 METERS SHORT OF TARGET.
THE ENEMY ROUND FELL 136 METERS SHORT.
WHAT ELEVATION ** 14.5
-----FIRE!!!
  ** BOOM **

TARGET DESTROYED!!!      *2 **  ROUNDS EXPENDED.
YOU HAVE LOST 1 SHIPS, AND THE ENEMY HAS LOST 2.
```

```
YOUR FLAGSHIP REPORTS THE SIGHTING OF AN ENEMY AIRCRAFT CARRIER
YOUR INSTRUMENTS READ THE RANGE TO THE TARGET AS 39604
METERS.
WHAT ELEVATION ** 29
-----FIRE!!!
SHOT FELL   171 METERS SHORT OF TARGET.
THE ENEMY ROUND FELL 263 METERS SHORT.
WHAT ELEVATION ** 29.3
-----FIRE!!!
  ** BOOM **

TARGET DESTROYED!!!      *2 **  ROUNDS EXPENDED.
YOU HAVE LOST 4 SHIPS, AND THE ENEMY HAS LOST 9.
          ********  PEACE  ********


YOU FIRED 20 ROUNDS.  THE ENEMY FIRED 19 ROUNDS.
YOU HAVE DECIMATED THE ENEMY..........THAT'S NICE
THE BATTLE IS OVER.............YOU WIN!!!!
```

```
LIST
SEAWAR

10  PRINT "YOU COMMAND A FLEET OF SHIPS OPERATING IN ENEMY TERRITORY!!!"
20  PRINT "DO YOU NEED INSTRUCTIONS";
30  DIM Q$[12]
40  INPUT Q$
50  IF Q$="YES" THEN 90
60  IF Q$="NO" THEN 170
70  PRINT "INPUT 'YES' OR 'NO'"
80  GOTO 40
90  PRINT "YOU TELL YOUR GUN CREWS THE ELEVATION TO SET THEIR GUNS."
100  PRINT "ELEVATION IS IN DEGREES FROM 0 TO 360."
119  PRINT "YOUR TASK FORCE CONSISTS OF 3 DESTROYERS, 2 CRUISERS,"
120  PRINT "2 BATTLESHIPS, AND 2 HEAVY AIRCRAFT CARRIERS."
130  PRINT "THE ENEMY HAS 9 SHIPS FOR HIS DEFENSE."
140  PRINT "IF YOU SUCCEED IN SINKING ALL HIS SHIPS BEFORE HE SINKS"
150  PRINT "YOURS, YOU HAVE WON.  HOWEVER, IF HE SINKS ALL YOUR SHIPS"
160  PRINT "BEFORE YOU HAVE DEFEATED HIM, YOU HAVE LOST!!"
170  PRINT "LET US BEGIN!!!"
180  DIM Z$[20],D$[40],I$[10],P$[72],M$[40],N$[40]
190  A=O=S1=S2=S=P1=P2=P4=0
200  REM SELECTS NAME OF ENEMY SHIP
210  PRINT ""
220  READ Z$
230  A=A+1
240  GOTO 320
250  RESTORE
260  IF O=9 OR A=9 THEN 840
270  FOR X=1 TO A
280  READ Z$
290  NEXT X
300  READ Z$
310  A=A+1
315  REM SELECTS BATTLE MODE
320  IF Z$="AIRCRAFT CARRIER" THEN 390
330  IF Z$="U-BOAT" THEN 2000
340  IF Z$="TORPEDO BOAT" THEN 360
350  LET P=1
360  GOTO 400
390  RESTORE
399  REM BEGINS BATTLE WITH SIGHTING AND READING
400  PRINT
405  PRINT "YOUR FLAGSHIP REPORTS THE SIGHTING OF AN ENEMY ";Z$
410  T=43000.-30000*RND(0)+(RND(0)*10)*.987654+102
420  IF T<10000 THEN 410
430  S=P2=0
440  T=INT(T)
450  IF Z$="U-BOAT" THEN 2030
460  PRINT  USING 470;T
470  IMAGE"YOUR INSTRUMENTS READ THE RANGE TO THE TARGET AS ",DDDDD
475  PRINT "METERS."
480  IF P=1 THEN 1480
490  IF S>4 THEN 510
500  GOTO 540
505  REM AFTER 5 TRYS BY US TARGET MOVES OUT OF RANGE
510  PRINT "ALL RIGHT, BAD SHOT, THE TARGET HAS MOVED OUT OF"
520  PRINT "RANGE !!!   LET'S TRY IT AGAIN !!!"
525  S1=S1+S
530  GOTO 320
535  REM INPUT ANGLE OF ELEVATION
540  PRINT "WHAT ELEVATION ** ";
550  ENTER 7,L,B
551  PRINT
560  IF L#-256 THEN 590
570  PRINT "             ADMIRAL !!  YOU HAVE TO BE FAST IN THIS GAME!!"
580  GOTO 1590
590  PRINT "-----FIRE!!!"
600  S=S+1
620  IF B>360 THEN 1410
630  IF B<0 THEN 750
640  IF B=0 THEN 770
650  IF B=90 THEN 980
660  IF B>330 THEN 770
670  IF B>180 THEN 1370
680  IF B>150 THEN 1300
690  IF B>90 THEN 1020
699  REM DISTANCE FROM TARGET CALCULATED
700  V1=675.285
705  E=INT(T-(V1^2/9.80665*SIN(2*B/57.3)))
710  IF ABS(E) <= 100 THEN 1050
720  IF E>100 THEN 1200
730  IF E<-100 THEN 1250
740  REM "GOOF"SHOTS
750  PRINT "GUN BACKFIRED, KILLING CREW!"
760  GOTO 820
770  PRINT "WHAT ARE YOU TRYING TO DO?  KILL SOME FISH?  THE SHELL"
780  PRINT "EXPLODED UNDER WATER FIFTY METERS FROM YOUR SHIP!!!"
790  GOTO 1590
820  PRINT "             ADMIRAL PLEASE !!!!"
830  GOTO 1590
839  REM PEACE AND WINNER
840  PRINT "             ********  PEACE  ********"
850  PRINT
860  PRINT
870  PRINT
880  PRINT  USING 890;S1,S2
890  IMAGE "YOU FIRED ",DD," ROUNDS.  THE ENEMY FIRED ",DD," ROUNDS."
900  IF O=9 THEN 920
910  IF A=9 THEN 950
920  PRINT "ALL OF YOUR SHIPS HAVE BEEN SUNK.   SO SORRY"
930  PRINT "THE BATTLE IS OVER..............THE ENEMY WINS."
940  GOTO 2220
950  PRINT "YOU HAVE DECIMATED THE ENEMY..........THAT'S NICE"
960  PRINT "THE BATTLE IS OVER.............YOU WIN!!!!"
970  GOTO 2220
975  REM  "GOOF" SHOTS
980  PRINT "YOU IDIOT!! YOU SHOT STRAIGHT UP!!, AND THE SHELL"
990  PRINT "LANDED ON YOUR OWN GUN POSITION, DESTROYING IT!!!"
1000  GOTO 1590
1020  PRINT "HEY STUPID, YOU'RE FIRING ON YOU OWN SHIPS!!!"
1030  GOTO 1590
1040  REM ENEMY SHIP SUNK
1050  IF Z$="U-BOAT" THEN 1070
1060  GOTO 1090
1070  PRINT "DEPTH CHARGE EXPLODED RIGHT ON TOP OF THAT BABY!!!"
1080  GOTO 1100
1090  PRINT "  ** BOOM **"
1100  PRINT ""
1110  M$="TARGET DESTROYED!!!       **"
1120  N$=" **  ROUNDS EXPENDED."
1130  PRINT  USING 1140;M$,S,N$
1140  IMAGE 26A,D,21A
1141  PRINT  USING 1142;O,A
1142  IMAGE "YOU HAVE LOST ",D," SHIPS, AND THE ENEMY HAS LOST ",D,"."
1150  S1=S1+S
1160  P=0
1190  GOTO 250
1200  IF Z$="U-BOAT" THEN 2130
1210  PRINT  USING 1220;ABS(E)
1220  IMAGE "SHOT FELL ",5D," METERS SHORT OF TARGET."
1230  GOTO 1590
1250  IF Z$="U-BOAT" THEN 2160
1260  PRINT  USING 1270;ABS(E)
1270  IMAGE "SHELL OVERSHOT TARGET BY ",6D," METERS."
1280  GOTO 1590
1300  REM "GOOF" SHOTS
1310  PRINT "   YOU SHOT A PROJECTILE, INTO THE AIR,"
1320  PRINT "   IT FELL TO THE WATER, YOU KNOW NOT WHERE."
1330  PRINT "BUT I DO, YOU IDIOT, YOU JUST SANK YOUR OWN FLEET TANKER!!"
1340  S1=S1+1
1350  IF P=1 THEN 1590
1360  GOTO 490
1370  PRINT "WHAT ARE YOU TRYING TO DO?? DRILL A NEW HATCH??  THE SHELL"
1380  PRINT "EXPLODED IN YOUR SHIP, DESTROYING IT!!!"
1385  O=O+1
1386  IF O=9 THEN 840
1390  IF P=1 THEN 1590
1400  GOTO 820
1410  PRINT "WHERE DID U LEARN TO TYPE? ";B;"DEGREES EXCEEDS 360 BY"
1420  PRINT B-360;" DEGREES."
1430  S1=S1+1
1440  IF P=1 THEN 1590
1450  GOTO 490
1480  PRINT "THE ENEMY ";Z$;" IS FIRING ON YOUR SHIPS!"
1490  P4=1234*RND(RND(0))+(RND(0)*10)
1500  IF P4>500 THEN 1490
1510  IF P2=1 THEN 1600
1520  IF INT(P4)<100 THEN 1800
1530  IF Z$="U-BOAT" THEN 2100
1540  PRINT  USING 1550;INT(P4)
1550  IMAGE"HIS FIRST ROUND FELL ",DDD," METERS SHORT."
1560  LET S2=S2+1
1570  GOTO 490
1590  IF P2=1 THEN 1490
1600  P1=1250*RND(RND(0))+(RND(0)*10)
1610  IF P1>P4 THEN 1600
1620  IF P1<(P4-400) THEN 1600
1630  IF P1<100 THEN 1710
1640  LET P4=P1
1650  S2=S2+1
1660  IF Z$="U-BOAT" THEN 2190
1670  PRINT  USING 1680;INT(P1)
1680  IMAGE"THE ENEMY ROUND FELL ",DDD," METERS SHORT."
1700  GOTO 490
1710  S2=S2+1
1720  LET P2=1
1730  GOSUB 1850
1750  PRINT "THE ENEMY ";Z$;" SANK ";D$
1760  O=O+1
1770  IF O=9 THEN 840
1780  IF D$="YOUR LAST BATTLESHIP!!" THEN 840
1790  GOTO 490
1800  P2=1
1810  GOSUB 1850
1820  PRINT "IN FACT, HE JUST SANK ";D$
1830  O=O+1
1840  GOTO 1770
1850  RESTORE
1860  FOR C=1 TO (9+O)
1870  READ D$
1880  NEXT C
1890  READ D$
1920  DATA "U-BOAT","210 MM SHORE GUN","70,000 TON CRUISER"
1930  DATA "BATTLESHIP","TORPEDO BOAT","HEAVY FRIGATE"
1940  DATA "E-TYPE DESTROYER","GUIDED-MISSILE SHIP","AIRCRAFT CARRIER"
1950  DATA "ONE OF YOUR DESTROYERS!!","YOUR HEAVY CRUISER!!"
1960  DATA "ANOTHER OF YOUR DESTROYERS!!","ONE OF YOUR BATTLESHIPS!!"
1970  DATA "YOUR LAST DESTROYER!!","YOUR AIRCRAFT CARRIER!!"
1975  DATA "YOUR LIGHT CRUISER!!","YOUR LAST AIRCRAFT CARRIER!!"
1980  DATA "YOUR LAST BATTLESHIP!!"
1990  RETURN
2000  PRINT "YOUR FLAGSHIP HAS DETECTED A U-BOAT APPROACHING AT 5 ";
2005  PRINT "FATHOMS."
2010  P=1
2020  GOTO 410
2030  PRINT "YOUR SUBMARINE DETECTION EQUIMENT READS THE RANGE TO THE";
2031  PRINT "TARGET"
2040  T=INT(T-1500)
2050  IF T<0 THEN 410
2060  PRINT  USING 2070;T
2070  IMAGE "AS ",5D," METERS."
2080  PRINT "THE U-BOAT HAS COMMENCED FIRING TORPEDOES AT YOUR SHIPS."
2090  GOTO 1490
2100  PRINT  USING 2110;(INT(P4)-50)
2110  IMAGE "HIS FIRST TORPEDO EXPLODED ",3D," METERS BEHIND YOUR SHIP."
2120  GOTO 1560
2130  PRINT  USING 2140;ABS(E)
2140  IMAGE"DEPTH CHARGE EXPLODED ",4D," METERS SHORT OF TARGET."
2150  GOTO 1590
2160  PRINT  USING 2170;ABS(E)
2170  IMAGE "DEPTH CHARGE EXPLODED ",4D," METERS AFT OF TARGET."
2180  GOTO 1590
2190  PRINT  USING 2200;(INT(P1)-50)
2200  IMAGE "THE ENEMY TORPEDO EXPLODED ",3D," METERS IN FRONT OF YOUR"
2201  PRINT "SHIP."
2210  GOTO 490
2220  END
```

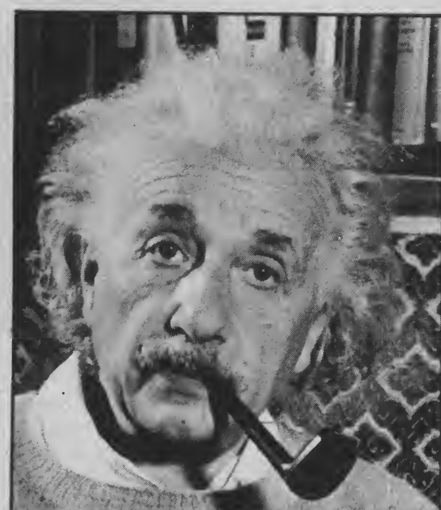
Relativity for computers: All arithmetic

Three hundred years ago Isaac Newton discovered that to solve the problems of universal gravitation he had set himself he had to invent the branch of mathematics called calculus. Eighty years ago, in formulating special relativity, Einstein caused a revolution in physics by making mass a variable and uniting space and time, but his equations, though changing the content of Newton's, followed Newton's calculus lead. Calculus, in fact dominates classical and most modern physics. For three centuries it has been the physicist's best mathematical friend.

But now the world has digital computers. Digital computers can do calculations with incredible speed. They have prodigious memories. People want to use them wherever a lot of computation must be done in a short time. But computers are not as smart as Newton, let alone Einstein, or even a college freshman. They can't do calculus. They can only do arithmetic, but they do it very fast.

So a group at the Computer Sciences Department of the University of Wisconsin at Madison under the direction of Donald Greenspan is working out a reformulation of Newton's and Einstein's work, not conceptually, but mathematically, trying to get it on an arithmetic basis so a computer can handle it. They have been successful with Newtonian classical mechanics and fluid dynamics. Greenspan's latest paper (UW's Computer Sciences Technical Report #232) details the first part of a method for special relativity. Later they hope to go to general relativity.

The sticking points in calculus are the concepts of limit and continuity. Limit is a way of getting around Zeno's paradox, which so bedeviled ancient Greek mathematicians, and showing that an infinite succession of infinitely small steps can in fact lead up to but never surpass a limit that is nevertheless a finite number. Continuity is a property of the sort of equations called functions. A function sets up a relation between an (independent) variable quantity or quantities and another (dependent) variable, so that, knowing the value of the independent variable, one can calculate the dependent one. A function is continuous at a point if an infinitesimal change in the independent variable yields a new finite value for the dependent variable that is only infinitesimally different from the previous one. These two concepts are the keys to the development of integral and differential calculus, and when they had been evolved, 18th-, 19th- and 20th-century physics and engineering were on their way.



If Newton looks disapproving and Einstein surprised, it may be because the calculus is being calculated out of their theoretical derivations.

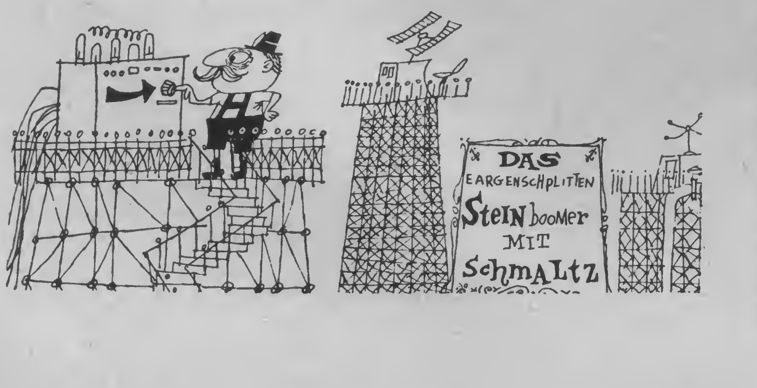
What the Wisconsin group has achieved is a way of replacing the infinitesimal steps with finite ones—finite-difference equations— that nevertheless come out to the same thing, in the limit, as the continuous processes of calculus. And it works in practice. "What we do runs," says Greenspan. "Everything we do must go on the computer. Doing it in theory is not enough."

Greenspan says the method should be of wide use to physicists and engineers with all kinds of mechanical, fluid dynamical and special relativistic computations to do. But it is much more than that.

As they were working through the method, the group found, in Greenspan's words, that they were doing physics arithmetically. The conservation laws that define and characterize Newtonian and Einsteinian physics come out of the arithmetic procedure. It is thus not merely a crutch for the computer but an alternate way of deriving and justifying the physically meaningful mathematical statements that have traditionally been gotten by calculus methods. And that could work an important change in high-school curricula.

Calculus—for reasons considered good and sufficient by mathematics teacher—has traditionally been reserved for the first or even second year of college. High-school physics teaching has suffered from the lack. Much of its content had to be brought in on faith or justified by handwaving methods. The new formulation could provide a way of making high-school physics more interesting by deriving its contents in a satisfactory way that the pupils could follow. Greenspan is enthusiastic about the idea of a change, but also extremely realistic. "You have 300 years of vested interest in calculus to overcome," he says. □

Nuclear Research	Das Whizkidden grupe.
Preliminary Design	Das Uppen-das-klaudsen grupe.
Administration	Das Oudtgeschmardten grupe.
Rocket Engine	Firenschpitter mid schmoken-und-schnorten.
Design	Das Raundscholder und Reddischeiz grupe.
Project Engineer	Das Schwettenoudter.
Management	Das Ulzerenbalden grupe.
Nuclear Warhead	Das eargeschplitten laudenboomer.
Hydrogen Device	Das eargeschplitten laudenboomer mit ein grosse hollengraund und alles kaput.



Contest!

LOOK! **CONTEST RULES** WIN $

Contests are open only to subscribers.

All contest entries must include the following information: your name, home address, home phone, age, school or company affiliation, contest problem number and your answer(s). Problems that call for the use of a computer must also include program listing, output, paper tape or card deck. Entries should be in BASIC or FORTRAN and should not use unique features of a particular compiler. If you are a high school student, you should also include the name, home address and phone number of your teacher.

Entries not including the above information will be automatically disqualified. Entries cannot be acknowledged.

The winner will receive a two-year subscription extension (or gift subscription) to *Creative Computing*. Second and third place winners will receive a one-year subscription extension (or gift subscription). Winners will be announced in *Creative Computing* two issues from now.

The winner, along with two associates (teacher, friends, etc.) will form a committee to judge the next contest. Contest entries may be judged on computer technique, uniqueness of the program, form of the program or output, length, running time, etc. Decisions of this committee are final.

All correct entries (winners or not) will go into a year-end drawing for $25, $15, and $10 prizes.

Deadline for entries: July 31, 1975

CONTEST PROBLEM 5. ROMAN NUMERAL

Write a program to convert any number from 1 to 3000 to its equivalent Roman numeral. The seven Roman symbols are:

M	1000
D	500
C	100
L	50
X	10
V	5
I	1

The rules for forming Roman numerals are:

1. If a symbol precedes one of smaller value, its value is added.
2. If a symbol precedes one of larger value, its value is subtracted; then the difference is added to the rest of the number.
3. Numbers are written as simply as possible using only C, X, and I as subtrahends. Some examples are MCMLXIV, 1964; DXLIX, 549.

Your program should accept as input the decimal number and output the Roman numeral. Convert the following numbers in your contest entry: 1, 14, 400, 549, 999, 1964, 1975, 2500, 2994, 3000.

CONTEST PROBLEM 6. SEVEN DIGITS

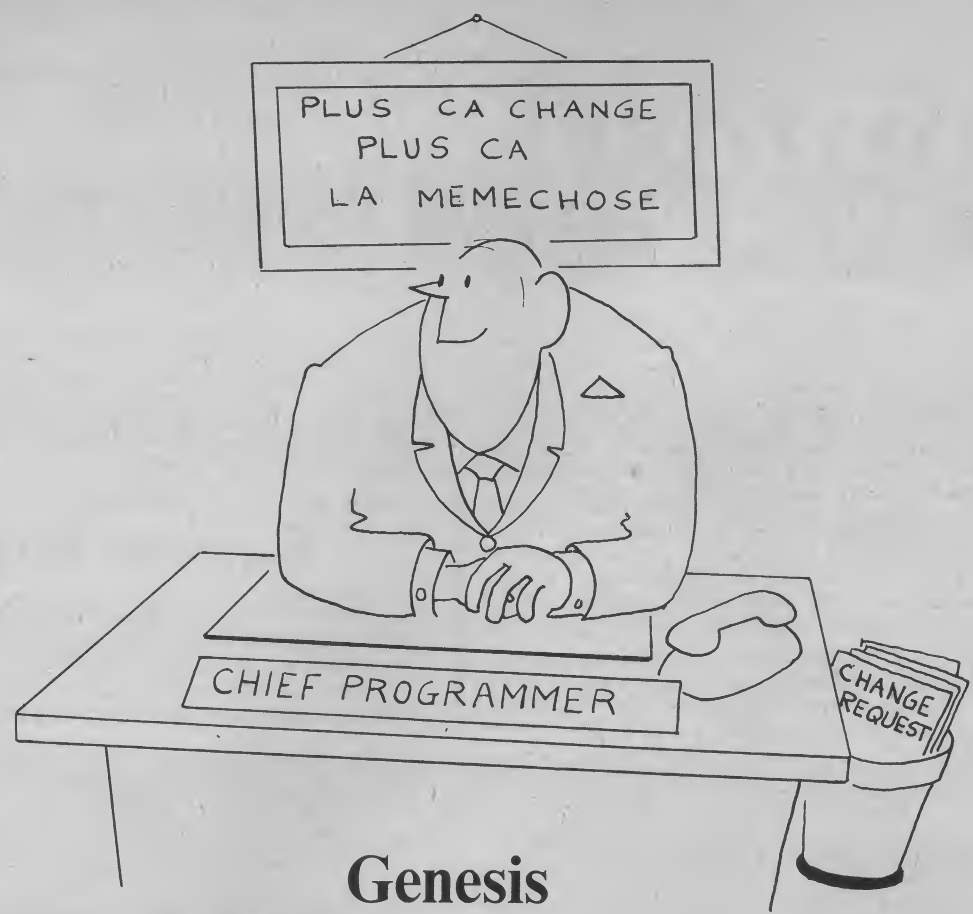
In the octonary (modulus 7) system, write a program to find the seven 7-digit squares which contain no duplicate digits. Here is one: 1242^2 = 1567204. Note: a trivial, "brute force" entry will not win.

CONTEST 2 RESULTS

Judged by Melinda Harp, Augusta, GA

1st	Marty Handlon, Henry Ford II School, Sterling Hts., MI
2nd	Joel Harrison, Livingston High School, NJ
3rd	Robert Marlan, Irondequoit HS, Rochester, NY
HM	Scott Davidson, W. New Mexico U., Silver City, NM
	James Gingerich, Bethany Christian HS, Goshen, IN
	Ronald Li, Ardsley High School, NY
	Mike Sullivan, Curtis Jr. HS, Sudbury, MA

Most of the programs used similar logic. Marty's was easy to follow, had a nice method for checking digits, and used the computer for testing that all conditions were met. Some of the others left out desirable checks on the numbers generated while having some other helpful features such as (1) Remarks and Comments, (2) Checks for unique solution and (3) General formulas for setting up and checking digits.



Genesis
Release 2.5

by Michael L. Coleman
with illustrations by Stew Burgess

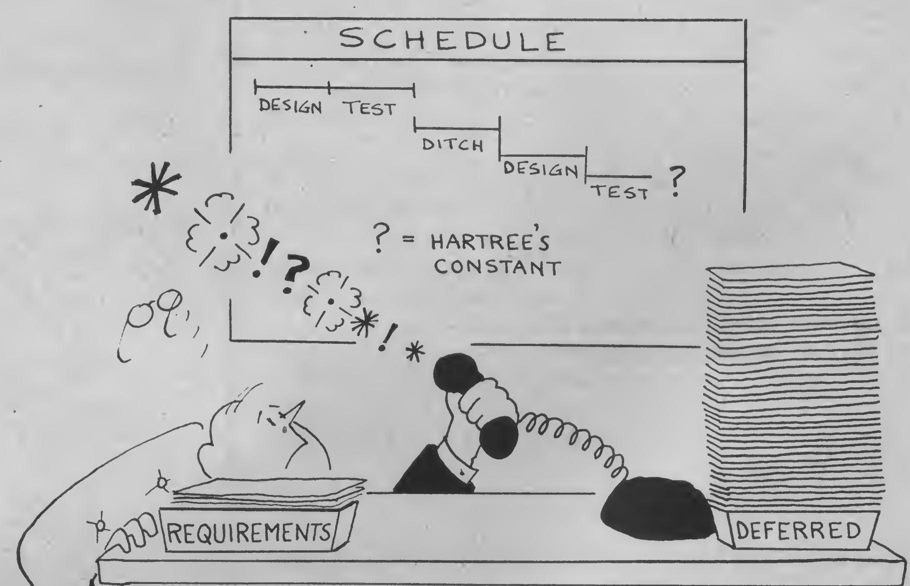
IN THE BEGINNING the Project Manager created the Programming Staff. The Programming Staff was without form and structure. And the Project Manager said, "Let there be Organization;" and there was Organization. And the Project Manager saw that Organization was good; and the Project Manager separated the workers from the supervisors, and he called the supervisors—"Management," and he called the workers—"Exempt."

And the Project Manager said, "Let there be a mission in the midst of the Organization, and let it separate the workers, one from another." And the Project Manager created the mission and he called it—"The System." And the Project Manager separated those who were to benefit from The System from those who were to build it. And he called the former—"Users," and he called the latter—"Programmers."

And the Project Manager said, "Let all the Programmers in the Organization be gathered together into one place, and let a Chief Programmer be brought up to lead them." And it was so. And the Project Manager saw that he was competent.

And the Project Manager said unto the Chief Programmer, "Create for me a schedule, so that I may look upon the schedule and know the Due Date."

And the Chief Programmer went among his staff and consulted with them. And the staff was divided into two parts, one part was called—"Ana-



lysts," and the other part was called—"Application Programmers." And the Analysts went back to their desks and estimated, as was their custom. And it came to pass that each Analyst brought his estimate to the Chief Programmer, whereupon he collected them, summarized them, and drew a PERT Chart.

And the Chief Programmer went unto the Project Manager and presented to him the estimate saying, "It shall take ten months." And the Project Manager was not pleased and said, "I have brought you up from the depths of the staff; you have not grasped the 'Big Picture.'" And the Project Manager hired consultants, and authorized overtime, and he said to the Chief Programmer, "Behold, see all that I have done! The Due Date will be in five months." The Chief Programmer was much impressed and went from before the Project Manager and proceeded to implement The System.

And the Chief Programmer sent his Analysts to the Users and said, "Let Specifications be written." And there were meetings, and lunches, and telephone calls. And the Specifications were written. And there was a Payday and the Happy Hour, one month.

And the Chief Programmer examined the Specifications and saw that they were too ambitious. And he separated the mandatory features from the optional features; and he called the mandatory features—"Requirements," and he called the optional features—"Deferred," and the Users called him names. And the Chief Programmer gave the Specifications to the Analysts and said, "Let the Requirements be analyzed and let the Files be designed." And it was so. And the Chief Programmer said, "Let the Software Houses put forth their Salesmen, and let us have a Data Management System." And it was so. The Software Houses brought forth all manner of Salesmen who presented their packages, and claimed wondrous things for them, each according to his own file structure. And it came to pass that a Data Management System was selected; and the Chief Programmer saw that it was good. And there was a Payday and the Happy Hour, a second month.

And the Chief Programmer said, "Let the System be divided into parts, and let each part be called a 'Module.' And let programming teams be formed and let each be assigned to write a Module." And it was so. And the Chief Programmer created the programming teams with two levels, a greater and a lesser; and he called the greater the "Senior Programmers," and he called the lesser the "Junior Programmers." And he gave the greater dominion over the lesser. And the Chief Programmer saw it was good. And the Junior Programmers saw it differently. And there was a Payday and the Happy Hour, a third month.

And the Chief Programmer said, "Let the programming be started and let much overtime be consumed, for there is but two months left." And the Programmers, both the Senior and the Junior, were much afraid, and they strove to please the Chief Programmer. And they flowcharted, and they coded, each in his own fashion. And the Chief Programmer looked upon the work and liked it not. And the Chief Programmer said, "Let there be a Standard;" and there was a Standard. And the Programmers looked upon the Standard and liked it not. And there

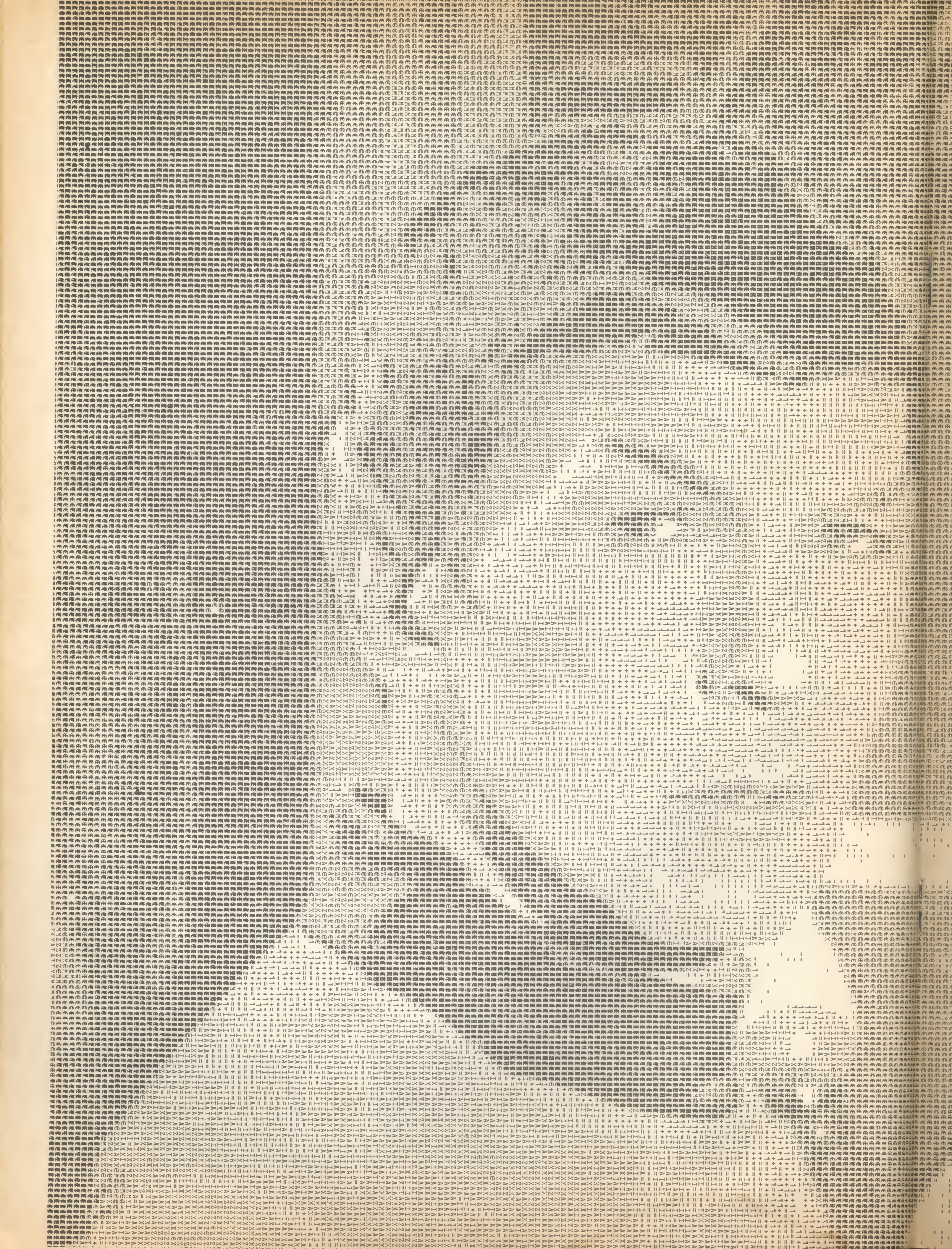
was a Payday and the Happy Hour, a fourth month.

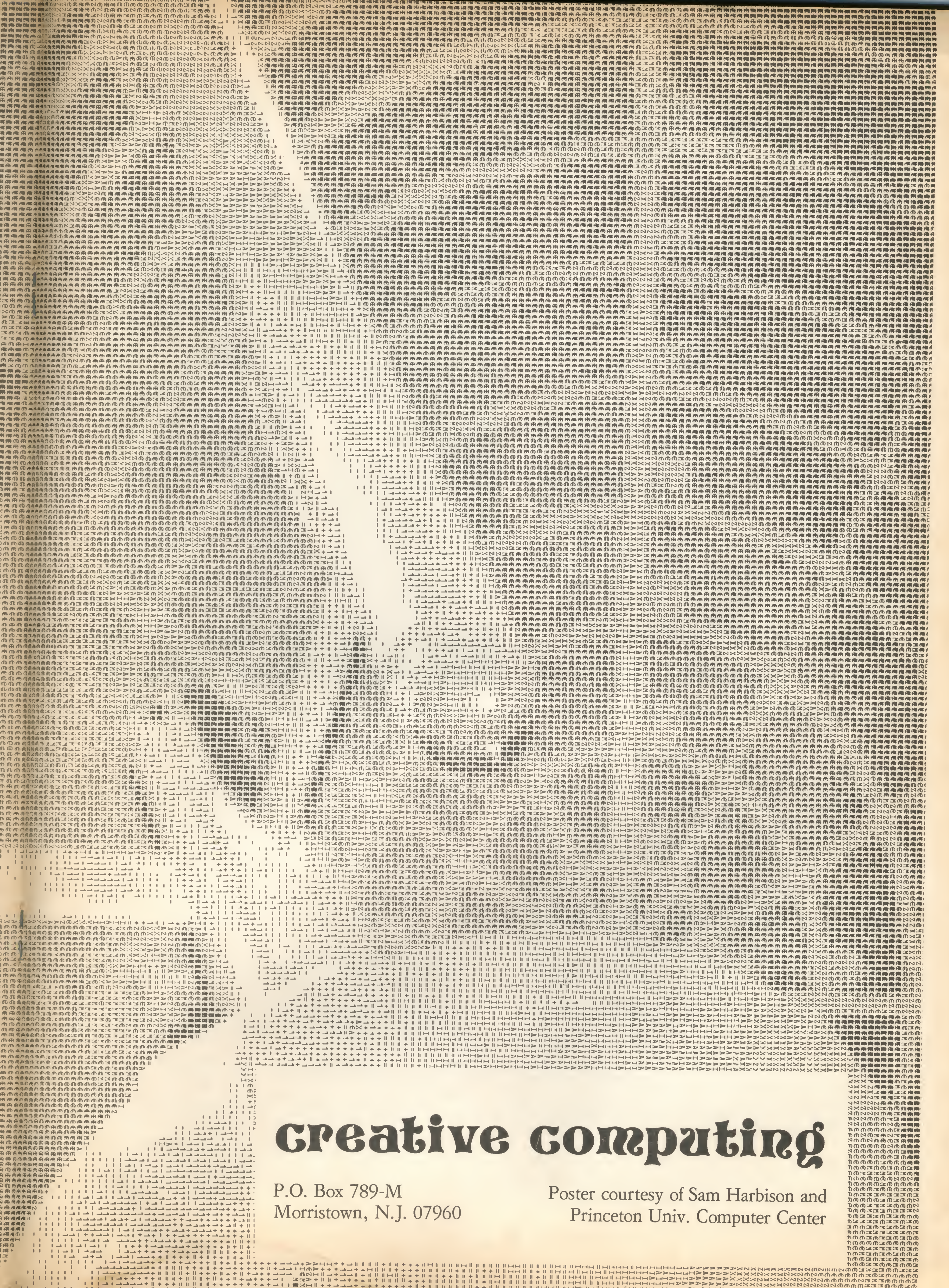
And the Chief Programmer said, "Let there be Progress Reports, so we can monitor and control;" and there were Progress Reports. And the Chief Programmer looked upon the Progress Reports and saw that the Due Date was not to be met. And the Chief Programmer arose, pressed his suit, shaved his beard, and went unto the Project Manager, and groveled. And the Chief Programmer pointed his fingers, and caused Blame to issue forth upon all manner of creatures who sold Hardware and Software. And the Chief Programmer asked for an Extension.

And the Project Manager was exceedingly angry, and cast doubts upon the Chief Programmer's ancestry; and uttered a multitude of threats. But it came to pass that an Extension was granted; and the Chief Programmer took the Extension back to the programming teams, and there was much rejoicing. And the programming of the modules was completed. And there was a Payday and the Happy Hour, a fifth month.

And the Chief Programmer said, "Let the Modules be integrated, one with another, so that System Testing may begin." And it was so. Two by two the Modules were integrated, one with another. And great difficulties were experienced, and many hours of overtime were used, and many cups of coffee were consumed. And it came to pass that System Testing was completed. And there was a Payday and the Happy Hour, a sixth month.

Then the Chief Programmer did go to the Project Manager and said unto him, "Behold, I bring you good tidings of a great joy which will come to all the Users; for on this day The System is completed." And suddenly there was with them a multitude of Users praising the Chief Programmer and saying, "Glory be to The System in the highest, but can you make this one small change"? ■





creative computing

P.O. Box 789-M
Morristown, N.J. 07960

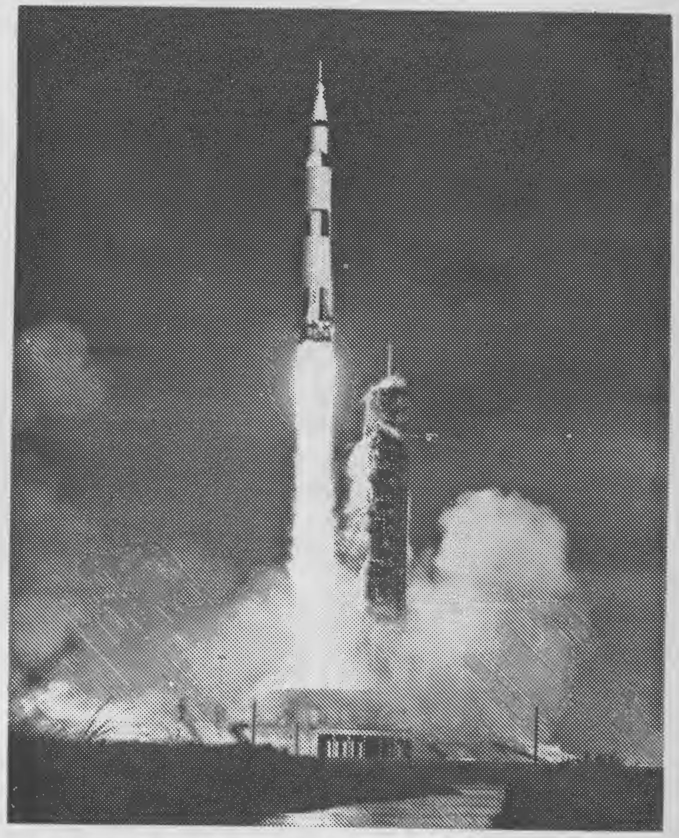
Poster courtesy of Sam Harbison and Princeton Univ. Computer Center

Another new game from Creative Computing . . .

SUPER STAR TREK

History

by David Ahl



Many versions of Star Trek have been kicking around various college campuses since the late sixties. I recall playing one at Carnegie-Mellon Univ. in 1967 or 68, and a very different one at Berkeley. However, these were a far cry from the one written by Mike Mayfield of Centerline Engineering and/or Custom Data. This was written for an HP2000C and completed in October 1972. It became the "standard" Star Trek in February 1973 when it was put in the HP contributed program library and onto a number of HP Data Center machines.

In the summer of 1973, I converted the HP version to BASIC-PLUS for DEC's RSTS-11 compiler and added a few bits and pieces while I was at it. Mary Cole at DEC contributed enormously to this task too. Later that year I published it under the name SPACWR (Space War — in retrospect, an incorrect name) in my book *101 Basic Computer Games.* It is difficult today to find an interactive computer installation that does not have one of these versions of Star Trek available.

Of course, a program like Star Trek does not stay static for long. Of the many extensions I have seen, by far the best is by Bob Leedom of Westinghouse Defense and Electronic Systems Center. It's presented here as SUPER STAR TREK.

Quadrant Nomenclature

Recently, certain critics have professed confusion as to the origin of the "quadrant" nomenclature used on all standard CG (Cartesian Galactic) maps. Naturally, for anyone with the remotest knowledge of history, no explanation is necessary; however, the following synopsis should suffice for the critics:

As every schoolboy knows, most of the intelligent civilizations in the Milky Way had originated galactic designations of their own choosing well before the Third Magellanic Conference*, at which the so-called "2^6 Agreement" was reached. In that historic document, the participant cultures agreed, in all two-dimensional representations of the galaxy, to specify 64 major subdivisions, ordered as an 8 x 8 matrix. This was partially in deference to the Earth culture (which had done much in the initial organization of the Federation), whose century-old galactic maps had always shown 16 major regions named after celestial landmarks of the Earth sky. Each of these regions was divided into four "quadrants," designated by ancient "Roman Numerals" (the origin of which has been lost).

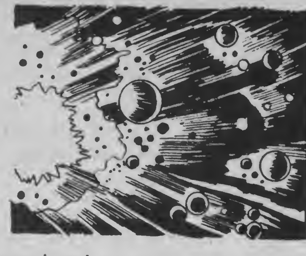
To this day, the official logs of starships originating on near-Earth starbases still refer to the major galactic areas as "quadrants."

The relation between the Historical and Standard nomenclatures is shown in the simplified CG map below.

*Conference held at Federation Starbase 1, Stardates 1016-1021.

	1	2	3	4	5	6	7	8
1	ANTARES				SIRIUS			
	I	II	III	IV	I	II	III	IV
2	RIGEL				DENEB			
	I	II	III	IV	I	II	III	IV
3	PROCYON				CAPELLA			
	I	II	III	IV	I	II	III	IV
4	VEGA				BETELGEUSE			
	I	II	III	IV	I	II	III	IV
5	CANOPUS				ALDEBARAN			
	I	II	III	IV	I	II	III	IV
6	ALTAIR				REGULUS			
	I	II	III	IV	I	II	III	I
7	SAGITTARIUS				ARCTURUS			
	I	II	III	IV	I	II	III	IV
8	POLLUX				SPICA			
	I	II	III	IV	I	II	III	IV

Super STAR TREK Rules and Notes



by Robert Leedom and David Ahl

1. OBJECTIVE: You are Captain of the starship "Enterprise" with a mission to seek and destroy a fleet of Klingon warships (usually about 17) which are menacing the United Federation of Planets. You have a specified number of stardates in which to complete your mission. You also have two or three Federation starbases for resupplying your ship.

2. You will be assigned a starting position somewhere in the galaxy. The galaxy is divided into an 8 x 8 quadrant grid. The astronomical name of a quadrant is called out upon entry into a new region. (See "Quadrant Nomenclature.") Each quadrant is further divided into an 8 x 8 section grid.

3. On a section diagram, the following symbols are used:

<*>	Enterprise	>!<	Starbase
†††	Klingon	*	Star

4. You have eight commands available to you. (A detailed description of each command is given in the program instructions.)

NAV	Navigate the Starship by setting course and warp engine speed.
SRS	Short-range sensor scan (one quadrant)
LRS	Long-range sensor scan (9 quadrants)
PHA	Phaser control (energy gun)
TOR	Photon torpedo control
SHE	Shield control (protects against phaser fire)
DAM	Damage and state-of-repair report
COM	Call library computer

5. Library computer options are as follows (more complete descriptions are in program instructions):

0	Cumulative galactic record
1	Status report
2	Photon torpedo course data
3	Starbase navigation data
4	Direction/distance calculator
5	Quadrant nomenclature map

6. Certain reports on the ship's status are made by officers of the Enterprise who appeared on the original Roddenberry TV Show — Spock, Scott, Uhura, Chekov, etc.

7. Klingons are non-stationary within their quadrants. If you try to maneuver on them, they will move and fire on you.

8. Firing and damage notes:

A. Phaser fire diminishes with increased distance between combatants.
B. If a Klingon zaps you hard enough (relative to your shield strength) he will generally cause damage to some part of your ship with an appropriate "Damage Control" report resulting.
C. If you don't zap a Klingon hard enough (relative to his shield strength) you won't damage him at all. Your sensors will tell the story.
D. Damage control will let you know when out-of-commission devices have been completely repaired.

9. Your engines will automatically shut down if you should attempt to leave the galaxy, or if you should try to maneuver through a star, a starbase, or — heaven help you — a Klingon warship.

10. In a pinch, or if you should miscalculate slightly, some shield control energy will be automatically diverted to warp engine control (if your shields are operational!).

11. While you're docked at a Starbase, a team of technicians can repair your ship (if you're willing for them to spend the time required — and the repairmen *always* underestimate . . .).

12. If, to save maneuvering time toward the end of the game, you should cold-bloodedly destroy a Starbase, you get a nasty note from Starfleet Command. If you destroy your *last* Starbase, you lose the game! (For those who think this is too harsh a penalty, delete lines 5360 - 5390, and you'll just get a "you dumdum!"-type message on all future status reports.)

13. End game logic has been "cleaned up" in several spots, and it is possible to get a new command after successfully completing your mission (or, after resigning your old one).

14. For those of you with certain types of CRT/keyboards setups (e. g. Westinghouse 1600), a "bell" character is inserted at appropriate spots to cause the following items to flash on and off on the screen:

- The Phrase "*RED*" (as in Condition:Red)
- The character representing your present quadrant in the cumulative galactic record printout.

15. PROGRAMMING NOTES: This version of Star Trek was created for a Data General Nova 800 system with 32k of core. So that it will fit, the instructions are separated from the main program via a CHAIN. One minor problem: RANDOMIZE (Statement 160) should be moved after the return from the chained instructions, say to statement 245. It appears that the program should run in DEC BASIC-PLUS but it's going to be fun/trouble/challenging to convert it to DEC 8-family, HP, Honeywell, or other machines.

16. Paper tapes and other things. Neither *Creative Computing* nor Westinghouse are in the business of making and distributing paper tapes. Please DO NOT write either of us (Bob Leedom, David Ahl) asking for paper tapes. If you want to write us about other things try:

Robert C. Leedom
3429 Rollingview Ct.
Ellicott City, Md. 21043

or

David H. Ahl
Creative Computing

```
0010 REM INSTRUCTIONS FOR "STREK" GAME
0020 REM VERSION "STINST2" 12/8/74
0030 DIM A$[5]
0040 FOR I=1 TO 9
0050   ON I THEN GOSUB 0240, 0360, 0540, 0640, 0720, 0780, 0860, 0910, 0960
0060   PRINT
0070   PRINT "(TO CONTINUE, HIT 'RETURN')"
0080   PRINT
0090   INPUT A$
0100 NEXT I
0110 PRINT "1.  WHEN YOU SEE 'COMMAND ?' PRINTED, ENTER ONE OF THE LEGAL"
0120 PRINT "      COMMANDS (NAV, SRS, LRS, PHA, TOR, SHE, DAM, COM,  OR XXX). "
0130 PRINT "2.  IF YOU SHOULD TYPE IN AN ILLEGAL COMMAND, YOU'LL GET A SHORT"
0140 PRINT "      LIST OF THE LEGAL COMMANDS PRINTED OUT. "
0150 PRINT "3.  SOME COMMANDS REQUIRE YOU TO ENTER DATA.   (FOR EXAMPLE, THE"
0160 PRINT "      'NAV' COMMAND COMES BACK WITH 'COURSE (1-9) ?'. )  IF YOU"
0170 PRINT "      TYPE IN ILLEGAL DATA (LIKE NEGATIVE NUMBERS), THAT COMMAND"
0180 PRINT "      WILL BE ABORTED. "
0190 PRINT
0200 PRINT "HIT (CAR RET) TO CONTINUE "
0210 INPUT A$
0220 CHAIN "STREK" THEN GOTO 0250
0230 REM *** EXIT HERE ***
0240 PRINT
0250 PRINT "      INSTRUCTIONS FOR  ** STAR TREK **"
0260 PRINT
0270 PRINT "THE GALAXY IS DIVIDED INTO AN 8 X 8 QUADRANT GRID, "
0280 PRINT "AND EACH QUADRANT IS FURTHER DIVIDED INTO AN 8 X 8 SECTOR GRID. "
0290 PRINT
0300 PRINT "   YOU WILL BE ASSIGNED A STARTING POINT SOMEWHERE IN THE GALAXY"
0310 PRINT "TO BEGIN A TOUR OF DUTY AS COMMANDER OF THE STARSHIP 'ENTERPRISE'."
0320 PRINT "YOUR MISSION: TO SEEK AND DESTROY THE FLEET OF KLINGON WARSHIPS"
0330 PRINT "WHICH ARE MENACING THE UNITED FEDERATION OF PLANETS. "
0340 PRINT
0350 RETURN
0360 PRINT
0370 PRINT "YOU HAVE THE FOLLOWING COMMANDS AVAILABLE TO YOU AS"
0380 PRINT "CAPTAIN OF THE STARSHIP:"
0390 PRINT "'NAV' COMMAND = WARP ENGINE CONTROL --"
0400 PRINT "  COURSE IS IN A CIRCULAR NUMERICAL          4  3  2"
0410 PRINT "  VECTOR ARRANGEMENT AS SHOWN.                . . . "
0420 PRINT "  INTEGER AND REAL VALUES MAY BE               ... "
0430 PRINT "  USED. (THUS, COURSE 1. 5 IS HALF-         5 ----- 1"
0440 PRINT "  WAY BETWEEN 1 AND 2. )                       ... "
0450 PRINT "                                              . . . "
0460 PRINT "  VALUES MAY APPROACH 9. 0, WHICH             6  7  8"
0470 PRINT "  ITSELF IS EQUIVALENT TO 1. 0. "
0480 PRINT "                                             COURSE"
0490 PRINT " ONE WARP FACTOR IS THE SIZE OF"
0500 PRINT " ONE QUADRANT.   THEREFORE, TO GET"
0510 PRINT " FROM QUADRANT 6,5 TO 5,5, YOU WOULD"
0520 PRINT " USE COURSE 3, WARP FACTOR 1. "
0530 RETURN
0540 PRINT "'SRS' COMMAND = SHORT RANGE SENSOR SCAN"
0550 PRINT "  SHOWS YOU A SCAN OF YOUR PRESENT QUADRANT. "
0560 PRINT "  SYMBOLOGY ON YOUR SENSOR SCREEN IS AS FOLLOWS:"
0570 PRINT "       <*> = YOUR STARSHIP'S POSITION"
0580 PRINT "       +++ = KLINGON BATTLE CRUISER"
0590 PRINT "       >!< = FEDERATION STARBASE (REFUEL/REPAIR/RE-ARM HERE!)"
0600 PRINT "        *  = STAR"
0610 PRINT "  A CONDENSED 'STATUS REPORT' WILL ALSO BE PRESENTED. "
0620 PRINT
0630 RETURN
0640 PRINT "'LRS' COMMAND = LONG RANGE SENSOR SCAN"
0650 PRINT "  SHOWS CONDITIONS IN SPACE FOR ONE QUADRANT ON EACH SIDE"
0660 PRINT "  OF THE ENTERPRISE (WHICH IS IN THE MIDDLE OF THE SCAN)"
0670 PRINT "  THE SCAN IS CODED IN THE FORM '###', WHERE THE UNITS DIGIT"
0680 PRINT "  IS THE NUMBER OF STARS, TENS DIGIT IS THE NUMBER OF STARBASES, "
0690 PRINT "  AND HUNDREDS DIGIT IS THE NUMBER OF KLINGONS. "
0700 PRINT "    EXAMPLE --  207 = 2 KLINGONS, NO STARBASES, 7 STARS. "
0710 RETURN
0720 PRINT "'PHA' COMMAND = PHASER CONTROL."
0730 PRINT "  ALLOWS YOU TO DESTROY THE KLINGON BATTLE CRUISERS BY"
0740 PRINT "  ZAPPING THEM WITH SUITABLY LARGE UNITS OF ENERGY TO"
0750 PRINT "  DEPLETE THEIR SHIELD POWER.  (REMEMBER, KLINGONS HAVE"
0760 PRINT "  PHASERS, TOO!)"
0770 RETURN
0780 PRINT "'TOR' COMMAND = PHOTON TORPEDO CONTROL."
0790 PRINT "  TORPEDO COURSE IS THE SAME AS USED IN WARP ENGINE CONTROL. "
0800 PRINT "  IF  YOU HIT THE KLINGON VESSEL, HE IS DESTROYED AND"
0810 PRINT "  CANNOT FIRE BACK AT YOU.  IF YOU MISS, YOU ARE SUBJECT TO"
0820 PRINT "  HIS PHASER FIRE. "
0830 PRINT "     NOTE: THE LIBRARY-COMPUTER ('COM' COMMAND) HAS AN"
0840 PRINT "       OPTION TO COMPUTE TORPEDO TRAJECTORY FOR YOU (OPTION 2). "
0850 RETURN
0860 PRINT "'SHE' COMMAND = SHIELD CONTROL."
0870 PRINT "  DEFINES NUMBER OF ENERGY UNITS TO BE ASSIGNED TO SHIELDS. "
0880 PRINT "  ENERGY IS TAKEN FROM TOTAL SHIP'S ENERGY.  NOTE THAT THE"
0890 PRINT "  TOTAL ENERGY INCLUDES SHIELD ENERGY. "
0900 RETURN
0910 PRINT "'DAM' COMMAND = DAMAGE CONTROL REPORT"
0920 PRINT "  GIVES STATE OF REPAIR OF ALL DEVICES, WHERE A NEGATIVE"
0930 PRINT "  'STATE OF REPAIR' SHOWS THAT THE DEVICE IS TEMPORARILY"
0940 PRINT "  DAMAGED. "
0950 RETURN
0960 PRINT "'COM' COMMAND = LIBRARY-COMPUTER"
0970 PRINT "  THE LIBRARY-COMPUTER CONTAINS SIX OPTIONS:"
0980 PRINT "  OPTION 0 = CUMULATIVE GALACTIC RECORD"
0990 PRINT "     WHICH SHOWS COMPUTER MEMORY OF THE RESULTS OF ALL PREVIOUS"
1000 PRINT "     LONG RANGE SENSOR SCANS. "
1010 PRINT "  OPTION 1 = STATUS REPORT"
1020 PRINT "     WHICH SHOWS THE NUMBER OF KLINGONS, STARDATES, AND STARBASES"
1030 PRINT "     REMAINING IN THE GAME. "
1040 PRINT "  OPTION 2 = PHOTON TORPEDO DATA"
1050 PRINT "     WHICH GIVES DIRECTIONS AND DISTANCE FROM THE ENTERPRISE"
1060 PRINT "     TO ALL KLINGONS IN YOUR QUADRANT"
1070 PRINT "  OPTION 3= STARBASE NAV DATA"
1080 PRINT "     WHICH GIVES DIRECTION AND DISTANCE TO ANY STARBASE"
1090 PRINT "     WITHIN YOUR QUADRANT"
1100 PRINT "  OPTION 4 = DIRECTION/DISTANCE CALCULATOR"
1110 PRINT "     WHICH ALLOWS YOU TO ENTER COORDINATES FOR"
1120 PRINT "     DIRECTION/DISTANCE CALCULATIONS. "
1130 PRINT "  OPTION 5 = GALACTIC 'REGION NAME' MAP"
1140 PRINT "     WHICH PRINTS THE NAMES OF THE SIXTEEN MAJOR GALACTIC
1150 PRINT "     REGIONS REFERRED TO IN THE GAME. "
1160 RETURN
1170 END
```

```
0010 REM [VERSION "STREK7", 1/12/75  RCL]
0020 REM
0030 REM
0040 REM ***          *** STAR TREK ***       ***
0050 REM *** SIMULATION OF A MISSION OF THE STARSHIP ENTERPRISE,
0060 REM *** AS SEEN ON THE STAR TREK TV SHOW.
0070 REM *** ORIGINAL PROGRAM BY MIKE MAYFIELD; MODIFIED VERSION
0080 REM *** PUBLISHED IN DEC'S "101 BASIC GAMES", BY DAVE AHL.
0090 REM ***    MODIFICATIONS TO THE LATTER (PLUS DEBUGGING) BY
0100 REM ***       BOB LEEDOM -- APRIL & DECEMBER 1974,
0110 REM ***  WITH A LITTLE HELP FROM HIS FRIENDS ...
0120 REM *** COMMENTS, EPITHETS, AND SUGGESTIONS SOLICITED --
0130 REM *** ADDRESS TO:   R. C. LEEDOM
0140 REM ***      WESTINGHOUSE DEFENSE & ELECTRONIC SYSTEMS CNTR.
0150 REM ***      BOX 746, M. S. 338   BALTIMORE, MD  21203
0160 RANDOMIZE
0170 PRINT  TAB(15); "* * * STAR TREK * * *"
0180 PRINT
0190 PRINT "DO YOU NEED INSTRUCTIONS (YES/NO)";
0200 DIM A$[20]
0210 INPUT A$
0220 IF A$<>"YES" THEN GOTO 0240
0230 CHAIN "STINST"
0240 REM  PROGRAM BEGINS HERE .....
0250 DIM Z$[72], Q$[72], R$[72], S$[72]
0260 DIM G1$[150], G2$[16], G3$[3], G4$[3]
0270 FOR I=1 TO 72
0280   LET Z$[I, I]=" "
0290   LET Q$[I, I]=" "
0300   LET R$[I, I]=" "
0310   LET S$[I, I]=" "
0320 NEXT I
0330 DIM G[8, 8], C[9, 2], K[3, 3], N[3], Z[8, 8]
0340 DIM O1$[40], C$[10]
0350 DIM A1$[20], T$[6]
0360 DIM D$[96], O3$[60]
0370 LET T=INT(RND(1)*20+20)*100
0380 LET T0=T
0390 LET T9=30
0400 LET D0=0
0410 LET E0=3000
0420 LET E=E0
0430 LET P=10
0440 LET P0=P
0450 LET S9=200
0460 LET S=0
0470 DEF FND(D)=SQR((K[I, 1]-S1)^2+(K[I, 2]-S2)^2)
0480 REM INITIALIZE ENTERPRISE'S POSITION
0490 LET Q1=INT(RND(1)*8+1)
0500 LET Q2=INT(RND(1)*8+1)
0510 LET S1=INT(RND(1)*8+1)
0520 LET S2=INT(RND(1)*8+1)
0530 MAT C=ZER
0540 LET C[3, 1]=-1
0550 LET C[2, 1]=-1
0560 LET C[4, 1]=-1
0570 LET C[4, 2]=-1
0580 LET C[5, 2]=-1
0590 LET C[6, 2]=-1
0600 LET C[1, 2]=1
0610 LET C[2, 2]=1
0620 LET C[6, 1]=1
0630 LET C[7, 1]=1
0640 LET C[8, 1]=1
0650 LET C[8, 2]=1
0660 LET C[9, 2]=1
0670 DIM D[8]
0680 FOR I=1 TO 8
0690   LET D[I]=0
0700 NEXT I
0710 LET A1$="NSLPTSDCX"
0720 LET D$="WARP ENGINESS. R.  SENSORSL. R.  SENSORSPHASER CNTRL"
0730 LET D$=D$, "PHOTON TUBESDAMAGE CNTRLSHIELD CNTRLCOMPUTER"
0740 LET G4$="III"
0750 LET G1$="ANTARES. SIRIUS. RIGEL. DENEB. PROCYON. CAPELLA. VEGA. "
0760 LET G1$=G1$, "BETELGEUSE. CANOPUS. ALDEBARAN. ALTAIR. REGULUS. "
0770 LET G1$=G1$, "SAGITTARIUS. ARCTURUS. POLLUX. SPICA. "
0780 LET B9=0
0790 LET K9=0
0800 LET A1$="NSLPTSDCX"
0810 REM  SET UP WHAT EXISTS IN GALAXY ...
0820 FOR I=1 TO 8
0830   FOR J=1 TO 8
0840     LET R1=RND(1)
0850     IF R1>. 98 THEN GOTO 0900
0860     IF R1>. 95 THEN GOTO 0930
0870     IF R1>. 8 THEN GOTO 0960
0880     LET K3=0
0890     GOTO 0980
0900     LET K3=3
0910     LET K9=K9+3
0920     GOTO 0980
0930     LET K3=2
0940     LET K9=K9+2
0950     GOTO 0980
0960     LET K3=1
0970     LET K9=K9+1
0980     LET R1=RND(1)
0990     IF R1>. 96 THEN GOTO 1020
1000     LET B3=0
1010     GOTO 1040
1020     LET B3=1
1030     LET B9=B9+1
1040     LET S3=INT(RND(1)*8+1)
1050     LET G[I, J]=K3*100+B3*10+S3
1060     REM K3=#KLINGONS B3=#STARBASES S3=#STARS
1070     LET Z[I, J]=0
1080   NEXT J
1090 NEXT I
1100 LET K7=K9
1110 DIM X$[2], X0$[5]
1120 LET X$=""
1130 LET X0$=" IS "
1140 IF B9<>0 THEN GOTO 1200
1150 LET B9=1
1160 IF G[6, 3]>=200 THEN GOTO 1190
1170 LET G[6, 3]=G[6, 3]+100
1180 LET K9=K9+1
1190 LET G[6, 3]=G[6, 3]+10
1200 IF B9=1 THEN GOTO 1230
1210 LET X$="S"
1220 LET X0$=" ARE "
1230 PRINT "YOUR ORDERS ARE AS FOLLOWS:"
1240 PRINT "     DESTROY THE"K9" KLINGON WARSHIPS WHICH HAVE INVADED"
1250 PRINT "  THE GALAXY BEFORE THEY CAN ATTACK FEDERATION HEADQUARTERS"
1260 PRINT "  ON STARDATE"T0+T9"; THIS GIVES YOU"T9" DAYS.   THERE"X0$
1270 PRINT " "B9" STARBASE"; X$" IN THE GALAXY FOR RESUPPLYING YOUR SHIP. "
1280 PRINT
1290 PRINT "HIT 'RETURN' WHEN READY TO ASSUME COMMAND ---"
```

```
1300 INPUT A$
1310 REM *** HERE ANY TIME ENTER NEW QUADRANT ...
1320 LET Z4=Q1
1330 LET Z5=Q2
1340 LET K3=0
1350 LET B3=0
1360 LET S3=0
1370 LET G5=0
1380 LET D4=.5*RND(1)
1390 IF Q1<1 THEN GOTO 1600
1400 IF Q1>8 THEN GOTO 1600
1410 IF Q2<1 THEN GOTO 1600
1420 IF Q2>8 THEN GOTO 1600
1430 GOSUB 9030
1440 PRINT
1450 IF T<>T0 THEN GOTO 1490
1460 PRINT "YOUR MISSION BEGINS WITH YOUR STARSHIP LOCATED"
1470 PRINT "IN THE GALACTIC QUADRANT, '"G2$"'."
1480 GOTO 1500
1490 PRINT "NOW ENTERING "G2$" QUADRANT ..."
1500 PRINT
1510 LET X=G[Q1,Q2]*.01
1520 LET K3=INT(X)
1530 LET B3=INT((X-K3)*10)
1540 LET S3=G[Q1,Q2]-INT(G[Q1,Q2]*.1)*10
1550 IF K3=0 THEN GOTO 1590
1560 PRINT "COMBAT AREA      CONDITION RED"
1570 IF S>200 THEN GOTO 1590
1580 PRINT "   SHIELDS DANGEROUSLY LOW"
1590 MAT K=ZER
1600 FOR I=1 TO 3
1610   LET K[I,3]=0
1620 NEXT I
1630 LET Q$=Z$
1640 LET R$=Z$
1650 LET S$=Z$[1,48]
1660 REM POSITION ENTERPRISE IN QUADRANT, THEN PLACE 'K3' KLINGONS,
1670 REM   'B3' STARBASES, AND 'S3' STARS ELSEWHERE.
1680 LET A$="<*>"
1690 LET Z1=S1
1700 LET Z2=S2
1710 GOSUB 8670
1720 FOR I=1 TO K3
1730   GOSUB 8590
1740   LET A$="+++"
1750   LET Z1=R1
1760   LET Z2=R2
1770   GOSUB 8670
1780   LET K[I,1]=R1
1790   LET K[I,2]=R2
1800   LET K[I,3]=S9
1810 NEXT I
1820 FOR I=1 TO B3
1830   GOSUB 8590
1840   LET A$=">!<"
1850   LET Z1=R1
1860   LET Z2=R2
1870   GOSUB 8670
1880   LET B4=Z1
1890   LET B5=Z2
1900 NEXT I
1910 FOR I=1 TO S3
1920   GOSUB 8590
1930   LET A$=" * "
1940   LET Z1=R1
1950   LET Z2=R2
1960   GOSUB 8670
1970 NEXT I
1980 GOSUB 6430
1990 IF S+E<=10 THEN GOTO 2020
2000 IF E>10 THEN GOTO 2060
2010 IF D[7]>=0 THEN GOTO 2060
2020 PRINT "<7>** FATAL ERROR **<7>   YOU'VE JUST STRANDED YOUR SHIP IN SPACE!!"
2030 PRINT "YOU HAVE INSUFFICIENT MANEUVERING ENERGY, AND SHIELD CONTROL"
2040 PRINT "IS PRESENTLY INCAPABLE OF CROSS-CIRCUITING TO ENGINE ROOM!"
2050 GOTO 6260
2060 PRINT "COMMAND";
2070 INPUT A$
2080 FOR I=1 TO 9
2090   IF A$[1,1]<>A1$[I,I] THEN GOTO 2160
2100   IF I<>2 THEN GOTO 2140
2110   IF LEN(A$)<2 THEN GOTO 2140
2120   IF A$[2,2]="R" THEN GOTO 2140
2130   LET I=6
2140   ON I THEN GOTO 2300, 1980, 4000, 4260, 4700, 5530, 5690, 7290
2150   IF A$="XXX" THEN GOTO 6270
2160 NEXT I
2170 PRINT "ENTER ONE OF THE FOLLOWING:"
2180 PRINT "NAV (TO SET COURSE)"
2190 PRINT "SRS (FOR SHORT RANGE SENSOR SCAN)"
2200 PRINT "LRS (FOR LONG RANGE SENSOR SCAN)"
2210 PRINT "PHA (TO FIRE PHASERS)"
2220 PRINT "TOR (TO FIRE PHOTON TORPEDOES)"
2230 PRINT "SHE (TO RAISE OR LOWER SHIELDS)"
2240 PRINT "DAM (FOR DAMAGE CONTROL REPORT)"
2250 PRINT "COM (TO CALL ON LIBRARY-COMPUTER)"
2260 PRINT "XXX (TO RESIGN YOUR COMMAND)"
2270 PRINT
2280 GOTO 1990
2290 REM  COURSE CONTROL BEGINS HERE
2300 PRINT "COURSE (1-9)";
2310 INPUT C1
2320 IF C1>=1 THEN GOTO 2350
2330 PRINT "  LT. SULU REPORTS, 'INCORRECT COURSE DATA, SIR!'"
2340 GOTO 1990
2350 IF C1<9 THEN GOTO 2380
2360 IF C1>9 THEN GOTO 2330
2370 LET C1=1
2380 PRINT "WARP FACTOR (0-8)";
2390 INPUT W1
2400 IF W1<=0 THEN GOTO 2420
2410 IF W1<=8 THEN GOTO 2450
2420 PRINT "  CHIEF ENGINEER SCOTT REPORTS 'THE ENGINES WON'T"
2430 PRINT "           TAKE WARP "W1"!'"
2440 GOTO 1990
2450 IF D[1]>=0 THEN GOTO 2490
2460 IF W1<=.2 THEN GOTO 2490
2470 PRINT "WARP ENGINES ARE DAMAGED.  MAXIMUM SPEED = WARP 0.2"
2480 GOTO 2300
2490 LET N=INT(W1*8+.5)
2500 IF E-N>=0 THEN GOTO 2590
2510 PRINT "ENGINEERING REPORTS 'INSUFFICIENT ENERGY AVAILABLE"
2520 PRINT "   FOR MANEUVERING AT WARP"W1"!'"
2530 IF S<N-E THEN GOTO 1990
2540 IF D[7]<0 THEN GOTO 1990
2550 PRINT "DEFLECTOR CONTROL ROOM ACKNOWLEGES"S" UNITS"
2560 PRINT "   OF ENERGY PRESENTLY DEPLOYED TO SHIELDS."
2570 GOTO 5530
2580 REM KLINGONS MOVE/FIRE ON MOVING STARSHIP ...
2590 FOR I=1 TO K3
2600   IF K[I,3]<=0 THEN GOTO 2700
2610   LET A$="   "
2620   LET Z1=K[I,1]
2630   LET Z2=K[I,2]
2640   GOSUB 8670
2650   GOSUB 8590
2660   LET K[I,1]=Z1
2670   LET K[I,2]=Z2
2680   LET A$="+++"
2690   GOSUB 8670
2700 NEXT I
2710 GOSUB 6000
2720 LET D1=0
2730 LET D6=W1
2740 IF W1<1 THEN GOTO 2770
2750 LET D6=1
2760 REM MAKE REPAIRS TO SHIP
2770 FOR I=1 TO 8
2780   IF D[I]>=0 THEN GOTO 2880
2790   LET D[I]=D[I]+D6
2800   IF D[I]<0 THEN GOTO 2880
2810   IF D1=1 THEN GOTO 2840
2820   LET D1=1
2830   PRINT "DAMAGE CONTROL REPORT:"
2840   PRINT  TAB(8);
2850   LET R1=I
2860   GOSUB 8790
2870   PRINT " REPAIR COMPLETED"
2880 NEXT I
2890 REM DAMAGE/IMPROVEMENT DURING SOME VES
2900 IF RND(1)>.2 THEN GOTO 3070
2910 LET R1=INT(RND(1)*8+1)
2920 IF RND(1)>=.6 THEN GOTO 3000
2930 LET D[R1]=D[R1]-(RND(1)*5+1)
2940 PRINT
2950 PRINT "DAMAGE CONTROL REPORT:";
2960 GOSUB 8790
2970 PRINT " DAMAGED"
2980 PRINT
2990 GOTO 3070
3000 LET D[R1]=D[R1]+(RND(1)*3+1)
3010 PRINT
3020 PRINT "DAMAGE CONTROL REPORT:";
3030 GOSUB 8790
3040 PRINT " STATE OF REPAIR IMPROVED"
3050 PRINT
3060 REM BEGIN MOVING STARSHIP **
3070 LET A$="   "
3080 LET Z1=INT(S1)
3090 LET Z2=INT(S2)
3100 GOSUB 8670
3110 LET X1=C[C1,1]+(C[C1+1,1]-C[C1,1])*(C1-INT(C1))
3120 LET X=S1
3130 LET Y=S2
3140 LET X2=C[C1,2]+(C[C1+1,2]-C[C1,2])*(C1-INT(C1))
3150 LET Q4=Q1
3160 LET Q5=Q2
3170 FOR I=1 TO N
3180   LET S1=S1+X1
3190   LET S2=S2+X2
3200   IF S1<1 THEN GOTO 3500
3210   IF S1>=9 THEN GOTO 3500
3220   IF S2<1 THEN GOTO 3500
3230   IF S2>=9 THEN GOTO 3500
3240   LET S8=INT(S1)*24+INT(S2)*3-26
3250   IF S8>72 THEN GOTO 3280
3260   IF Q$[S8,S8+2]="   " THEN GOTO 3360
3270   GOTO 3320
3280   IF S8>144 THEN GOTO 3310
3290   IF R$[S8-72,S8-70]="   " THEN GOTO 3360
3300   GOTO 3320
3310   IF S$[S8-144,S8-142]="   " THEN GOTO 3360
3320   LET S1=S1-X1
3330   LET S2=S2-X2
3340   PRINT "WARP ENGINES SHUT DOWN AT SECTOR "S1","S2" DUE TO BAD NAVIGATION."
3350   GOTO 3370
3360 NEXT I
3370 LET A$="<*>"
3380 LET Z1=INT(S1)
3390 LET Z2=INT(S2)
3400 GOSUB 8670
3410 GOSUB 3910
3420 LET T8=1
3430 IF W1>=1 THEN GOTO 3450
3440 LET T8=.1*INT(10*W1)
3450 LET T=T+T8
3460 IF T>T0+T9 THEN GOTO 6220
3470 REM SEE IF DOCKED, THEN GET COMMAND
3480 GOTO 1980
3490 REM EXCEEDED QUADRANT LIMITS
3500 LET X=8*Q1+X+N*X1
3510 LET Y=8*Q2+Y+N*X2
3520 LET Q1=INT(X/8)
3530 LET Q2=INT(Y/8)
3540 LET S1=INT(X-Q1*8)
3550 LET S2=INT(Y-Q2*8)
3560 IF S1<>0 THEN GOTO 3590
3570 LET Q1=Q1-1
3580 LET S1=8
3590 IF S2<>0 THEN GOTO 3620
3600 LET Q2=Q2-1
3610 LET S2=8
3620 LET X5=0
3630 IF Q1>=1 THEN GOTO 3670
3640 LET X5=1
3650 LET Q1=1
3660 LET S1=1
3670 IF Q1<=8 THEN GOTO 3710
3680 LET X5=1
3690 LET Q1=8
3700 LET S1=8
3710 IF Q2>=1 THEN GOTO 3750
3720 LET X5=1
3730 LET Q2=1
3740 LET S2=1
3750 IF Q2<=8 THEN GOTO 3790
3760 LET X5=1
3770 LET Q2=8
3780 LET S2=8
3790 IF X5=0 THEN GOTO 3860
3800 PRINT "LT. UHURA REPORTS MESSAGE FROM STARFLEET COMMAND:"
3810 PRINT "  'PERMISSION TO ATTEMPT CROSSING OF GALACTIC PERIMETER"
3820 PRINT "   IS HEREBY *DENIED*.  SHUT DOWN YOUR ENGINES.'"
3830 PRINT "CHIEF ENGINEER SCOTT REPORTS 'WARP ENGINES SHUT DOWN"
3840 PRINT "        AT SECTOR "S1","S2" OF QUADRANT "Q1","Q2".'"
3850 IF T>T0+T9 THEN GOTO 6220
3860 IF 8*Q1+Q2=8*Q4+Q5 THEN GOTO 3370
3870 LET T=T+1
3880 GOSUB 3910
3890 GOTO 1320
3900 REM MANEUVER ENERGY S/R ***
3910 LET E=E-N-10
3920 IF E>=0 THEN GOTO 3980
3930 PRINT "SHIELD CONTROL SUPPLIED ENERGY TO COMPLETE THE MANEUVER."
```

```
3940 LET S=S+E
3950 LET E=0
3960 IF S>0 THEN GOTO 3980
3970 LET S=0
3980 RETURN
3990 REM L. R.  SENSOR SCAN CODE ***
4000 IF D[3]>=0 THEN GOTO 4030
4010 PRINT "LONG RANGE SENSORS ARE INOPERABLE"
4020 GOTO 1990
4030 PRINT "LONG RANGE SENSOR SCAN FOR QUADRANT"Q1","Q2
4040 LET O1$=". . . . . . . . . . . . . . . . . . . "
4050 PRINT O1$
4060 FOR I=Q1-1 TO Q1+1
4070   DIM N[3]
4080   FOR I1=1 TO 3
4090     LET N[I1]=0
4100   NEXT I1
4110   FOR J=Q2-1 TO Q2+1
4120     IF I<1 THEN GOTO 4180
4130     IF I>8 THEN GOTO 4180
4140     IF J<1 THEN GOTO 4180
4150     IF J>8 THEN GOTO 4180
4160     LET N[J-Q2+2]=G[I,J]
4170     LET Z[I,J]=G[I,J]
4180   NEXT J
4190   DIM P1$[20]
4200   LET P1$=": ### : ### : ### :"
4210   PRINT USING P1$,N[1],N[2],N[3]
4220   PRINT O1$
4230 NEXT I
4240 GOTO 1990
4250 REM *** PHASER CONTROL CODE BEGINS HERE
4260 IF K3>0 THEN GOTO 4300
4270 PRINT "SCIENCE OFFICER SPOCK REPORTS 'SENSORS SHOW"
4280 PRINT "  NO ENEMY SHIPS IN THIS QUADRANT.'"
4290 GOTO 1990
4300 IF D[4]>=0 THEN GOTO 4330
4310 PRINT "PHASERS INOPERATIVE"
4320 GOTO 1990
4330 IF D[8]>=0 THEN GOTO 4350
4340 PRINT "COMPUTER FAILURE HAMPERS ACCURACY"
4350 PRINT "PHASERS LOCKED ON TARGET; "
4360 PRINT "ENERGY AVAILABLE ="E
4370 PRINT "NUMBER OF UNITS TO FIRE:";
4380 INPUT X
4390 IF X<=0 THEN GOTO 1990
4400 IF E-X<0 THEN GOTO 4360
4410 LET E=E-X
4420 GOSUB 6000
4430 IF D[7]>=0 THEN GOTO 4450
4440 LET X=X*RND(1)
4450 LET H1=INT(X/K3)
4460 FOR I=1 TO 3
4470   IF K[I,3]<=0 THEN GOTO 4670
4480   LET H=INT((H1/FND(0))*(RND(1)+2))
4490   IF H>.15*K[I,3] THEN GOTO 4530
4500   PRINT "SENSORS SHOW NO DAMAGE"
4510   PRINT "    TO ENEMY AT "K[I,1]","K[I,2]
4520   GOTO 4670
4530   LET K[I,3]=K[I,3]-H
4540   PRINT H" UNIT HIT ON KLINGON AT SECTOR "K[I,1]","K[I,2]
4550   IF K[I,3]<=0 THEN GOTO 4580
4560   PRINT "   (SENSORS SHOW"K[I,3]" UNITS REMAINING)"
4570   GOTO 4670
4580   PRINT " *** KLINGON DESTROYED ***"
4590   LET K3=K3-1
4600   LET K9=K9-1
4610   LET A$="   "
4620   LET Z1=K[I,1]
4630   LET Z2=K[I,2]
4640   GOSUB 8670
4650   LET G[Q1,Q2]=K3*100+B3*10+S3
4660   IF K9<=0 THEN GOTO 6370
4670 NEXT I
4680 GOTO 1990
4690 REM PHOTON TORPEDO CODE BEGINS ***
4700 IF D[5]>=0 THEN GOTO 4730
4710 PRINT "PHOTON TUBES ARE NOT OPERATIONAL "
4720 GOTO 1990
4730 IF P>0 THEN GOTO 4760
4740 PRINT "ALL PHOTON TORPEDOES EXPENDED"
4750 GOTO 1990
4760 PRINT "TORPEDO COURSE (1-9)";
4770 INPUT C1
4780 IF C1>=1 THEN GOTO 4810
4790 PRINT " ENSIGN CHEKOV REPORTS, 'INCORRECT COURSE DATA, SIR!'"
4800 GOTO 1990
4810 IF C1>9 THEN GOTO 4790
4820 IF C1<9 THEN GOTO 4850
4830 IF C1>=9 THEN GOTO 4760
4840 LET C1=1
4850 LET X1=C[C1,1]+(C[C1+1,1]-C[C1,1])*(C1-INT(C1))
4860 LET X2=C[C1,2]+(C[C1+1,2]-C[C1,2])*(C1-INT(C1))
4870 LET E=E-2
4880 LET X=S1
4890 LET Y=S2
4900 LET P=P-1
4910 PRINT "TORPEDO TRACK:"
4920 LET X=X+X1
4930 LET Y=Y+X2
4940 LET X3=INT(X+.5)
4950 LET Y3=INT(Y+.5)
4960 IF X3<1 THEN GOTO 5490
4970 IF X3>=9 THEN GOTO 5490
4980 IF Y3<1 THEN GOTO 5490
4990 IF Y3>=9 THEN GOTO 5490
5000 PRINT "                  "X3","Y3
5010 LET A$="   "
5020 LET Z1=X
5030 LET Z2=Y
5040 GOSUB 8830
5050 IF Z3<>0 THEN GOTO 4920
5060 LET A$="+++"
5070 LET Z1=X
5080 LET Z2=Y
5090 GOSUB 8830
5100 IF Z3=0 THEN GOTO 5210
5110 PRINT "*** KLINGON DESTROYED ***"
5120 LET K3=K3-1
5130 LET K9=K9-1
5140 IF K9<=0 THEN GOTO 6370
5150 FOR I=1 TO 3
5160   IF X3<>K[I,1] THEN GOTO 5180
5170   IF Y3=K[I,2] THEN GOTO 5190
5180 NEXT I
5190 LET K[I,3]=0
5200 GOTO 5430
5210 LET A$=" * "
5220 LET Z1=X
5230 LET Z2=Y
5240 GOSUB 8830
5250 IF Z3=0 THEN GOTO 5280
5260 PRINT "STAR AT"X3;","Y3;" ABSORBED TORPEDO ENERGY."
5270 GOTO 5500
5280 LET A$=">!<"
5290 LET Z1=X
5300 LET Z2=Y
5310 GOSUB 8830
5320 IF Z3=0 THEN GOTO 4760
5330 PRINT "*** STARBASE DESTROYED ***"
5340 LET B3=B3-1
5350 LET B9=B9-1
5360 IF B9>0 THEN GOTO 5400
5370 PRINT "THAT DOES IT, CAPTAIN!!  YOU ARE HEREBY RELIEVED OF COMMAND"
5380 PRINT "   AND SENTENCED TO 99 STARDATES AT HARD LABOR ON CYGNUS 12!!"
5390 GOTO 6270
5400 PRINT "STARFLEET COMMAND REVIEWING YOUR RECORD TO CONSIDER"
5410 PRINT "   COURT MARTIAL!"
5420 LET D0=0
5430 LET A$="   "
5440 LET Z1=X
5450 LET Z2=Y
5460 GOSUB 8670
5470 LET G[Q1,Q2]=K3*100+B3*10+S3
5480 GOTO 5500
5490 PRINT "TORPEDO MISSED"
5500 GOSUB 6000
5510 GOTO 1990
5520 REM *** SHIELD CONTROL STARTS HERE
5530 IF D[7]>=0 THEN GOTO 5560
5540 PRINT "SHIELD CONTROL INOPERABLE"
5550 GOTO 1990
5560 PRINT "ENERGY AVAILABLE ="E+S".  NUMBER OF UNITS TO SHIELDS:";
5570 INPUT X
5580 IF X>=0 THEN GOTO 5620
5590 IF S<>X THEN GOTO 5620
5600 PRINT "(SHIELDS UNCHANGED)"
5610 GOTO 1990
5620 IF E+S-X<0 THEN GOTO 5560
5630 LET E=E+S-X
5640 LET S=X
5650 PRINT "DEFLECTOR CONTROL ROOM REPORT:"
5660 PRINT "  'SHIELDS NOW AT "S" PER YOUR COMMAND'"
5670 GOTO 1990
5680 REM *** DAMAGE CONTROL STARTS HERE
5690 IF D[6]>=0 THEN GOTO 5910
5700 PRINT "DAMAGE CONTROL REPORT NOT AVAILABLE"
5710 IF D0=0 THEN GOTO 1990
5720 LET D3=0
5730 FOR I=1 TO 8
5740   IF D[I]>=0 THEN GOTO 5760
5750   LET D3=D3+.1
5760 NEXT I
5770 IF D3=0 THEN GOTO 1990
5780 LET D3=D3+D4
5790 IF D3<1 THEN GOTO 5810
5800 LET D3=.9
5810 PRINT "TECHNICIANS STANDING BY TO EFFECT REPAIRS TO YOUR SHIP;"
5820 PRINT "ESTIMATED TIME TO REPAIR: ";
5830 PRINT USING ".# STARDATES",D3
5840 PRINT "WILL YOU AUTHORIZE THE REPAIR ORDER (YES/NO)";
5850 INPUT A$
5860 IF A$<>"YES" THEN GOTO 1990
5870 FOR I=1 TO 8
5880   LET D[I]=0
5890 NEXT I
5900 LET T=T+D3+.1
5910 PRINT
5920 PRINT "DEVICE          STATE OF REPAIR"
5930 FOR R1=1 TO 8
5940   GOSUB 8790
5950   PRINT USING "  -##.##",D[R1]
5960 NEXT R1
5970 PRINT
5980 GOTO 5710
5990 REM "KLINGONS SHOOTING" CODE BEGINS ***
6000 IF K3<=0 THEN GOTO 6210
6010 IF D0=0 THEN GOTO 6040
6020 PRINT "STAR BASE SHIELDS PROTECT THE ENTERPRISE"
6030 GOTO 6210
6040 FOR I=1 TO 3
6050   IF K[I,3]<=0 THEN GOTO 6200
6060   LET H=INT((K[I,3]/FND(0))*(2+RND(1)))
6070   LET S=S-H
6080   PRINT H" UNIT HIT ON ENTERPRISE FROM SECTOR"K[I,1]","K[I,2]
6090   IF S<0 THEN GOTO 6240
6100   PRINT "        (SHIELDS DOWN TO"S" UNITS.)"
6110   IF H<20 THEN GOTO 6200
6120   IF RND(1)>.6 THEN GOTO 6200
6130   IF H/S<=.02 THEN GOTO 6200
6140   LET D2=H/S+.5*RND(1)
6150   LET R1=INT(RND(1)*8+1)
6160   LET D[R1]=D[R1]-D2
6170   PRINT "DAMAGE CONTROL REPORTS '";
6180   GOSUB 8790
6190   PRINT "DAMAGED BY THE HIT!'"
6200 NEXT I
6210 RETURN
6220 PRINT "IT IS STARDATE "T
6230 GOTO 6270
6240 PRINT
6250 PRINT "THE ENTERPRISE HAS BEEN DESTROYED.  THE FEDERATION WILL BE CONQUERED."
6260 PRINT "IT IS STARDATE "T"."
6270 PRINT "THERE WERE "K9" KLINGON BATTLE CRUISERS LEFT AT"
6280 PRINT " THE END OF YOUR MISSION."
6290 PRINT
6300 PRINT
6310 PRINT "THE FEDERATION IS IN NEED OF A NEW STARSHIP COMMANDER"
6320 PRINT "FOR A SIMILAR MISSION -- IF THERE IS A VOLUNTEER,"
6330 PRINT "LET HIM STEP FORWARD AND ENTER 'AYE'."
6340 INPUT A$
6350 IF A$="AYE" THEN GOTO 0240
6360 GOTO 9250
6370 PRINT "CONGRATULATIONS, CAPTAIN! THE LAST KLINGON BATTLE CRUISER"
6380 PRINT "  MENACING THE FEDERATION HAS BEEN DESTROYED."
6390 PRINT
6400 PRINT "YOUR EFFICIENCY RATING IS "((K7/(T-T0))*1000)"."
6410 GOTO 6290
6420 REM S. R.  SENSOR SCAN & STARTUP SUBR. ***
6430 FOR I=S1-1 TO S1+1
6440   FOR J=S2-1 TO S2+1
6450     IF INT(I+.5)<1 THEN GOTO 6540
6460     IF INT(I+.5)>8 THEN GOTO 6540
6470     IF INT(J+.5)<1 THEN GOTO 6540
6480     IF INT(J+.5)>8 THEN GOTO 6540
6490     LET A$=">!<"
6500     LET Z1=I
6510     LET Z2=J
6520     GOSUB 8830
6530     IF Z3=1 THEN GOTO 6580
6540   NEXT J
6550 NEXT I
6560 LET D0=0
6570 GOTO 6650
```

```
6580 LET D0=1
6590 LET C$="DOCKED"
6600 LET E=3000
6610 LET P=10
6620 PRINT "SHIELDS DROPPED FOR DOCKING PURPOSES"
6630 LET S=0
6640 GOTO 6720
6650 IF K3>0 THEN GOTO 6690
6660 IF E<E0*.1 THEN GOTO 6710
6670 LET C$=" GREEN"
6680 GOTO 6720
6690 LET C$=" <7>*RED*<7>"
6700 GOTO 6720
6710 LET C$="YELLOW"
6720 IF D[2]>=0 THEN GOTO 6770
6730 PRINT
6740 PRINT "*** SHORT RANGE SENSORS ARE OUT ***"
6750 PRINT
6760 GOTO 7270
6770 LET O1$="---------------------------------"
6780 PRINT O1$
6790 DIM N5$[4]
6800 LET N5$="####"
6810 PRINT " ";
6820 FOR I=1 TO 22 STEP 3
6830   PRINT Q$[I,I+2]" ";
6840 NEXT I
6850 PRINT
6860 PRINT " ";
6870 FOR I=25 TO 46 STEP 3
6880   PRINT Q$[I,I+2]" ";
6890 NEXT I
6900 PRINT "          STARDATE        ";
6910 PRINT USING "####.#",T
6920 PRINT " ";
6930 FOR I=49 TO 70 STEP 3
6940   PRINT Q$[I,I+2]" ";
6950 NEXT I
6960 PRINT "          CONDITION       ";
6970 PRINT C$
6980 PRINT " ";
6990 FOR I=1 TO 22 STEP 3
7000   PRINT R$[I,I+2]" ";
7010 NEXT I
7020 PRINT "          QUADRANT          "Q1","Q2
7030 PRINT " ";
7040 FOR I=25 TO 46 STEP 3
7050   PRINT R$[I,I+2]" ";
7060 NEXT I
7070 PRINT "          SECTOR            "S1","S2
7080 PRINT " ";
7090 FOR I=49 TO 70 STEP 3
7100   PRINT R$[I,I+2]" ";
7110 NEXT I
7120 PRINT "          TOTAL ENERGY      ";
7130 PRINT USING N5$,E+S
7140 PRINT " ";
7150 FOR I=1 TO 22 STEP 3
7160   PRINT S$[I,I+2]" ";
7170 NEXT I
7180 PRINT "          PHOTON TORPEDOES ";
7190 PRINT USING N5$,P
7200 PRINT " ";
7210 FOR I=25 TO 46 STEP 3
7220   PRINT S$[I,I+2]" ";
7230 NEXT I
7240 PRINT "          SHIELDS           ";
7250 PRINT USING N5$,S
7260 PRINT O1$
7270 RETURN
7280 REM *** LIBRARY COMPUTER CODE BEGINS HERE
7290 IF D[8]>=0 THEN GOTO 7320
7300 PRINT "COMPUTER DISABLED"
7310 GOTO 1990
7320 PRINT "COMPUTER ACTIVE AND AWAITING COMMAND:";
7330 INPUT A
7340 IF A<0 THEN GOTO 1990
7350 PRINT
7360 LET H8=1
7370 IF A=0 THEN GOTO 7540
7380 ON A THEN GOTO 7900, 8070, 8500, 8150, 7400
7390 GOTO 7450
7400 REM *** SETUP TO CHANGE C.G.RECORD TO GALAXY MAP
7410 LET H8=0
7420 LET G5=1
7430 PRINT "                          THE GALAXY"
7440 GOTO 7550
7450 PRINT "FUNCTIONS AVAILABLE FROM LIBRARY-COMPUTER:"
7460 PRINT "   0 = CUMULATIVE GALACTIC RECORD"
7470 PRINT "   1 = STATUS REPORT"
7480 PRINT "   2 = PHOTON TORPEDO DATA"
7490 PRINT "   3 = STARBASE NAV DATA"
7500 PRINT "   4 = DIRECTION/DISTANCE CALCULATOR"
7510 PRINT "   5 = GALAXY 'REGION NAME' MAP"
7520 GOTO 7320
7530 REM *** CUM GALACTIC RECORD CODE BEGINS ***
7540 PRINT "COMPUTER RECORD OF GALAXY FOR QUADRANT "Q1","Q2":"
7550 PRINT "      1     2     3     4     5     6     7     8"
7560 LET O3$="   ----- ----- ----- ----- ----- ----- ----- -----"
7570 PRINT O3$
7580 DIM N1$[2],N2$[8],N$[5]
7590 FOR I=1 TO 8
7600   LET N1$="#"
7610   PRINT USING N1$,I;
7620   IF H8=0 THEN GOTO 7740
7630   FOR J=1 TO 8
7640     LET N2$="   ###"
7650     LET N$=""
7660     IF I<>Q1 THEN GOTO 7700
7670     IF J<>Q2 THEN GOTO 7700
7680     LET N$="<7>"
7690     PRINT N$;
7700     PRINT USING N2$,Z[I,J];
7710     PRINT N$;
7720   NEXT J
7730   GOTO 7850
7740   LET Z4=I
7750   LET Z5=1
7760   GOSUB 9030
7770   LET J0=INT(15-.5*LEN(G2$))
7780   PRINT  TAB(J0);
7790   PRINT G2$;
7800   LET Z5=5
7810   GOSUB 9030
7820   LET J0=INT(39-.5*LEN(G2$))
7830   PRINT  TAB(J0);
7840   PRINT G2$;
7850   PRINT
7860   PRINT O3$
7870 NEXT I
7880 GOTO 1990
7890 REM *** STATUS REPORT CODE BEGINS HERE ***
7900 PRINT "   STATUS REPORT:"
7910 LET X$=""
7920 IF K9=1 THEN GOTO 7940
7930 LET X$="S"
7940 PRINT K9" KLINGON"X$" LEFT"
7950 LET V5=(T0+T9)-T
7960 PRINT USING "MISSION MUST BE COMPLETED IN ##.# STARDATES",V5
7970 LET X$=""
7980 IF B9=1 THEN GOTO 8040
7990 LET X$="S"
8000 IF B9<>0 THEN GOTO 8040
8010 PRINT "YOUR STUPIDITY HAS LEFT YOU ON YOUR OWN IN"
8020 PRINT "  THE GALAXY -- YOU HAVE NO STARBASES LEFT!"
8030 GOTO 5690
8040 PRINT "THE FEDERATION IS MAINTAINING"B9" STARBASE"X$" IN THE GALAXY"
8050 GOTO 5690
8060 REM CODE FOR TORPEDO DATA, BASE NAV, D/D CALCULATOR ***
8070 LET H8=0
8080 FOR I=1 TO 3
8090   IF K[I,3]<=0 THEN GOTO 8480
8100   LET W1=K[I,1]
8110   LET X=K[I,2]
8120   LET C1=S1
8130   LET A=S2
8140   GOTO 8220
8150   PRINT "DIRECTION/DISTANCE CALCULATOR:"
8160   PRINT "YOU ARE AT QUADRANT ("Q1","Q2") SECTOR ("S1","S2")"
8170   PRINT "PLEASE ENTER --"
8180   PRINT "  INITIAL COORDINATES (X,Y)";
8190   INPUT C1,A
8200   PRINT "  FINAL COORDINATES (X,Y)";
8210   INPUT W1,X
8220   LET X=X-A
8230   LET A=C1-W1
8240   IF X<0 THEN GOTO 8350
8250   IF A<0 THEN GOTO 8410
8260   IF X>0 THEN GOTO 8280
8270   IF A=0 THEN GOTO 8370
8280   LET C1=1
8290   IF ABS(A)<=ABS(X) THEN GOTO 8330
8300   LET V5=C1+(((ABS(A)-ABS(X))+ABS(A))/ABS(A))
8310   PRINT "DIRECTION ="V5
8320   GOTO 8460
8330   PRINT "DIRECTION ="C1+(ABS(A)/ABS(X))
8340   GOTO 8460
8350   IF A>0 THEN GOTO 8390
8360   IF X=0 THEN GOTO 8410
8370   LET C1=5
8380   GOTO 8290
8390   LET C1=3
8400   GOTO 8420
8410   LET C1=7
8420   IF ABS(A)>=ABS(X) THEN GOTO 8450
8430   PRINT "DIRECTION ="C1+(((ABS(X)-ABS(A))+ABS(X))/ABS(X))
8440   GOTO 8460
8450   PRINT "DIRECTION ="C1+(ABS(X)/ABS(A))
8460   PRINT "DISTANCE ="SQR(X^2+A^2)
8470   IF H8=1 THEN GOTO 1990
8480 NEXT I
8490 GOTO 1990
8500 IF B3<>0 THEN GOTO 8530
8510 PRINT "MR. SPOCK REPORTS, 'SENSORS SHOW NO STARBASES IN THIS QUADRANT.'"
8520 GOTO 1990
8530 PRINT "FROM ENTERPRISE TO STARBASE:"
8540 LET W1=B4
8550 LET X=B5
8560 GOTO 8120
8570 REM *** END OF LIBRARY-COMPUTER CODE
8580 REM S/R FINDS RANDOM HOLE IN QUADRANT
8590 LET R1=INT(RND(1)*8+1)
8600 LET R2=INT(RND(1)*8+1)
8610 LET A$="   "
8620 LET Z1=R1
8630 LET Z2=R2
8640 GOSUB 8830
8650 IF Z3=0 THEN GOTO 8590
8660 RETURN
8670 REM *** INSERTION IN STRING ARRAY FOR QUADRANT ***
8680 LET S8=INT(Z1+.5)*24+INT(Z2+.5)*3-26
8690 IF S8>72 THEN GOTO 8720
8700 LET Q$[S8,S8+2]=A$
8710 GOTO 8780
8720 IF S8>144 THEN GOTO 8760
8730 LET S8=S8-72
8740 LET R$[S8,S8+2]=A$
8750 GOTO 8780
8760 LET S8=S8-144
8770 LET S$[S8,S8+2]=A$
8780 RETURN
8790 REM *** PRINTS DEVICE NAME FROM ARRAY ***
8800 LET S8=R1*12-11
8810 PRINT D$[S8,S8+11],
8820 RETURN
8830 REM *** STRING COMPARISON IN QUADRANT ARRAY ***
8840 LET Z1=INT(Z1+.5)
8850 LET Z2=INT(Z2+.5)
8860 LET S8=Z1*24+Z2*3-26
8870 LET Z3=0
8880 IF S8>72 THEN GOTO 8920
8890 IF Q$[S8,S8+2]<>A$ THEN GOTO 9000
8900 LET Z3=1
8910 GOTO 9000
8920 IF S8>144 THEN GOTO 8970
8930 LET S8=S8-72
8940 IF R$[S8,S8+2]<>A$ THEN GOTO 9000
8950 LET Z3=1
8960 GOTO 9000
8970 LET S8=S8-144
8980 IF S$[S8,S8+2]<>A$ THEN GOTO 9000
8990 LET Z3=1
9000 RETURN
9010 REM ** S/R PRODUCES QUADRANT NAME IN G2$ FROM Z4,Z5(=Q1,Q2)
9020 REM ** (CALL WITH G5=1 TO GET REGION NAME ONLY)
9030 LET L2=2
9040 IF Z5>=5 THEN GOTO 9060
9050 LET L2=1
9060 LET L3=2*(Z4-1)+L2
9070 LET I3=1
9080 LET I0=1
9090 FOR L=1 TO LEN(G1$)
9100   IF G1$[L,L]<>"." THEN GOTO 9140
9110   IF I3=L3 THEN GOTO 9150
9120   LET I0=L+1
9130   LET I3=I3+1
9140 NEXT L
9150 LET G2$=G1$[I0,L-1]
9160 IF G5=1 THEN GOTO 9240
9170 LET L3=Z5
9180 IF Z5<=4 THEN GOTO 9200
9190 LET L3=Z5-4
9200 LET G3$="IV"
9210 IF L3=4 THEN GOTO 9230
9220 LET G3$=G4$[1,L3]
9230 LET G2$=G2$," ",G3$
9240 RETURN
9250 STOP
9260 END
```

```
* RUN
                 * * * STAR TREK * * *

DO YOU NEED INSTRUCTIONS (YES/NO) ? NO
YOUR ORDERS ARE AS FOLLOWS:
    DESTROY THE 16 KLINGON WARSHIPS WHICH HAVE INVADED
  THE GALAXY BEFORE THEY CAN ATTACK FEDERATION HEADQUARTERS
  ON STARDATE 2830; THIS GIVES YOU 30 DAYS.  THERE ARE
  3 STARBASES IN THE GALAXY FOR RESUPPLYING YOUR SHIP.

HIT 'RETURN' WHEN READY TO ASSUME COMMAND ---
 ?

YOUR MISSION BEGINS WITH YOUR STARSHIP LOCATED
IN THE GALACTIC QUADRANT, 'BETELGEUSE II'.

---------------------------------

          *               *                STARDATE          2800.0
   *               .  <*>     *            CONDITION          GREEN
                              *            QUADRANT            4, 6
                          *                SECTOR              3, 6
                                           TOTAL ENERGY        3000
                                           PHOTON TORPEDOES      10
                                           SHIELDS                0

---------------------------------
COMMAND ? LRS
LONG RANGE SENSOR SCAN FOR QUADRANT 4, 6
...................
:   7 :   1 :   8 :
...................
:   7 :   6 :   7 :
...................
:   6 :   8 :   8 :
...................
COMMAND ? NAV
COURSE (1-9) ? 7.7
WARP FACTOR (0-8) ? 3

NOW ENTERING ARCTURUS IV QUADRANT ...

---------------------------------
               *       *       *
                                           STARDATE          2801.0
                         <*>  *            CONDITION          GREEN
                      *                    QUADRANT            7, 8
                      *                    SECTOR              3, 6
                                           TOTAL ENERGY        2966
                                           PHOTON TORPEDOES      10
                      *           *        SHIELDS                0
---------------------------------
COMMAND ? LRS
LONG RANGE SENSOR SCAN FOR QUADRANT 7, 8
...................
:   4 :   2 :   0 :
...................
:   3 :  .8 :   0 :
...................
:   4 : 106 :   0 :
...................
COMMAND ? SHE
ENERGY AVAILABLE = 2966. NUMBER OF UNITS TO SHIELDS: ? 1000
DEFLECTOR CONTROL ROOM REPORT:
  'SHIELDS NOW AT  1000 PER YOUR COMMAND'
COMMAND ? NAV
COURSE (1-9) ? 7
WARP FACTOR (0-8) ? 1.2

NOW ENTERING SPICA IV QUADRANT ...

COMBAT AREA      CONDITION RED
---------------------------------
         +++
          *                                STARDATE          2802.0
                                           CONDITION          *RED*
                                           QUADRANT            8, 8
                         <*>               SECTOR              5, 6
  *                                        TOTAL ENERGY        2946
  *                             *          PHOTON TORPEDOES      10
                          *         *      SHIELDS             1000
---------------------------------
COMMAND ? COM
COMPUTER ACTIVE AND AWAITING COMMAND: ? 2

DIRECTION = 3.75
DISTANCE = 5
COMMAND ? TOR
TORPEDO COURSE (1-9) ? 3.75
TORPEDO TRACK:
                4, 5
                3, 5
                2, 4
                1, 3
*** KLINGON DESTROYED ***
COMMAND ? LRS
LONG RANGE SENSOR SCAN FOR QUADRANT 8, 8
...................
:   3 :   8 :   0 :
...................
:   4 :   6 :   0 :
...................
:   0 :   0 :   0 :
...................
COMMAND ? COM
COMPUTER ACTIVE AND AWAITING COMMAND: ? 0

COMPUTER RECORD OF GALAXY FOR QUADRANT  8, 8:
      1     2     3     4     5     6     7     8
    ----- ----- ----- ----- ----- ----- ----- -----
1     0     0     0     0     0     0     0     0
    ----- ----- ----- ----- ----- ----- ----- -----
2     0     0     0     0     0     0     0     0
    ----- ----- ----- ----- ----- ----- ----- -----
3     0     0     0     0     7     1     8     0
    ----- ----- ----- ----- ----- ----- ----- -----
4     0     0     0     3     7     6     7     0
    ----- ----- ----- ----- ----- ----- ----- -----
5     3     0     0     3     6     8     8     0
    ----- ----- ----- ----- ----- ----- ----- -----
6     0     3     0     0     0     0     4     2
    ----- ----- ----- ----- ----- ----- ----- -----
7     0     0     0     0     0     0     3     8
    ----- ----- ----- ----- ----- ----- ----- -----
8     0     0     0     0     0     0     4     6
    ----- ----- ----- ----- ----- ----- ----- -----
COMMAND ? NAV
COURSE (1-9) ? 5
WARP FACTOR (0-8) ? 3
```

```
COMMAND ? NAV
COURSE (1-9) ? 5
WARP FACTOR (0-8) ? .4
 105 UNIT HIT ON ENTERPRISE FROM SECTOR 1, 1
      (SHIELDS DOWN TO 1258 UNITS.)
SHIELD CONTROL SUPPLIED ENERGY TO COMPLETE THE MANEUVER.
---------------------------------
 +++
                                           STARDATE          2817.2
        <*>                                CONDITION          *RED*
                                           QUADRANT            2, 3
              *                            SECTOR              3, 3
              *                            TOTAL ENERGY        1255
                     *                     PHOTON TORPEDOES       5
                                           SHIELDS             1255
---------------------------------
COMMAND ? TOR
TORPEDO COURSE (1-9) ? 4
TORPEDO TRACK:
                2, 2
                1, 1
*** KLINGON DESTROYED ***
COMMAND ? LRS
LONG RANGE SENSOR SCAN FOR QUADRANT 2, 3
...................
:  12 :   8 :   5 :
...................
: 104 :   3 :   4 :
...................
:   7 :   5 :   6 :
...................
COMMAND ? NAV
COURSE (1-9) ? 4
WARP FACTOR (0-8) ? 1
ENGINEERING REPORTS 'INSUFFICIENT ENERGY AVAILABLE
   FOR MANEUVERING AT WARP 1!'
DEFLECTOR CONTROL ROOM ACKNOWLEGES 1255 UNITS
   OF ENERGY PRESENTLY DEPLOYED TO SHIELDS.
ENERGY AVAILABLE = 1253. NUMBER OF UNITS TO SHIELDS: ? 1100
DEFLECTOR CONTROL ROOM REPORT:
  'SHIELDS NOW AT  1100 PER YOUR COMMAND'
COMMAND ? NAV
COURSE (1-9) ? 4
WARP FACTOR (0-8) ? 1

NOW ENTERING ANTARES II QUADRANT ...
```

Space permits us to print only a small portion of the whole run; however, notes for the mission log are reproduced below.

*   *   *

CAPTAIN'S NOTES — FOR MISSION LOG:

Stardate	*Remarks*
2804.0	—Bad news. 4 Stardates out, we've only got one Klingon to our credit so far, and our Warp Engines just went bad. Fortunately there's a Starbase nearby.
2805.2	—Sensor failure en route to Starbase — computer will have to help out on docking.
2806.2	—Repaired and ready to go.
2809.2	—Bad move! Tried to power up to Warp 5 without checking sensor readout — saved by auto-shut-down, but engines conked out again! Back to Altair III for repairs . . .
2815.5	—More bad luck — computer failure while moving into battle area, so I eyeballed the first torpedo shot. Missed! They fired back and further damaged the computer. Second shot put one of the three out, but as I maneuvered, they hit the computer again! One Klingon came up to meet me — finished him off with a photon torpedo. Phasers held up long enough to get the third one.
2815.8	—Condition yellow: too much energy deployed to shields. Will leave it this way until after the battle I expect in Rigel III.
2817.2	—While moving in for a better shot (since my computer can't help me aim), shield control had to help out on maneuvering energy. Fortunately, the enemy fought back without tactical movement.

Mr. Spock's 7th Sense

by T. R. Kibler
Georgia State University

The Starship Enterprise had as its five year mission to explore strange new worlds, to seek out new civilizations. Besides Captain Kirk and Mr. Spock and its crew, the Enterprise carried an impressive array of scientific apparatus with which to carry out its exploration. Not the least of these was the Transporter which could, by separating the atoms of matter in one location and reassembling them in another, transport people or objects from place to place. It was most often used to transport survey parties to the surface of a planet for exploration. However, before a team would beam down to a planet, Mr. Spock, as chief science officer, would use the Enterprise's sensors to scan the planet's surface to give a detailed analysis of the planet. From the Enterprise's orbit the sensors could measure the atmosphere and surface nature of a planet, including the general mineral makeup. The sensors could even detect the presence or absence of life forms — including the relative level of its civilization.

While the Transporter has to await the future and further research, the sensor is available to us today. Today sensors might not have the speed or the range of those found on the Enterprise, but they are real and most useful.

The principle of the sensor is quite simple — identify objects by the characteristic radiation they give off. However, once one gets to the details, things get a bit messy.

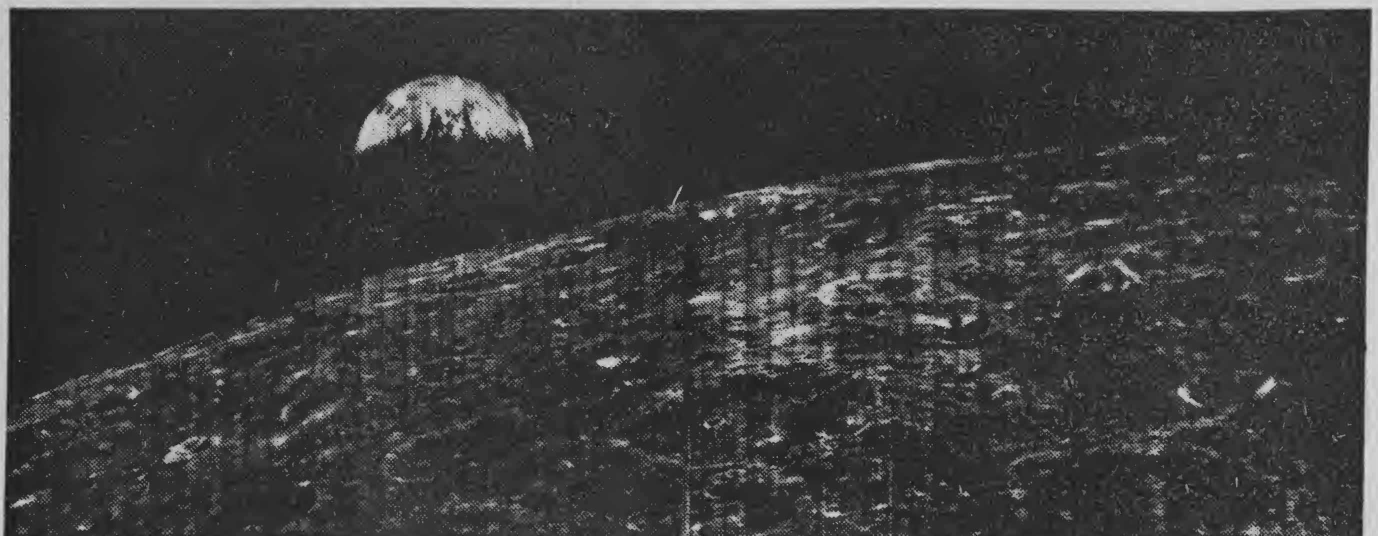
Everything on the surface of a planet or just beneath its surface or in its atmosphere (if there is one) transmits, reflects or emits radiation. Furthermore, the transmission, reflection and emission of radiation of a given object or substance forms an identifiable pattern. Just as a sample placed in a spectrophotometer has a spectral fingerprint which identifies it by the spectral absorption of its elements, so an area of the surface of a planet has a radiation pattern that identifies it and its components.

There are, however, some problems in scanning that don't occur in a spectrophotometer. In a spectrophotometer you have a well defined and controlled environment while in scanning from orbit the atmosphere filters the outside radiation and scatters the reflected and emitted radiation. Also in a spectrophotometer the angle at which light hits the sample is controlled, while in scanning the strength and characteristics of the outside radiation may vary as the zenith angle changes. Further in scanning, the surface composition, surface structures, and surface entities appear different as they are stacked one upon another. However, many of these interferences can be accounted for by analysis. That is, first the atmosphere can be analyzed as to its composition and then its effects accounted for and factored out of surface observations. Similar procedures can be done to separate particular surface objects from their immediate environment.

Again, in principle the hardware for the sensor is basically simple but the actual design becomes complex. The sensor would simply collect the radiation from some desired region of the planet and sample several wavelengths known to have discrimination qualities for surface features of interest. The patterns of readings would then be corrected for atmospheric and light source variations and then matched against standards to determine the things being observed.

In practice there are many technical problems to be considered and many tradeoffs to be weighted. For example, two problems are: what is the proper scan area that is to be observed at a single instance and what is the minimum discrimination to be made in that area. The larger the area scanned and smaller the discrimination within that area the greater the data rate from the sensor for a given scan speed. The question of area size and discrimination size is bracketed on one side by efficiency and research restraints and on the other by physical limits. The larger the scan area the faster an area of the planet can be scanned. If hourly ocean current changes are needed then the scan area must be large enough to scan the ocean area of interest in one hour. However, if the smallest element of discrimination is too small, the mass of data will overrun any system's ability to transmit or store that data — let alone analyze it. Thus for ocean analysis one would need large scan areas and large areas of discrimination. If bird counts in a sanctuary were desired, the smallest picture element must be quite small, but the scan area must also be small in order to keep within the limits of the data collection system.

Further as the size of a picture element becomes



smaller, the requirements of the optical system become more exact. Such problems as general system noise and, in particular, optical distortion become acute. There seems to be a lower limit of about 5-10 centimeters set by atmospheric scattering.

Since there is a loss of detail as the size of the picture element is increased, the obvious answer is to have two sensors systems, one with a large field of vision and large picture elements for large areas survey and locating areas of interest and a second sensor system for small discrimination within smaller areas.

To try to put all of this in perspective — without giving too much detail — let's look at a sketch of a single sensor for an earth satellite. The sensor's lenses continuously scan a strip of the planet at right angles to the motion of the satellite. The motion of the satellite determines the length, direction and speed of the scan. The strip scanned at any one instant will be broken down into picture elements. Thus a strip 185 km (100 miles) long might be broken into picture elements 15 meters long. Further, exposure time would be such that the satellite motion would be 15 meters for each exposure. Thus each exposure or reading of the sensor would sense a strip 185 km by 15 meters at a resolution of 15 meters square.

The radiation coming through the lenses of the sensor would be broken by prisms or filters into the various spectral bands required. The spectral bands being chosen to maximize the characterization of surface features of interest, while minimizing the number of such bands required. Each of the 4 or 5 such bands would have its own set of photodectors. Each set of photodectors would have 12333 elements. Each element would record one of 256 (8 bits) levels of radiation intensity. The digital output from each of the sets of dectors would be digitally compressed and encoded and then multiplexed with the signals from the other spectral bands. The digital compression reduces the volume of data by up to 60% but the encoding for error detection and correction adds back about 25% overhead to the reduced data. This multiplexed signal is then transmitted to earth receiving stations to be stored for later analysis.

The analysis consists of filtering out signal noises, enhancing the desired images and identifying what has been scanned. These are all done by digital techniques. The area of the surface that was scanned becomes a matrix with each element of the matrix being the intensity output from one photodector. Since the atmospheric absorption is different for different wavelengths, the data from each spectral band is corrected differently for the atmospheric effects to that band.

Digital enhancement continues with the elimination of as much blurring as possible and providing as much contrast as possible for objects of interest. The objects of interest vary from analysis to analysis. For example, one researcher may be interested in corn crops in a given area while another is interested in subsurface water and soil composition in the same area.

While outlines may be detectable in several of the spectral bands it is the distinctive patterns across the different spectral bands that give the actual identification of objects. For example, it could be seen from any of the matrices that there was a one acre square in the middle of a much larger area. But by considering the different spectral patterns it could be identified as either a square island in a lake or a pond in a pasture or a poppy patch in the middle of a cornfield.

This sort of analysis only tells about the physical characteristics within a small area. To reveal cultural levels and patterns, it is necessary to accumulate data into a larger picture. The amounts of artificial illumination and heat given off from an area can be checked to find a city and determine its general energy consumption. The comparison of the number of roads to the number of fields identifies an area as being primarily agricultural or industrial. The comparison of the number of forests to the number of cultivated fields helps to identify the level of agricultural development.

These techniques can be done today, although not in real time as the Enterprise's sensors could. The Earth Resources Technology Satellite of 1972 (ERTS-1) had sensors of this general nature to be used for analysis of the earth's resources. The ERTS have the advantage of being able to calibrate their sensors and analysis by scanning known areas. However, when it comes to foreign planets the problems may be more difficult — for as the crew of the Enterprise often discovered, not all planets and civilizations developed in the same manner.

SATELLITES AND COMPUTERS HELP
MANAGE EARTH'S RESOURCES

GREENBELT, MD — Operations Command and Control Center console for the ERTS-1 spacecraft at NASA's Goddard Spaceflight Center.

The ERTS program is a first step in the merger of space and remote sensing technologies into a system devoted to developing the ability for more efficient management of Earth's resources. Design of the observatory based on the highly successful Nimbus Meteorological satellites which have regularly returned pictures of the Earth weather status since 1964. The ERTS observatory will operate in a polar orbit 900 kilometers (about 560 miles) above the Earth and return images from two independently functioning multi-spectral sensors. A data collection system onboard the observatory will gather environmental information from Earth-based platforms and relay this data to the ground processing facility, at NASA's Goddard Space Flight Center, Greenbelt, Maryland. Federal agencies participating with NASA in the ERTS-1 project are the Department of Agriculture, Commerce, Interior, Defense and the Environmental Protection Agency.

THE AUTHOR

Tom Kibler, Technology Editor of Creative Computing *is Manager of Scientific Programming at the Computer Center, Georgia State University where he is also a part-time instructor in information systems. Prior to coming to GSU in 1973, Tom was a designer and researcher for IBM and prior to that a consultant and systems programmer at UC, Berkeley and Stanford Univ.*

Computers and the Weather

In 1922 an Englishman named Lewis Richardson developed a mathematical process for predicting the weather by assigning numerical values to such weather conditions as temperature, humidity, and barometric pressure, and plugging them into complex formulas. Unfortunately, the lengthy calculations made his "prediction" six days late.

In 1946, Princeton University's John von Neuman set the first computers to work on forecasting the weather, calling the project the most complex problem ever conceived. By 1955, the first computer-generated forecasts were produced on a regular basis and scientists predicted that accurate forecasts were just a step away.

The battle to predict the weather accurately is being waged by meteorologists from Bangkok to Brasilia, and the meteorologists' most powerful weapon against Mother Nature remains the computer.

Ed Olson, Control Data Corp's world weather project manager stated that "the value of all installed computer systems used for weather forecasting worldwide exceeds $150 million. Control Data holds 40% of that business, and that makes us the leader."

Olson expects the market for computers in the field of meteorology almost to double by 1980. One reason for such a tremendous market is apparent. To forecast tomorrow's weather, scientists compare and anlyze measurements of weather conditions gathered twice daily from more than 10,000 observation points around the globe. The only way to process that information, before tomorrow, is by computer.

Meteorologists use satellite photos to support computer weather forecasts in predicting the path of a hurricane so timely warnings can be given to threatened areas.

According to Olson, there are four problems in forecasting weather: "Sufficient information is not yet available, and the means to analyze, process and transmit the information are still in an embryonic stage."

The first of these problems scientists hope to solve is data collection. Weather measurements — normally wind speed, wind direction, temperature, humidity, and barometric pressure — are collected primarily from ships, ground stations, weather balloons, aircraft, and in some cases, satellites. Although adequate for landlocked areas, this information base is inadequate over the oceans, causing poor coastal region forecasts. To solve the problem, scientists plan by the end of the decade to position as many as 12 weather satellites above the seas, increasing by 10 to 15 times the present ocean weather information base. Through satellite photographs scientists will capture a much more current picture of the world's weather conditions. But curiously, solving one problem only makes other problems more complex.

"As you gather more and more weather information," Olson said, "you need more and more horsepower to process it. The first computer Control Data ever sold, a 1604 used by the Navy for weather prediction, performed 150,000 operations per second. By 1980 the significant amount of additional data available will require a machine capable of performing at hundreds of times this speed. Right now that machine doesn't exist."

The increased volume of weather information will also make data transmission problems more complex. To forecast the weather for one region of the world scientists must monitor the movement and interaction of weather conditions throughout the world. Because the weather itself knows no national boundaries, this monitoring process requires international cooperation.

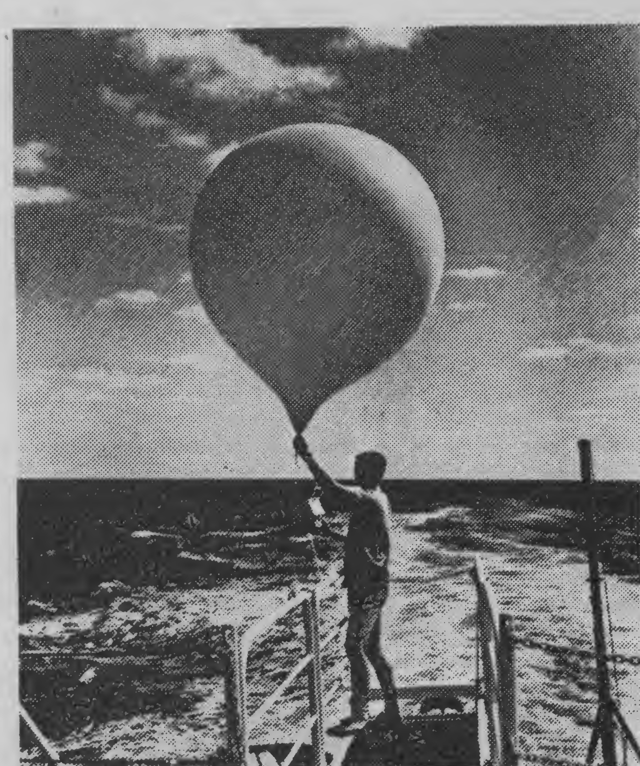
The World Meteorological Organization (WMO), has developed a global plan to solve the problem of transmitting weather information rapidly. The WMO plan consists of a network of computers that makes a world map look like a printed circuit. Linked by telephone lines and eventually by communications satellites, the WMO network will allow meteorologists to share instantly a world-wide weather information bank. "More countries participate in the WMO than belong to the United Nations," Olson said, "and the development of the network is being stimulated by WMO efforts to assist underdeveloped nations in purchasing basic systems."

Aside from the need for more powerful computers, the processing function is also far from perfect. As scientists learn more about the intricate relationships among air, earth, ocean, and sun, they must refine and change the complex formulas which are the basis for the computers' calculations.

The computer-generated forecasts must also be checked for accuracy, causing costly delays to correct insufficient or wrong information. Ed Olson recalls a time when a computer receiving weather information from a U. S. Navy destroyer fixed the ship's position in a wheat field near Salina, Kansas. It was in the Mediterranean Sea.

Meteorologist Paul Wolff, vice president of Ocean Data Systems Inc., notes that "with weather satellites providing valuable data above the oceans and with additional computer power, completely accurate one day forecasts could be available before 1980." He added, "Until we achieve completely accurate one day forecasts, long range forecasts of greater than 5 days are relatively unreliable."

With a unique spirit of international cooperation and the benefit of computer technology, scientists today can realistically say "Accurate weather forecasts are just a step away." But until that time the surest safeguard against wind, rain, and sun remains the invention of the ancient Egyptions — the umbrella.



In the early morning a meteorologist sends instruments aloft with a small balloon to monitor weather conditions in the upper atmosphere.

CLIMATE BY COMPUTER

Computer simulation of complex systems like the atmosphere is a tricky business, but two IBM scientists are trying to use computers to answer at least one pressing question: Is dust pollution contributing to the global cooling trends? Their tentative answer is negative. "Initial results indicate that the effect of dust on the earth's [climate] has been overestimated."

LOST CITY METEORITE FALL

On January 9, 1970, a 22-pound meteorite found outside the small farming community of Lost City, Okla., 45 miles east of Tulsa, turned out to be an object of historic significance. It was the first meteorite ever to be recovered in a search guided by trajectory information computed from photographic data, and it was the second meteorite whose orbit around the sun, prior to entering the earth's atmosphere, was determined from photographic observations.

The find was made by staff members of the Smithsonian's Prairie Network, a system of 16 automatic cameras in seven states that was set up to photograph fireballs and aid in the recovery of meteorites. The Lost City Meteorite, discovered in a snow-covered field five days after the fireball was photographed over northeastern Oklahoma on the night of January 4, was the Network's first successful recovery of freshly fallen material.

The fireball, brighter than a full moon, was seen from as far away as central Nebraska. Traveling in an east-southeast direction, it caused sonic booms heard from Tulsa to Tahlequah — 60 miles apart. Network films of the meteorite's descent were quickly analyzed; and, with the help of a computer to compare photos from two different stations, the impact point was predicted to be a spot three miles east of Lost City. On January 9, the 22-pound specimen was found only a half-mile from the predicted impact point.

Computer Simulation of the Atmosphere

Since the nineteen-hundreds, weather scientists have known that all weather is part of a complex global fabric, and that conditions in one region are affected by those in neighboring regions. However, with inadequate knowledge of atmospheric physics and poor data-gathering facilities, global forecasting remained a dream until after World War II. The war sparked vigorous weather research, and meteorologists for the first time began building numerical models that bore some semblance to reality. More important, the first computers—originally used for ballistics ranging—became available for peacetime use. In 1946, famed computer pioneer John Von Neumann saw the value of high-speed computing for meteorology and began to assemble a group of brilliant young scientists at Princeton University. Using a machine known as the MANIAC (for Mathematical Analyzer, Numerical Integrator and Computer), Von Neumann's group in 1950 made a first—and wildly successful—computer run of their model. But later tests revealed inadequacies—according to one account, the computer once forecast a blizzard for Georgia in July.

Since then, computers and models alike have grown steadily more sophisticated: computer simulation remains an expensive and arcane specialty flourishing at only a handful of laboratories, including U.C.L.A., the Rand Corporation, the National Center for Atmospheric Research in Boulder, Colo., England's Meteorological Office and Princeton, where the descendants of the original group have continued Von Neumann's work. Now funded by the National Oceanic and Atmospheric Administration, the Princeton group is using the world's largest and fastest computer—an Advanced Scientific Computer made by Texas Instruments.

For purposes of numerical simulation, the earth's entire atmosphere is divided into boxes extending several hundred kilometers on a side and a kilometer or so in depth A typical model may deal with 60,000 of these boxes. The computer is fed information about the boxes and about the basic laws of physics. It is then asked to compute on the basis of these laws, what will happen to the molecules in each of the boxes as temperature, humidity and wind speed change in neighboring boxes. In other words, it is asked to predict the weather all over the world, and to repeat this prediction every five minutes or so for as long as the model holds together.

The accuracy and range of the prediction obviously depend upon the reliability of the data and the model—and perhaps upon some intrinsic limits not yet understood. "We're now issuing five-day forecasts," says Donald Gilman, head of the long-range forecast division of the National Weather Service. "The consensus is that these models may let us see 10 to 14 days ahead for our daily predictions, although estimates range from one to four weeks. We are appreciably more accurate than we were 20 years ago, but it may be difficult to go on from here. That's one of the things the Global Atmospheric Research Program is designed to tell us: how much further we can expect to get. These models are very sensitive to little disturbances. If you give the model any sort of random kick, such as an error in wind speed, on day one the results you get three months later are very, very different from what you get without the kick. It will be very difficult to distinguish small but real atmospheric disturbances from random background 'noise.'"

To predict climatic trends years or decades in advance, it is clearly impractical to recompute the world's weather every five minutes. Even with large "boxes," it takes tens of hours to run a model for a prediction of a week or two. With finer, more accurate grids, say 65 kilometers on a side, computation time becomes prohibitive.

LESS THAN 1% OF THE EARTH'S SURFACE WATER IS DRINKABLE

While 70% of the earth's surface is covered with water, only 1% is "fresh" . . . and a substantial amount of *that* is polluted.

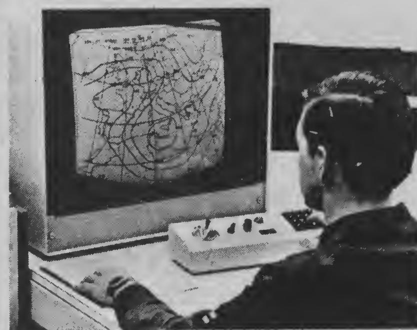
NASA HAS A SATELLITE (E.R.T.S.) 570 MILES IN SPACE . . . TO MONITOR EARTH'S WATER RESOURCES

The Earth Resources Technology Satellite orbits over the same spots on Earth every 18 days. It can thus detect deterioration of water resources.

Navy display speeds weather forecasts

A communications and display system in prototype operation at the Navy's Fleet Numerical Weather Central in Monterey, Calif., is expected to improve and speed weather forecasts dramatically.

Developed by Genisco Technology Corp., Compton, Calif., the Naval Environmental Display Station (NEDS) provides full communications, remote processing, automatic graphic storage, retrieval and TV display capability up to 80 data bits. The system incorporates a special data-compression technique that permits the Naval Weather Service to use the existing Teletype network for transmitting weather and oceanographic data to the fleet.

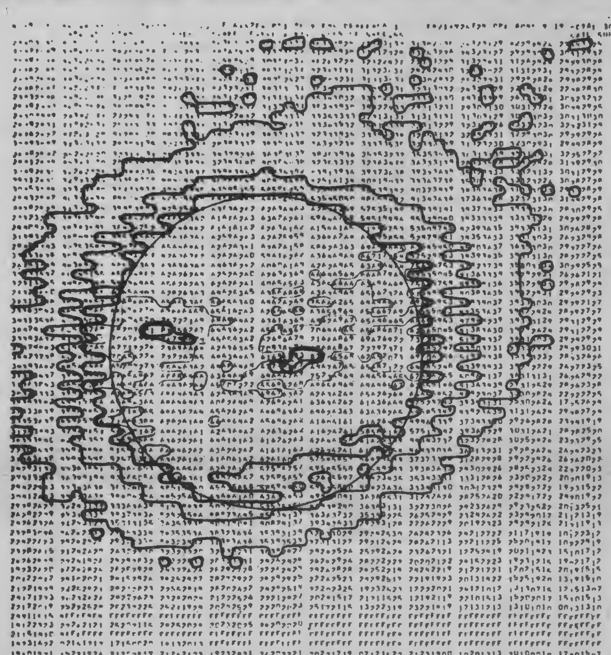


Weather conditions are observed on a CRT display of the Naval Environmental Display Station now under test at Monterey, Calif.

Traditionally the service uses facsimile equipment to transmit graphic data over costly, wideband transmission lines. Weather and oceanographic maps received over the system often are of poor quality and difficult to interpret. Correlation is done manually, and it involves the overlaying of maps by hand to make predictions.

The system being tested has two TV monitors and a keyboard that permits a forecaster to view alphanumeric and color graphic material, while a plotter/printer makes copies of any material of interest.

All data received, selected and stored are automatically logged into a computer index and become available, upon demand to the forecaster. He can call for a CRT display of the index, which lists the weather maps, messages and other data in the system's disc storage. He then calls for a display of the desired weather maps, via the keyboard, to do his forecast. ■■



COMPUTER PICTURE OF THE SUN

This is a computer picture taken by the Naval Research Laboratory ultraviolet experiment aboard the Orbiting Solar Observatory-7 launched from Cape Kennedy on September 29, 1971. The picture, received at NASA's Goddard Space Flight Center, shows the sun's disc and inner corona out to two solar radii (1,382,000 km. or 864,000 miles). The smooth circle depicts the approximate size and position of the visible sun. The wiggly lines are isotopes, separating regions of different ultraviolet intensity as on a countour map. Two regions of intense solar activity in the center of the disc are apparent. This image was recorded just two hours after a solar flare occurred in the region near the center of the disc which ultimately stretched off to beyond two solar radii.
(Photo courtesy NASA).

Lehigh Offers Decks For World Models

Complete programming for a variety of "World" models now is available from Lehigh University, according to W. E. Schiesser.

The Fortran IV source decks are available — at prices ranging up to $30 per deck of 1,800 cards — to allow computer-running of the Forester World 2 model, the Behrens natural resource utilization model, the Boyd extension of the Forrester World 2 model, and the Battelle Globe 6 model.

Dr. Schiesser informs us that two introductory models have just been released, one on the world food problem which deals with the ultimate carrying capacity of the world's agricultural system. The other model is on the world energy system; it contains the essential elements of the supply/demand interaction for five major sources of energy, with intersource competition.

Additional information, including references to documentation on the different models, is available from Prof. Schiesser at the Computing Center, Packard Laboratory, Lehigh University, Bethlehem, Pa. 18015.

"Human history becomes more and more a race between education and catastrophe."
H. G. Wells

Hunting Tornadoes

For the past two years Dr. Bruce Morgan has spent a portion of each spring zigzagging across Oklahoma's checkerboard of farms and oil fields . . . searching. In the spring of 1973 he found what he was looking for—a powerful storm which spawned a tornado before his eyes.

"It was a very peculiar sight," Morgan recalled. "The sun was shining where we were. There was no sound; we couldn't hear anything. Very light debris—tiny pieces of paper—floated down, blowing in the wind like snow. Three miles away from us this 3,000-foot tornado looking something like a gigantic ice-cream cone was smashing through Union City, Oklahoma. This big white column was just grinding its way across the ground."

Morgan and a three-person team from the National Severe Storms Laboratory (NSSL) recorded the tornado on film and made qualitative scientific observations. The team shot more than 40,000 frames of film by the time the funnel finished its 10-mile path of destruction and curled back up into the clouds.

After targeting a region and obtaining a detailed forecast from the National Severe Storms Forecasting Center in Kansas City, the team drove to that area and positioned itself on the southeastern edge of the storm, the traditional spawning site for tornadoes. As the team chased a storm, often 300 miles in a day, it received updated weather information via radio-telephone from the NSSL base.

Although it was a record lean year for tornadoes, the storm-tracking unit went out 18 times during the first spring and monitored 14 storms, two of which produced small tornadoes. It was enough of a success to convince NOAA to renew the project for another year.

The next spring, the storm trackers had more severe weather than they could handle. The United States received the deficit of tornadoes plus a few extra. Tornadoes formed at a record rate throughout the entire country.

For Morgan and the other members of the chase crew, the successful tracking and photographing of the Union City tornado was the highlight of the season. They believe much valuable scientific information can be culled from the Union City film. For example, the tornado's size versus time can be reconstructed and compared to the various computer models which have been developed.

WEATHER FORECASTING BY SATELLITE AND COMPUTER

Almost 15 years ago, April 1, 1960, a new era in meteorology began with the launching by NASA from Cape Canaveral of the world's first weather satellite TIROS-1 (Television Infared Observation Satellite). Today a facsmile of the day's weather by satellite (ESSA Series) is transmitted by computer to the data center at the Environmental Science Services Administration at Suitland, MD. (Photo courtesy NASA).

Minis monitor weather at nuclear power sites

The Atomic Energy Commission requires that proposed nuclear power-station sites be monitored for weather conditions two years before the start of construction, all during construction and for two years after the beginning of operation. A computer-based system, operated by Digital Graphics Inc., Rockville, Md., has been monitoring five unattended sites since January in accordance with AEC requirements.

At each site 32 weather-monitoring instruments, installed on a 400-ft tower, are sampled once every 15 minutes by an on-site minicomputer—a Varian 620/L. In addition to gathering data, the computer checks the quality of data to indicate instrument malfunctions.

At four-hour intervals, each remote site is contacted via commercial telephone lines by a central Varian 73 minicomputer, which gathers the data. The central computer also resets the clock at the site, clears the memory and can provide program updates. It will print an alarm message if any instruments appear to be malfunctioning.

At infrequent intervals, the central computer serves as a time-sharing terminal for a large computer, transmitting many months' worth of processed weather data. The large computer is then used to simulate conditions such as probable vapor drift from a cooling tower or accidental nuclear-particle release.

The Automobile and Air Pollution*

by Herbert D. Peckham, Gavilan College

The most logical place to begin our study of air pollution is with the automobile. Table 1 presents some very interesting statistics concerning the relationship between cars and air pollution. First, the overall quantity of pollutants (141 million tons per year) is absolutely depressing! Second, the automobile plays a discouragingly large part in the overall pollution. Certainly in the production of carbon monoxide (93% of the total) and organics (66% of the total), the automobile is the villain! Last, it is clear that the automobile has little to do with pollution from sulfur oxides and particulates.

Table 1 – Total US Air Pollution (1970)

Pollutant	Millions of Tons Per Year			% Caused by Auto
	Auto	Other	Total	
Carbon Monoxide	66	5	71	93
Organics	12	7	19	63
Oxides of Nitrogen	6	7	13	46
Sulfur Oxides	1	25	26	4
Particulates	1	11	12	8
Total Pollutants	86	55	141	61

Before we can start building our automobile air pollution models, we need to know the rates at which automobiles produce the various pollutants. Of course, this is continually changing as automobile pollution controls become more severe. We will use 1970 estimates (hopefully, by 1980 or 1990 the values will be much lower). Also, we will assume a standard velocity of 40 miles per hour. Do you feel this is a reasonable choice? The rates of pollutant production are given in Table 2.

Table 2 – Average 1970 Pollutant Production Rates per Automobile Traveling at 40 MPH

Pollutants	Rate of Production		
Gases	liters/mile	cubic feet/mile	grams/hour
Org.	4.5*	6.4	483
NO_x	3.4**	4.8	231
SO_x	0.1***	0.14	11.4
CO	54.1	76.6	2710
	(grams/mile)		(grams/hour)
Particulates	0.5		20

**Assumes an average molecular weight of 60. Gas volumes computed at standard temperature and pressure.*

***Assumes equal parts nitric oxide and nitrogen dioxide are formed. Average molecular weight of oxides of nitrogen assumed to be 38.*

****Assumes 4 parts sulfur dioxide to 1 part sulfur trioxide are formed. Average molecular weight of sulfur oxides assumed to be 67.*

* This activity is reproduced from the Student Lab booklet *Air Pollution* from the Hewlett-Packard Computer Curriculum series. Additional background material and exercises are in the booklet. The Student Lab book and companion Teachers Advisor book are available for $1.00 each from Hewlett-Packard Computer Curriculum Project, 333 Logue Ave., Mountain View, California 94043.

Let's take an average residential district, composed of a mixture of apartments and single family dwellings, as the subject of our first air pollution model. Suppose that the residential district is square with one mile sides and that we are concerned with the air over the district up to an elevation of 500 feet. Moreover, we will assume that no air passes in or out of our residential district and that any pollutants created are uniformly distributed through the air up to our "ceiling" of 500 feet.

These assumptions are typical of the ones we will be making continually. Certainly they are crude, and you may be in complete disagreement. However, experience shows that it is a valid approach to start with a very crude model and then refine it.

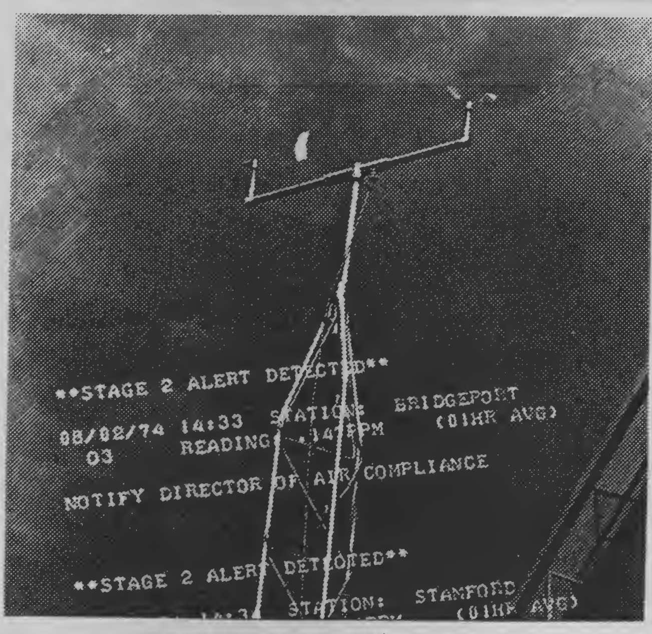
EXERCISE 1 – Estimating Number of Cars

> *How many automobiles would you expect to find in our residential area? How many automobiles would you expect to find running at some arbitrary time? You will have to make some assumptions to reach your answer. Be sure and state these assumptions explicitly. Compare your assumptions to those of other students. Do your assumptions stand up well under close examination?*

Now that you have estimated the number of cars, we will structure our first model. Let P stand for the number of cubic feet of pollutants at any time, R for the number of cubic feet of pollutants produced per hour by each car, and N for the number of cars operating at any given time. The simplest model we could construct would be

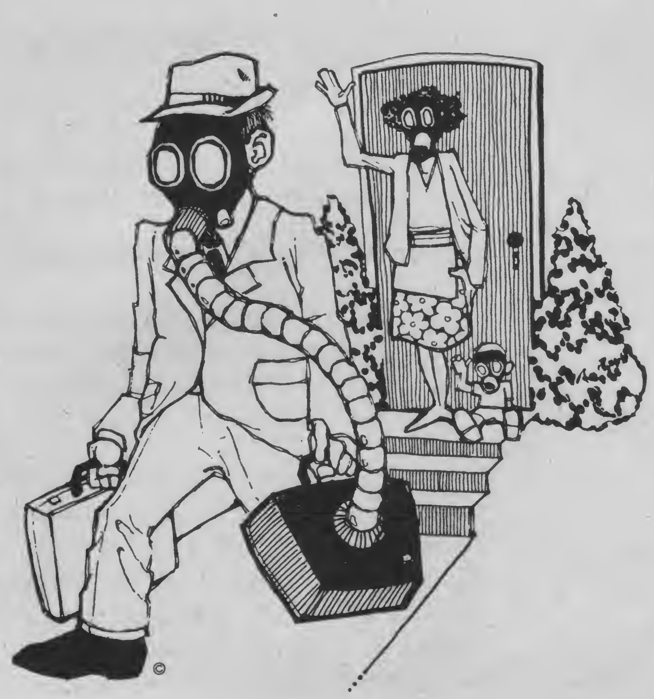
$$P_{new} = P_{old} + (R)(N). \qquad (1)$$

P_{new} is the amount of pollutants at the end of any hour. P_{old} is the amount at the end of the previous hour.



COMPUTER ALERTS CONNECTICUT TO AIR POLLUTION LEVELS

Connecticut's Department of Environmental Protection (DEP) is counting on a computer and electronic sensors to help fight air pollution. The IBM System/7 in Hartford, automatically records, analyzes and informs DEP of air pollution levels gathered by mobile trailers filled with electronic monitoring equipment. If pollutants rise beyond normal ranges the computer triggers a bell alarm to alert the air compliance director. Stage 2 Alerts are issued when people with heart or respiratory conditions might be affected by the air pollution. (Photo State of Connecticut)



EXERCISE 11 – Closing Down Freeways

> *Use the program from Exercise 8 and any wind velocity you desire. What happens to C if the freeways are shut down at some particular instant? Sketch a rough graph of the program printout.*

We still have serious flaws in our automobile air pollution model. We have been assuming *constant* values of N, W, and R_2. Clearly, this isn't realistic. It is common experience that there are morning and afternoon traffic rush hours, and that very little traffic is on the street in the middle of the night. Also, the wind rarely blows with constant velocity. Finally, the dissipation rate R_2 is certainly not constant. As we discussed previously, photochemical smog is the result of organics, oxides of nitrogen and sunlight. It stands to reason that R_2 should be smaller during hours of sunlight than during hours of darkness.

It will be fairly easy to take these ideas into account and make our model *much* more realistic. The key is to assume *maximum* values of N, W, and R_2, then take hourly decimal parts of the maximum values. Thus we can set up one list of 24 factors to be applied to N, another list for W, and so on. Each of these lists constitutes a time profile of each factor. Now, the model is

$$P_{new} = P_{old} + R_1 X_i N - Y_i W P_{old}/50 - Z_i R_2 P_{old}\,. \qquad (4)$$

X_i is the traffic profile factor (applied to N), Y_i is the wind profile factor, and Z_i is the dissipation profile factor. The subscript i is the hour number (1 to 24). So we can compare results, let's agree that hour number 1 in any day is from midnight to 1 a.m.

EXERCISE 12 – A Time Dependent Model

> *Write a BASIC program to evaluate the model given by (4) applied to the freeway example in Exercise 8. Assume reasonable sets of values for X, Y, and Z. Print out C every hour. Sketch your results in a simple graph.*

EXERCISE 13 – Political Questions

> *Use the model developed in Exercise 12 on a system whose characteristics are specified by you. Run the program to get a feel for the pollutant concentrations that come out of the model. Now, suppose that the edict has come down to cut down on pollution. Use your model and program to investigate the question. Make* realistic *suggestions as to how the pollution concentration from automobiles might be cut down.*

KEEP
AMERICA
BEAUTIFUL

Dynamic Modelling Using FORTRAN IV

Jay Martin Anderson
Bryn Mawr College

Introduction

"Dynamic modelling," as used in this paper, means the construction of formal models of systems whose behavior in time is followed by computer simulation. Specifically, the paper will refer to the techniques of System Dynamics, as pioneered and developed by M. I. T. Professor Jay W. Forrester.[1,2] System Dynamics is a general theory of system structure which rests on four essential elements:

(1) The cause-and-effect links between elements of a system and the position of these elements within feedback loops are identified.
(2) The model is expressed in a formal, mathematical language in which the qualitative interactions identified in (1) are made quantitative.
(3) The behavior of the model is examined by computer simulation.
(4) The consequences of changing system structure are evaluated by iterating on steps (1) - (3) until a viable policy or set of policies for the system under study has evolved.

The integrity, if not the beauty, of System Dynamics has often been commented upon by Forrester[3] and his students and colleagues. Our purpose here is not to debate the merits of System Dynamics as a technique or theory, nor to expound its practice, but rather to focus on step (3) of the preceding four-step program: the computer simulation of System Dynamics models.

In recent years a number of System Dynamics models have reached the public eye, including Forrester's[4] and Meadows'[5] World models, and the several environmental models described in *Toward Global Equilibrium.*[3] These models are cast in the computer language DYNAMO[6], developed expressly for the purpose of serving the System Dynamics community. DYNAMO affords a one-to-one relationship between computer equations and System Dynamics concepts, assumes for itself the labor of arranging the equations in a computable order and providing printed or plotted output. DYNAMO is a compile-and-go processor which provides its own careful diagnostics, and is available for use in a limited number of computational environments from Pugh-Roberts Associates, 65 Rogers Street, Cambridge, Mass. 02142.

In spite of its simplicity and beauty, DYNAMO falls short in classroom situations for at least two reasons. First, it is not widely available, and, in all but the versions for IBM OS/360 and IBM CP/CMS on the 360/67, it is an expensive proprietary product. Second, because it is a compile-and-go processor, there is no opportunity to form load modules for repetitive classroom use; the cost of recompiling the source program must be borne at every use.

It is to these shortcomings that the present paper is addressed. A recipie is provided for translating System Dynamics models or existing DYNAMO programs into FORTRAN. In following this recipie, the FORTRAN programmer takes upon himself much of the effort that the DYNAMO processor does for the DYNAMO programmer. Nonetheless, the result is a program which is considerably more "transportable," and which can reside as a load module for frequent classroom execution.

It will be assumed that the reader is familiar with the elements of System Dynamics as contained in *Principles of Systems*[2]. The particular recipie presented here is cast in IBM FORTRAN IV(G1) but can easily be modified for other dialects. The recipie treats only a subset of DYNAMO, but a subset wide enough to accomodate, for example, the WORLD models.

The emphasis is on *recipie*: a method for formulating System Dynamics models in FORTRAN, but not a program nor a compiler nor a processor for so doing. The recipie admits some latitude, both in the use of particular ingredients and in the embellishments possible in a well-equipped kitchen.

One example of the recipie is presented here: Forrester's World Model.[4] Two other examples along with the technical appendices are available from the author. They are a model for "The Tragedy of the Commons" and a predator-prey model illustrating one of the concepts in *The Silent Spring.* These two have both been used in the undergraduate classroom.[7]

The Recipie

The purpose of the dynamic modelling program is to describe the behavior in time of generalized systems. Mathematically, this behavior is the result of integration of coupled differential equations. It is presumed that rates of change are sufficiently slow that integration may be accomplished by a simple coarse-grid approximation to the area under a curve comprised of straight line segments. Rates, auxiliary variables, and levels may be calculated; up to ten such quantities may be tabulated in printed form and up to five may be plotted, although the FORTRAN programmer may easily circumvent these arbitrary limits.

The main program includes seven sections. These are Specifications, Functions, Inputs, Initialization, Auxiliaries and Rates, Outputs, and Levels. This seven-part structure corresponds to DYNAMO's ability to order modelling equations. Within each part, the order of the equations must be carefully planned by the FORTRAN programmer.

Specifications and Functions. These sections may be thought of as essentially instructions to the FORTRAN compiler; the remaining five sections form the logical flow of the modelling program, as shown in Figure 1.

Input. This section reads control information for the simulation, as well as values of constants, table-functions, and initial values. Information for the plotter subroutine is also read at this point. Parameters of the model may be printed to help clarify and annotate the subsequent output.

Initialization. This section provides for starting the simulation clock, some housekeeping, and setting initial values of all levels.

Auxiliaries and Rates. In this section the computation of auxiliaries and rates from existing levels, and from previously calculated auxiliaries or rates, is carried out.

Output. Results of the simulation can be printed line by line as the simulation proceeds, but information for plotting is best saved until an entire page of graphical output has been accumulated. The arbitrary limits of ten printed and five plotted variables were chosen for simplicity in constructing a page-wide line of tabular information and for clarity in reading simultaneous plots. A print-plot subroutine which forms plots on a line-printer much like those formed by DYNAMO, is described in the Appendix. Clearly the FORTRAN programmer with more sophisticated graphical devices will wish to call upon these in writing output.

Levels. The integration is completed, and the clock and levels are updated. The details of the seven-step "recipie" are given in the appendix.

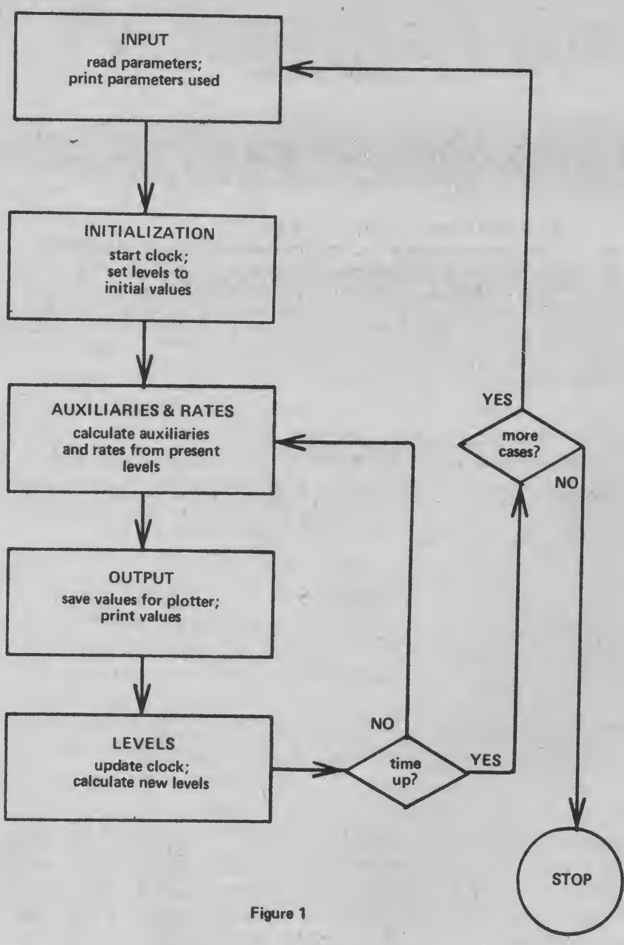


Figure 1

Required Subroutines

The recipie described briefly above and in detail in the Appendix requires additional subroutines to ease in writing DYNAMO-like programs. A limited subset of those provided for DYNAMO users might include CLIP, NOISE, RAMP, STEP, SWITCH, TABLE, and TABHL, as well as a simulation-plotter subroutine (SIMPLT). The Appendix includes source listings for the above.

It must be stressed that delays, which appear frequently in System Dynamics models, are not simply functions or subroutines, but "macros." Delays require integration, and therefore the programming of a delay requires insertion of FORTRAN source statements into several of the seven sections of the modelling program. An example, first- and third-order information delays (DYNAMO macros SMOOTH and DLINF3) are exhibited in the Appendix.

Output

The output of the program comprises three parts: a summary of parameters used in the model, a printed table, and a plot of various rates, levels, or auxiliaries as a function of time. The programmer may choose not to form a printed table or not to form a plot if he wishes, or to present the output in some other way. In the example, primary emphasis is placed upon a simple but crude print-plot, which should be within reach of all computer systems.

Example

The FORTRAN transcription of Forrester's World model, taken from his *World Dynamics* (reference 4) constitutes a simple and easily manageable program of just over a hundred lines. *World Dynamics* is ample documentation for the model. The source listing and "standard" simulation are shown here.

Appendices

Four appendices to this paper including a detailed explanation of the "Recipie," various subprograms, and three additional examples are available (free) direct from the author. The author has also offered to aid others in implementing these models, and will supply (at cost) card copies, source listings, and further documentation for each. Please write:

Jay Martin Anderson
Department of Chemistry
Bryn Mawr College
Bryn Mawr, Pennsylvania 19010

---

[1] Forrester, J. W., *Industrial Dynamics.* MIT Press, Cambridge, Mass. 1961.

[2] Forrester, J. W., *Principles of Systems.* Wright-Allen Press, Cambridge, Mass. 1968.

[3] See, for example, Forrester, J. W., "Counterintuitive Behavior of Social Systems." In *Toward Global Equilibrium: Collected Papers.* D. L. Meadows and D. H. Meadows, editors. Wright-Allen Press, Cambridge, Mass. 1973. First published in *Technology Review* 73(3): 3, January, 1971.

[4] Forrester, J. W., *World Dynamics.* Wright-Allen Press, Cambridge, Mass. 1971.

[5] Meadows, D. H., D. L. Meadows, J. Randers, and W. W. Behrens III. *The Limits to Growth.* Universe Books, New York. 1972.

[6] Pugh, A. L. III. *DYNAMO II Users' Manual.* 2nd edition. MIT Press, Cambridge, Mass. 1973.

[7] Anderson, J. M., "Computer Simulation in the Dynamics of Environmental Systems." In *Proceedings of the Fifth Conference on Computers in the Undergraduate Curricula.* Washington State University. Pullman, Washington. 1974.

PROGRAM LISTING

```
C   WORLD2
C   FROM WORLD DYNAMICS, BY J.W.FORRESTER (C) WRIGHT-ALLEN PRESS, 1971
C   FORTRAN IV(G1) TRANSCRIPTION, J.M.ANDERSON, SEPT 1974
C
C   1.  SPECIFICATIONS
      DIMENSION NAME(18),PP(51),PNR(51),PCI(51),PPOL(51),PQL(51)
      LOGICAL RPLOT/.TRUE./
      COMMON T,PP,PNR,PCI,PPOL,PQL
      REAL NR,NRUR,NRUN1,NRFM,NRFR,MSL,NRMM,NRI,NRUN,LA
C   TABLE ENTRIES FOR TABLE LOOKUP FUNCTIONS
      REAL BRMMT(6)/1.2,1.,.85,.75,.7,.7/,NREMT(5)/0.,.15,.5,.85,1./,
     1 DRMMT(11)/3.,1.8,1.,.8,.7,.6,.53,.5,.5,.5,.5/,
     2 DRPMT(7)/.92,1.3,2.,3.2,4.8,6.8,9.2/,
     3 DRFMT(9)/30.,3.,2.,1.4,1.,.7,.6,.5,.5/,
     4 DRCMT(6),BRCMT(6),BRFMT(5)/0.,1.,1.6,1.9,2./,
     5 BRPMT(7)/1.02,.9,.7,.4,.25,.15,.1/,
     6 FCMT(6)/2.4,1.,.6,.4,.3,.2/,FPCIT(7)/.5,1.,1.4,1.7,1.9,2.05,2.2/,
     7 CIMT(6)/.1,1.,1.8,2.4,2.8,3./,FPMT(7)/1.02,.9,.65,.35,.2,.1,.05/,
     8 POLCMT(6)/.05,1.,3.,5.4,7.4,8./,
     9 POLATT(7)/.6,2.5,5.,8.,11.5,15.5,20./,CFIFRT(5)/1.,.6,.3,.15,.1/,
     A QLMT(6)/.2,1.,1.7,2.3,2.7,2.9/,QLFT(5) /0.,1.,1.8,2.4,2.7/,
     B QLCT(11)/2.,1.3,1.,.75,.55,.45,.38,.3,.25,.22,.2/,
     C QLPT(7)/1.04,.85,.6,.3,.15,.05,.02/,CIQRT(5)/.7,.8,1.,1.5,2./,
     D NRMMT(11)/0.,1.,1.8,2.4,2.9,3.3,3.6,3.8,3.9,3.95,4./
      NAMELIST /PARMS/ NRUN1,POLN1,DRCMT,BRCMT,CIGN1,BRN1,FC1,DRN1,NRI,
     1 CIDN1,YEAR
      DATA BRN,ECIRN,NRUN,DRN,LA,PDN,FC,FN,CIAFN,CIGN,CIDN,POLS,POLN,
     1 CIAFT,QLS /.04,1.,1.,.028,135.E6,26.5,1.,1.,.3,.05,.025,3.6E9,
     2 1.,15.,1./
C   2.  FUNCTIONS
C   THERE ARE NONE
C
C   3.  INPUTS
      READ (5,1001) NAME
1001  FORMAT (18A4)
      READ (5,*) NUPL,NUPR,NU,NCPU
      NOCAL=NU*NCPU
      NUPT=NUPL*NCPU
      NUPR=NUPR*NCPU
      DT=1./FLOAT(NCPU)
      NCASE=0
1     READ (5,PARMS,END=100)
      NCASE=NCASE+1
      WRITE (6,1003) NAME,NCASE
1003  FORMAT ('1',18A4/' CASE ',I2/)
      WRITE (6,PARMS)
C
C   4.  INITIALIZATION
      NT=0
      P=1.65E9
      NR=NRI
      CI=0.4E9
      POL=0.2E9
      CIAF=0.2
2     NPL=0
C
C   5.  AUXILIARIES AND RATES
5     T=1900.+FLOAT(NT)/FLOAT(NCPU)
      CR=P/(LA*PDN)
      CIR=CI/P
      NRFR=NR/NRI
      CIRA=CIR*CIAF/CIAFN
      POLR=POL/POLS
      NREM=TABLE(NREMT,NRFR,0.,1.,.25)
      ECIR=CIR*(1-CIAF)*NREM/(1-CIAFN)
      MSL=ECIR/ECIRN
      BRMM=TABHL(BRMMT,MSL,0.,5.,1.)
      DRMM=TABHL(DRMMT,MSL,0.,5.,.5)
      FCM=TABLE(FCMT,CR,0.,5.,1.)
```

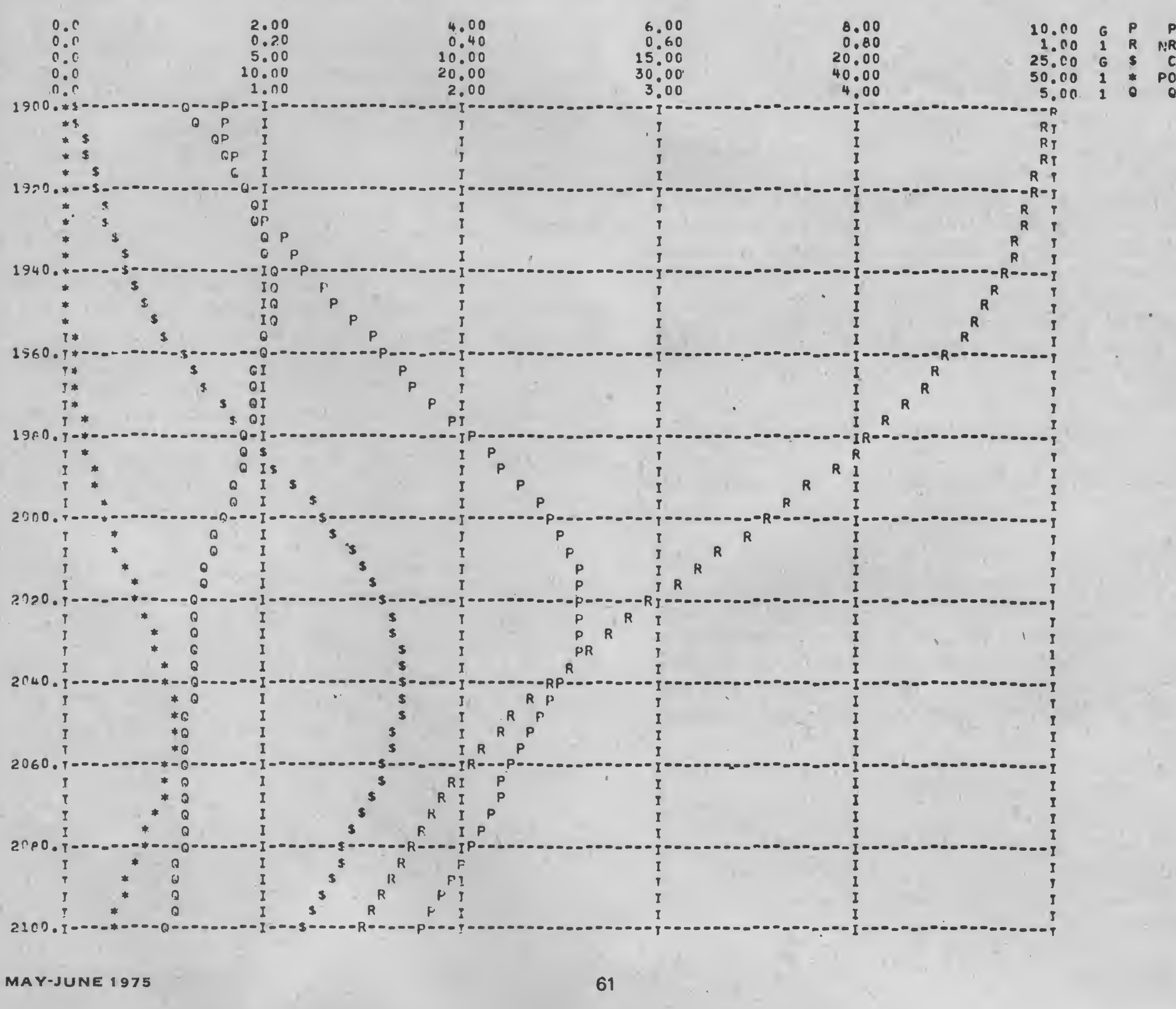
```
      FPCI=TABHL(FPCIT,CIRA,0.,6.,1.)
      FPM=TABLE(FPMT,POLR,0.,60.,10.)
      FR=FPCI*FCM*FPM*CLIP(FC,FC1,YEAR,T)/FN
      DRPM=TABLE(DRPMT,POLR,0.,60.,10.)
      DRFM=TABHL(DRFMT,FR,0.,2.,.25)
      DRCM=TABLE(DRCMT,CR,0.,5.,1.)
      BRCM=TABLE(BRCMT,CR,0.,5.,1.)
      BRFM=TABHL(BRFMT,FR,0.,4.,1.)
      BRPM=TABLE(BRPMT,POLR,0.,60.,10.)
      CIM=TABHL(CIMT,MSL,0.,5.,1.)
      POLCM=TABHL(POLCMT,CIR,0.,5.,1.)
      POLAT=TABLE(POLATT,POLR,0.,60.,10.)
      CFIFR=TABHL(CFIFRT,FR,0.,2.,.5)
      QLM=TABHL(QLMT,MSL,0.,5.,1.)
      QLC=TABHL(QLCT,CR,0.,5.,.5)
      QLF=TABHL(QLFT,FR,0.,4.,1.)
      QLP=TABLE(QLPT,POLR,0.,60.,10.)
      NRMM=TABHL(NRMMT,MSL,0.,10.,1.)
      CIQR=TABHL(CIQRT,QLM/QLF,0.,2.,.5)
      BR=P*CLIP(BRN,BRN1,YEAR,T)*BRFM*BRMM*BRCM*BRPM
      NRUR=P*CLIP(NRUN,NRUN1,YEAR,T)*NRMM
      DR=P*CLIP(DRN,DRN1,YEAR,T)*DRMM*DRPM*DRFM*DRCM
      CIG=P*CIM*CLIP(CIGN,CIGN1,YEAR,T)
      CID=CI*CLIP(CIDN,CIDN1,YEAR,T)
      POLG=P*CLIP(POLN,POLN1,YEAR,T)*POLCM
      POLA=POL/POLAT
C
C  6.  OUTPUTS
      IF (MOD(NT,NUPT).NE.0) GO TO 20
      NPL=NPL+1
      PP(NPL)=P
      PNR(NPL)=NRFR
      PCI(NPL)=CI
      PPOL(NPL)=POLR
      PQL(NPL)=QLS*QLM*QLC*QLF*QLP
C  NOTE THAT A DYNAMO 'SUPPLEMENTARY' APPEARS ONLY IN THE OUTPUT SECTION
      IF (NPL.GE.51) GO TO 40
C
C  7.  LEVELS & FINISH
20    IF (NT.GE.NOCAL) GO TO 40
      P=P+DT*(BR-DR)
      NR=NR-DT*NRUR
      CI=CI+DT*(CIG-CID)
      POL=POL+DT*(POLG-POLA)
      CIAF=CIAF+DT*(CFIFR*CIQR-CIAF)/CIAFT
      NT=NT+1
      GO TO 5
40    CALL SIMPLT (5,NUPL,NPL,NAME,RPLOT)
      RPLOT=.FALSE.
      IF (NT-NOCAL) 2,1,1
100   STOP
      END
```

SAMPLE RUN

Sample run of Forrester's World-2 Model below shows input data and output plot. The following variables are used on the plot.

- P POP, Population, billions of people
- R NRFR, Natural resource fraction remaining, dimensionless
- $ CAP, Capital stock, billions of dollars
- * POLR, pollution relative to 1970, dimensionless
- Q QL, Forrester's index of the Quality of Life, dimensionless

```
WORLD2 BY JWF

4 0 200 5

 &PARMS NRUN1=1.,POLN1=1.,DRCMT=.9,1.,1.2,1.5,1.9,3.,

 BRCMT=1.05,1.,.9,.7,.6,.55,CIGN1=.05,BRN1=.04,FC1=1.,

 DRN1=.028,NRI=9.E11,CIDN1=.025,YEAR=1974.,&END

0.        1.     E+10P POP

0.        1.         RNRFR

0.        2.5    E+10$ CAP

0.        50.        *POLR

0.        5.         Q QL
```



Structured Programming

By Christopher G. Hoogendyk

"That's a great deal to make a word mean,"
Alice said in a thoughtful tone.
"When I make a word to a lot of work like that,"
said Humpty Dumpty, "I always pay it extra."
Lewis Carroll
Through the Looking-Glass

A lot has been said about structured programming. You might ask, "Why another article?" The answer is that no paper yet has pulled together all the ideas and showed their connection (McCracken 1973). If you ask the average programmer what structured programming is, he will reply with a collection of rules and regulations about top-down planning, avoiding GOTOs, and formatting code on a page. Such a collection of rules is hard to remember and easily misused.

There is also disagreement and confusion about when a program can be called "structured". A programmer might say he has a structured program and a dozen others will disagree. Further, as programming theory develops, better methods will be used and the definition of "a structured program" will change. Let us throw out the idea of a structured program versus an unstructured program and look at programs as fitting into a spectrum determined by the skill of the programmer, the effort he puts into the program, the language facilities, etc. In these terms we would speak of a *well structured program*. If Dijkstra put his best efforts into designing and writing a particular program in the best Algol, we might look upon it as an extremely well structured program. But, but not drawing a dichotomy, we also admit that an average programmer can apply the ideas of structured programming in the language at hand and come up with a reasonably well structured program. There is no reason to develop a powerful conceptual tool for programming and then deny its use to a wide range of programmers and applications.

Structured programming, then, deals with the design and writing of well structured programs. Now, what is the central theme of structured programming? If you can't identify a central theme and show the connection between it and each rule, then most programmers (like me) will see structured programming only as a disconnected collection of rules and regulations. This failure to pull the ideas together will be reflected in the use and misuse of the ideas behind structured programming.

The central theme of structured programming is that *structure should reflect function*. This is a powerful design concept. It was the theme of Frank Lloyd Wright's architecture. When applied to programming this concept leads to worthwhile results on three different levels: design, coding, and display.

Program Design.

The first level is the planning or design of the overall program. A program should not be a black box that is patched together until it works. It should represent a systematic development of the ideas behind the program. This idea has been carefully developed and presented in case histories of program design by Dijkstra (1972), and has come to be known as top-down program design. The practical application of top-down design in program development projects is discussed by Miller and Lindamood (1972). Very briefly, top-down design means starting with a definition of the program's purpose and progressively decomposing it into subactions until you reach a level of description that can be coded directly into the programming language you are using. There are some key ideas that emerge from top-down design which are often expressed independently. One is that you shouldn't bind yourself with an early decision about particular program or data structures. This leads to greater flexibility in program design, and makes it easier to modify the design or the program itself if the requirements are changed. In its simplest form this means that a variable name should be used in the place of a recurring constant (this is called parameterization). Thus, instead of referring to device number 3 in your program, you would refer to the INPUT device. Then, at run time, INPUT could be initialized to 3. Dijkstra (1972) gives some more instructive examples.

Program Coding.

The second level at which we can apply the concept that structure should reflect function is in the actual coding. This was first expressed by Dijkstra (1968), although the ideas have been around a while longer. Dijkstra pointed out that the real subject of programming is the process that results from the run time execution of the program. He said, "We should do our utmost to

shorten the conceptual gap between the static program and the dynamic process, to make the correspondence between the program (spread out in text space) and the process (spread out in time) as trivial as possible." This is the source of the objection to the GOTO statement. The GOTO can lead to a complicated flow of execution through a program text that can be almost impossible for a human reader to follow. Work on the formal proof of program correctness has resulted in some good systematic ideas in program coding. Bohm and and Jacopini (1966) proved the logical superfluousness of the GOTO statement and showed that any problem could be expressed using the simple control structures of sequential processes, selection, and iteration. These correspond to the normal sequential execution of a program, the IF-THEN-ELSE construction, and the FOR-NEXT or DO constructions. The inclusion of BEGIN-END blocks lends a simple versatility to these constructions. Using "structured" coding results in real benefits for the human reader. It becomes easy to read a program from top to bottom without having to look back and forth all over the program to pull pieces together. A simple example in BASIC should demonstrate the strength of these simple control structures.

```
10 LET T=0
20 LET T=T+1
30 LET N=NO*EXP(R*T)
40 PRINT T,N
50 IF T 10 THEN 20
```

```
20 FOR T=1 TO 10
30 LET N=NO*EXP(R*T)
40 PRINT T,N
50 NEXT T
```

These two pieces of code perform identical functions. In the first, the programmer starts reading, has to think about the initialization of T, reads the next two lines, sees the IF, looks back to line 20, and begins to put it together. In the second piece of code, the programmer reads top to bottom without having to stop once. When structured coding is incorporated in a large program, where there are several levels of control, the improvement in readability demonstrated above is magnified many times over.

Program Display.

The third level at which we can apply the concept that structure should reflect function is in the actual display of program code on a page. The visual structure of the code should convey to the reader as much information as possible about the functional content of the code. An effective way of doing this is to uniformly indent FOR-NEXT or DO loops, and to doubly indent nested loops. The reader who encounters the FOR or DO can then see the extent of the loop immediately without stopping to scan through to find the end of the loop. An idea from Weinberg (1971) is to have programming aids for listing programs that would do various things such as indent loops uniformly, put keywords of the programming language in boldface, move all comments to the right hand side of the page, etc. Indenting and the use of white space in vertical spacing should be used by all programmers. Additional flourishes depend on ingenuity and available printing mechanisms.

Structured programming, then, is the application and expansion, at several levels, of the concept that structure should reflect function. The usefulness and success of structured programming has been demonstrated (Baker and Mills 1973). The reason for this success is that programming is a human activity. When a program is written the work isn't done. It has to be debugged. As it is used, more bugs will be found, or it will need to be modified to fit changing needs. Few successful programs are ever static. Programmers spend the bulk of their time debugging or reworking program code. Because of this, programs have to be readable. Since structured programming focuses on clear program organization, increased information content, and greater readability, it has a striking effect.

The object of this discussion has been to show how the seemingly disconnected ideas of structured programming are united by the theme that structure should reflect function. No attempt has been made to expand the ideas into detailed discussions or case histories. Those who have a serious interest in programming should read Dijkstra's articles.

References

Baker, F. Terry, and Harlan D. Mills, "Chief Programmer Teams", *Datamation,* December 1973, P. 58.

Bohm, Corrado, and Guiseppe Jacopini, "Flow Diagrams, Turing Machines and Languages With Only Two Formation Rules" *Communications of the ACM,* May 1966, P. 366.

Dijkstra, Edsger W., "GOTO Statement Considered Harmful", *Communications of the ACM,* March 1968, P. 147.

Dijkstra, Edsger W., "Notes on Structured Programming", *Structured Programming,* Academic Press, New York, 1972.

McCracken, Daniel D., "Revolution in Programming: an overview", *Datamation,* December 1973, p. 50.

Miller, Edward F., Jr. and George E. Lindamood, "Structured Programming: Top-down Approach", *Datamation,* December 1973, p. 55.

Weinberg, Gerald M., *The Psychology of Computer Programming.* Van Nostrand Reinhold, New York, 1971.

The author graduated from Dartmouth College in January 1973 with a B. A. in biology. He has since served as chief programmer for the CONDUIT project at Dartmouth, contributing to the development of standards for the preparation of transportable programs. His main interests are ecological modelling and educational uses of computing.

Recent Trends in Mathematics Curriculum Research

Marogt Critchfield
Project Solo,
University of Pittsburgh 15260

Mathematics curriculum reform in the last 10 years (1964-74) cannot be as neatly characterized as that of the decade which preceded it. The years 1954-64 saw the rise of a 'glamour movement,' the 'new mathematics,' which markedly changed the subject matter of school mathematics. The principal mechanism used to effect these changes was textbook writing. The change was accomplished with unusual swiftness and a good deal of publicity. A number of different research projects were involved, and yet there appears to have been unanimity regarding the changes needed and the topics to be incorporated. It is this period of research which most strongly influences school practice today, and which is subject to the greatest scrutiny.

By contrast, the ten years of research 1964-74 have not yet had a widespread influence on actual school practice. Also, the unanimity of the earlier period appears to have dissipated considerably during the second. (I suspect that a deeper analysis of the earlier period might reveal more diversity, as well.) There are a number of developing trends in the period 1964-74; I will discuss three of these, which I consider most importaht.

First, there appears to be a change in the attitude of professional educators toward curriculum research. I consider this to be as important as the development of any new movement. Evidence of this can be seen in a booklet put out by the National Council of Teachers of Mathematics in 1968 [14]. In contrast to a previous booklet with similar cover design and similar title put out in 1961 [13], this collection of articles is generally cooler and contains more diverse viewpoints. While calling for still more changes in the curriculum (and for much the same reasons as the earlier book), it does so only after having described the problems as having many different facets. It calls for greater care on the part of school decision makers in choosing from the available curriculum projects.

> "They should not accept change simply because it is the current fad, nor should they assume it is successful simply because it is new."

Some of the most prominent topics of the new math — sets, non-decimal number bases, and axiomatics come in for criticism, although no blanket condemnation is expressed:

> "it will become increasingly necessary, however, for educators to make value judgments as to which topics must be stressed heavily for which children."

Significantly, I think, the pamphlet encourages local district curriculum development, which it felt had gone into a decline with the emergence of large-scale, national curriculum projects.

> "Some local directors of curriculum apparently have decided that the period of curriculum-making at the school or district level was over and that the challenge now was to select the best program available that had been developed by the 'experts' . . . the 'best' may consist of a selection of useful topics from several programs."

This second booklet spotlights the provocative curriculum suggestions of the Cambridge Conference on School Mathematics (CCSM) at the very front. (CCSM will be the second major trend discussed here.) However, the authors of this pamphlet appear to find many problems that are not helped by CCSM and the controversy it sparked.

By 1973, this cooler attitude seems to have congealed'. The report of an NSF sponsored conference that summer, made the following statements [17]:

> "There is a substantial lack of trust and communication between the mathematics education community in the universities and that in the schools. Efforts need to be instigated to re-establish cooperation."
>
> "At the present time, there seems to be no clear consensus with regard to the mathematics which should be taught in K-12 and there is an urgent need for a program which will examine societal needs and delineate the goals of mathematics education with sufficient authority to provide a broadly acceptable base for curriculum development."

The second important trend of research activity is what I term a continuation and intensification of the spirit of 'new math.' The Cambridge Conference on School Mathematics (CCSM) held its first meeting in June, 1963. It was a brainstorming session, not a textbook writing one, but its recommendations fit into the category of wanting "more and better mathematics" in schools. Further, many of the old 'new math' approaches and topics, if not their current implementation, were re-affirmed in its recommendations. CCSM produced *Goals for School Mathematics* in 1964 [2] and went on to write two more documents, *Goals for Mathematical Education of Elementary Teachers,* 1966 [3] and *Goals for the Correlation of Elementary Science and Mathematics,* 1969 [4], as well as experimental units embodying their goals.

CCSM's goals, although accompanied by warnings that they were tentative and not to be used as a blueprint, were an audacious challenge by intellectuals to the schools and curriculum developers.

> ". . . thirteen years of mathematics in grades K to 12 should [give] a level of training comparable to three years of top-level college training today; . . . two years of calculus, and one semester each of modern algebra and probability theory." [2]
>
> "We propose to gain three years through a new organization of the subject matter and the virtual total abandonment of drill for drill's sake, . . ." [2]

Some of the other features mentioned in CCSM's 1963 report were:

(1) ". . . the parallel development of geometry and arithmetic (or algebra in later years) from kindergarten on."

(2) ". . . structure of the real number system and the basic ideas of geometry both synthetic and analytic . . . considerable attention . . . to inequalities in the earliest grades."

(3) "'spiral' curriculum which repeatedly returns to each topic, always expanding it and showing more connections with other topics."

(4) "(K-6) should be understandable by virtually all students; it should lead to a level of competence well above that of the general population today."

(5) "... for those who take mathematics only a few years after grade school ... an elementary feeling for probability and statistics ... [and] ... a nodding acquaintance with the calculus."

The 1969 book [4] contains some interesting thoughts about the mathematics and science curricula in general, which reveal a fuller development of CCSM's point of view:

> "Science and mathematics, by their inherent simplicity in comparison to most areas of knowledge, lend themselves to the development in children of attitudes of lifelong and general value.... They include (in no order of priority) a conviction that through analysis and synthesis comes understanding; a belief that quantitative measure adds dimensions to one's understanding that are always difficult and sometimes impossible to achieve by other means; a tolerance that permits consideration of all reasonable testable hypotheses which are consistant with available evidence; a healthy skepticism even toward conclusions supported by existing evidence; an optimism based on the belief that nothing is unknowable while much remains unknown; and finally, a belief that to understand, while indeed a means to power, is to enjoy and is therefore an end in itself."
>
> "In his school experience with science, a child can make his own observations and organize them, then make his own predictions and check them. Thus he can directly appreciate the power of the scientific style of thought."
>
> "We do not want these experiments to be done occasionally as a sort of special treat, ... but sufficiently often that the thought patterns that underlie the world of science will be habitual, if rudimentary, in every school graduate."
>
> "A primary message of education should, we believe, be that thinking is worthwhile. Unfortunately, education has often been directed away from the imaginative and creative toward uninteresting, rote attention to details."
>
> "Each child must be convinced that *his* thinking is worthwhile."

This is very exciting intellectual stuff: it's wise and idealistic in the best sense of that word. Reading it, one almost forgets two important things: (1) In spite of a clear intention to the contrary, these goals put pressure on researchers and schools to focus attention on still more new topics and new courses and away from such perennial problems as student motivation, teacher's job satisfaction, and the real need for students to acquire essential skills in an acceptable length of time. (2) The mathematicians and scientists who have come together as CCSM conferrees seem to have only a very hazy notion of what the non-mathematics-using citizen needs:

> "...difficult and important decisions are better made by people used to connecting reality with rationality .... through the vital process of constructing simplified conceptual models for real world objects and interactions .... There are severe limitations on a quantitative approach in a real life situation; but it seems better to go as far as one can with that approach than to abandon decisions to guess or superstition."
>
> "Environmental pollution, for example, is among the most critical problems of our times. Its solution will require the active cooperation of every individual. We will not get this cooperation until every citizen understands the problem well enough to feel the importance of his own role in the solution."

In our highly organized and specialized society, I wonder whether the ability of an individual to make "simplified conceptual models" will contribute directly to his role in decision making. Also, is it realistic to imply that a process which has "severe limitations" in the hands of professionals will survive in a classroom setting lacking any but the most primitive tools for computation and analysis?

One curriculum project has grown directly out of CCSM's efforts --Unified Science and Mathematics for Elementary Schools (USMES), and another has dedicated itself to implementing CCSM's goals-- Comprehensive School Mathematics Program (CSMP). However, there is not the boundless enthusiasm of old, nor the move to make swift changes in schools. Many researchers simply do not believe that the acceleration proposed is compatible with growth in understanding and enjoyment, especially at the elementary level. Burt Kaufman, Director of CSMP, an advocate of CCSM's goals, is cautious:

> "We've simply torn down the entire curriculum and rebuilt it from scratch. It could have a very big impact if the public is ready for it but it is going to be more difficult for the teacher." [12]

The third major trend in the 1964-74 period is a very different kind of phenomenon. It has some, but not all, of the aspects of a new 'glamour movement' --computers in education. (While not strictly a development in mathematics, computing has impinged more on the mathematics curriculum than other subject areas for complex reasons, some social, some technical.) Before 1964, pioneering research in computer-assisted instruction by Bitzer at the University of Illinois and Suppes at Stanford gave rise to high hopes and some inflated statements:

> "the kind of individualized instruction once possible only for a few members of the aristocracy can be made available to all students at all levels of ability." [Suppes, 1]

In the period since, and at the present, a good deal of research has been undertaken regarding not only computer-assisted instruction but a variety of other computer uses in education. An idea of the growth of research in computing can be gotten from Figure 1. This is a chart of some major mathematics curriculum (and related) projects, selected mainly from [10] and [15]. It is not exhaustive, but should encompass most projects mentioned in widely read journals. Computer manufacturers have actively promoted their products to schools for both educational and administrative purposes. All this would point to a new educational panacea destined to fade or be absorbed without any deep effect. However, at least two features of this trend definitely set it apart from those have come (and gone) before it.

First, the adoption of computers in schools is taking place independently as well as with funded research activity. The American Institutes for Research Survey [5] * figures of secondary schools using computers for educational or instructional (not administrative) purposes are:

1965------- 2%
1970-------13%.

Even though these figures must be considered very approximate, this growth cannot be accounted for by research programs.

*A new survey is scheduled for release in 1975.

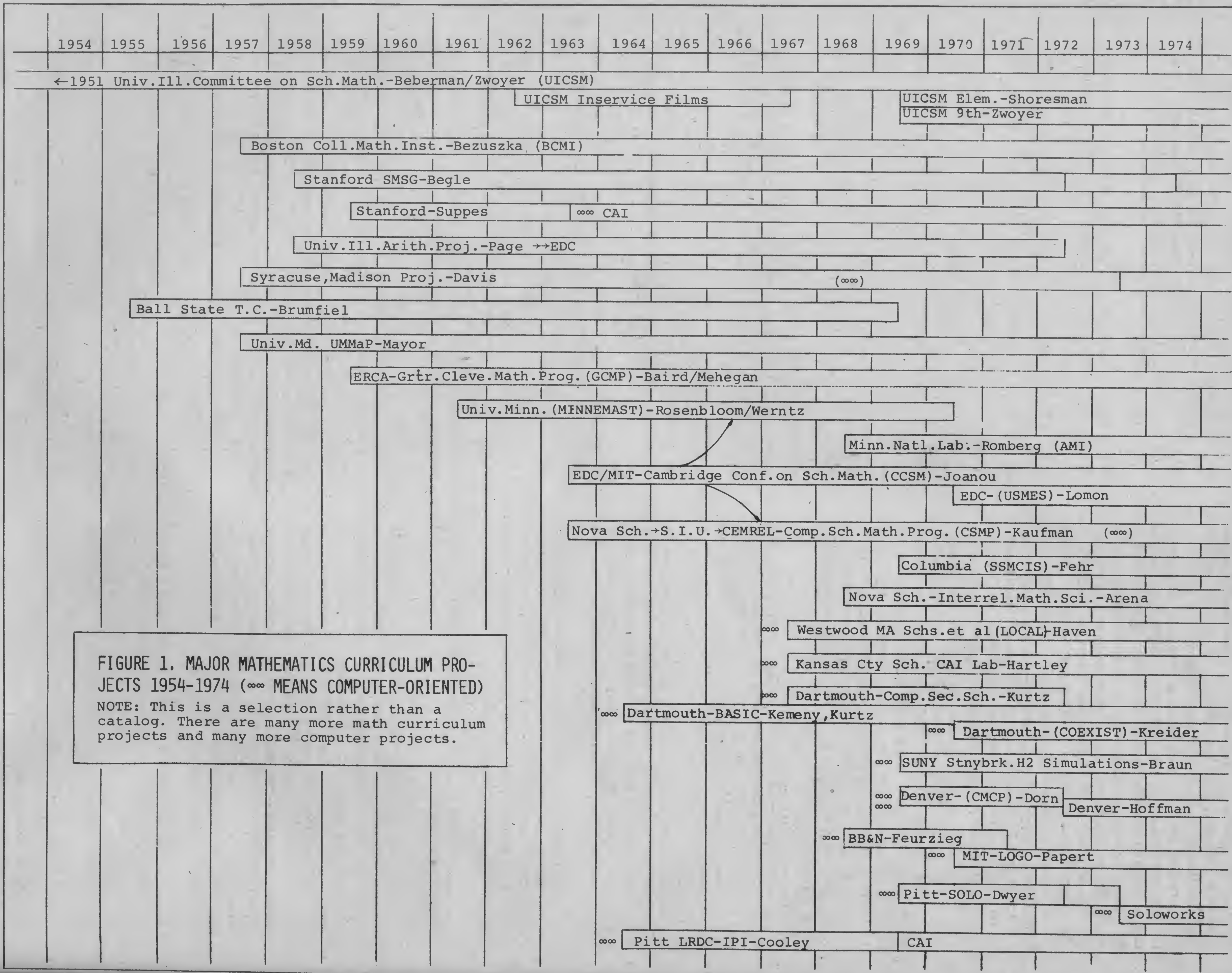


FIGURE 1. MAJOR MATHEMATICS CURRICULUM PROJECTS 1954-1974 (∞∞ MEANS COMPUTER-ORIENTED)

NOTE: This is a selection rather than a catalog. There are many more math curriculum projects and many more computer projects.

Secondly, computers are profoundly *un*like any other technical or curricular innovation that has invaded schools before. A computer can best be thought of as a machine that can transform itself into any other machine--that is, it can carry out any procedure that can be fully described to it. Therefore, it can be a medium of instruction, a lunar lander, a game player, a mathematical formula cruncher, an ecological system, and so on, ad infinitum. One writer likens the potential effects of the computer on education (and on society) to the effect that would be produced by the sudden introduction of writing and the printing press on a civilization that had developed without them [11]. Furthermore the kinds of 'machines' that a computer can be instructed to imitate range from extremely simple to extremely complex, offering the potential of a continuum of experiences for students (Figures 2 and 3)

```
GUESS   Ø5:Ø5 PM          19-OCT-74
5 RANDOMIZE
7 X=INT (RND*25)
1Ø PRINT " PICK A NUMBER FROM Ø TO 25 "
2Ø INPUT A
25 IF A >25 THEN PRINT" NUMBER TO LARGE-- TRY AGAIN": GOTO 1Ø
27 IF A <Ø THEN PRINT" NUMBER TOO SMALL--TRY AGAIN!":GOTO 1Ø
3Ø PRINT " YOU PICKED "A"!"
35 PRINT" THE COMPUTER PICKED "X" !"
4Ø PRINT " YOU MISSED BY "ABS(X-A)" !"
45 IF ABS(X-A)=Ø THEN PRINT" YOU WON !!!!! "
47 IF ABS(X-A)<>Ø THEN PRINT" YOU LOST!!"
5Ø INPUT "DO YOU WANT TO TRY AGAIN !";C$
6Ø IF LEFT(C$,1)="Y" THEN PRINT" AGAIN!" : GOTO 7
7Ø IF LEFT(C$,1)="N" THEN PRINT "THAT'S ALL!" :GOTO 9Ø
8Ø IF LEFT(C$,1)<>"Y" OR LEFT(C$,1)<>"N" THEN PRINT "WHAT--
NOT IN CORRECT FORM--PLEASE RETYPE?????": GOTO 5Ø
9Ø PRINT" BYE!!!"
1ØØ END

READY

RUN
GUESS   Ø5:Ø6 PM          19-OCT-74
 PICK A NUMBER FROM Ø TO 25
? 5
 YOU PICKED  5 !
 THE COMPUTER PICKED  2Ø  !
 YOU MISSED BY  15  !
 YOU LOST!!
DO YOU WANT TO TRY AGAIN !? Y
 AGAIN!
 PICK A NUMBER FROM Ø TO 25
? Ø
 YOU PICKED  Ø !
 THE COMPUTER PICKED  19  !
 YOU MISSED BY  19  !
 YOU LOST!!
DO YOU WANT TO TRY AGAIN !? H
WHAT-- INPUT DATA  NOT IN CORRECT FORM--PLEASE RETYPE?????
DO YOU WANT TO TRY AGAIN !? Y
 AGAIN!
 PICK A NUMBER FROM Ø TO 25
? 226
 NUMBER TO LARGE-- TRY AGAIN
 PICK A NUMBER FROM Ø TO 25
? -1
 NUMBER TOO SMALL--TRY AGAIN!
 PICK A NUMBER FROM Ø TO 25
? 5
 YOU PICKED  5 !
 THE COMPUTER PICKED  12  !
 YOU MISSED BY  7  !
 YOU LOST!!
DO YOU WANT TO TRY AGAIN !? Y
 AGAIN!
 PICK A NUMBER FROM Ø TO 25
? 7
 YOU PICKED  7 !
 THE COMPUTER PICKED  7  !
 YOU MISSED BY  Ø  !
 YOU WON !!!!!
```

FIGURE 2. Listing and run of a program, GUESS, by a beginner (Danny Cohen, Age 10).

FIGURE 3. A run of a program by an advanced student is not shown. For an example of such a run, please refer to SUPER STAR TREK found elsewhere in this issue.

Research in educational computing today encompasses a variety of approaches and the interchange of ideas is enlivened by a sharp philosophic disagreement between those researchers who believe, with Suppes, that "the truly revolutionary function of computers in education . . . lies in the novel area of computer-assisted instruction" [1], and those who believe, with Luehrmann, that "computing constitutes a new and fundamental intellectual resource. To use that resource as a mere delivery system for instruction, but not to give a student instruction in how he might use the resource himself, has been the chief failure of the CAI effort." [11], or even more strongly,

> ". . . education may have caught a tiger by the tail. It comes in the form of an activity called . . . 'solo mode' computing. Such use of computers (as the tiger image suggests) often exhibits an unexpected raw power for eliciting complex learning behaviors in all kinds of students." [7]

A taxonomy of educational computing is useful for understanding the implications of this research. The following is adapted from [8] and is intended to clarify the relationships between the types of educational computing.

COMPUTER-ACTIVATED LEARNING SCHEMA

A. DUAL MODE COMPUTING

The constraints on the learner are (primarily) pedagogically determined--the CAI lesson or other program flows from the program author's concept of how the student must proceed in learning the subject matter.

Minor skills required of the student.

Major design skills required of the program author.

I. DUAL, AUTHOR-DIRECTED COMPUTING
- --Drill and Practice } term CAI used most often here
- --Tutorials
- --Diagnostic Testing
- --Computer-Managed Instruction (CMI)[1]

II. DUAL, LEARNER-DIRECTED COMPUTING
- --Simulations
- --Games
- --Information Retrieval
- --Tutorials (sophisticated branching)[2]
- --Dialogue[2]

B. SOLO MODE COMPUTING

The constraints on the learner are (primarily) reality-determined--the student explores areas of the subject matter within the bounds of the computer system and his own imagination.

Increasing programming skills are required of students.[3]

Major guidance and some computing skills required of the teacher.

III. SOLO, HANDS-ON COMPUTING (SOMETIMES CALLED ALGORITHMIC OR PROBLEM SOLVING COMPUTING)
- --Writing programs, debugging them, running them.

IV. SOLO, LEARNER-ORGANIZED COMPUTING
- --Model Building (may include writing I, II, or III).

A higher level of student responsibility is indicated here; programs are used by others.

All of these types of computing are currently being researched and their emergence as integral parts of the newer mathematics education curricula lies in the near future. A recent conference on the K-12 mathematics curriculum gave the following recommendations regarding computing: [17]

"1. The computer should be an important part of any future curriculum efforts.

2. Emphasis should be placed on using the computer to involve students in problem solving activities. Computer use for drill and practice on computational skills should receive less attention.

3. Certain readiness concepts about the use of computers should be included in the elementary grades (1-6). These should involve the use of calculators and an exposure to algorithmic approaches.

4. In grade seven, students should be taught a programming language which is appropriate for the level of students involved. In this grade students should become familiar with information processing and the computer should be used as an integral part of the mathematics course.

5. The mathematics curriculum in grades 7-12 should be studied and revised in order to make optimal use of the computer as a tool in mathematics courses.

6. A one semester computer science course should be offered in grade 12 which may be selected as an option.

7. Societal uses and implications of the computer should be studied at some point in the school program, possibly in the 10th or 11th grade. The writing of modules on this subject that can be inserted in a social studies course is encouraged. Another possibility might be the development of a course 'Mathematics and the Computer in Society'.

8. There should be continued funding of efforts to investigate uses of the computer in a variety of instructional modes until more data are available regarding the value of these modes. Funded projects which explore the potential of different uses of the computer in education are encouraged.

9. If computers are to be systematically employed in the above ways in the schools, then the implications of this for widespread computer-access and teacher education should be effectively faced, spelled-out, and dealt with."

All these suggestions seem good and worthwhile. In fact, many of the original thoughts of the 'old' new math writers and 'new' 'new math' writers seem good and worthwhile. But the problems of their actual implementation in school are complex. The notion that such ideas can be packaged into infallible, teacher-proof forms, such as texts or CAI programs becomes more and more ridiculous.

If one impression can be derived from the history of mathematics curriculum research of the last twenty years it is that reform of curriculum (that is, the relatively tangible books, lists of topics, courses, and materials that codify and justify much of school life) must be related to more subtle and far reaching reforms. There must be conscious attention paid to the social relations that form the substructure of school life.* Such research is difficult to carry out, but the researcher involved in these reforms should, as a minimum, engage in face-to-face contacts with teachers at

[1] Computing as a research tool for developing and testing theories of learning and instruction is related to this type of computing. It is not part of this taxonomy because it does not exist at the level of on-going teaching-learning activities.

[2] Real "dialogue" is still more promise than fact. Research in this area is better characterized as part of "artificial intelligence" than education.

[3] Two books which emphasize the step-by-step build-up of programming skills needed for solo mode computing are [6] and [9].

all stages of their development, teachers of teachers, professional scholars whose specialty is education, and professional scholars in the disciplines to which the curriculum must be connected. These contacts should influence every stage of the innovation.

*Required reading for anyone engaged in making changes in schools should be Seymour Sarason's *The Culture of the School and the Problem of Change.* [16]

BIBLIOGRAPHY

1. Atkinson, Richard C. and H. A. Wilson, Eds. *Computer-Assisted Instruction, A Book of Readings,* Academic Press, 1969.
2. Cambridge Conference on School Mathematics. *Goals for School Mathematics,* (Educational Services Incorporated) Houghton Mifflin Co., Boston, 1967
3. -----. *Goals for Mathematical Education of Elementary School Teachers,* (Education Development Center, Inc.) Houghton Mifflin Co., Boston, 1967.
4. -----. *Goals for the Correlation of Elementary Science and Mathematics,* (Education Development Center, Inc.) Houghton Mifflin Co., Boston, 1969.
5. Darby, C. A., Jr. et al. *Survey of Computing Activities in Secondary Schools,* American Institutes for Research, Silver Springs, Md., October, 1970.
6. Dwyer, T. A. *A Guided Tour of Computer Programming in BASIC,* Houghton Mifflin Co., Boston, 1974.
7. Dwyer, T. A. "The Significance of Solo-Mode Computing for Curriculum Design," *EDU--Issue No. 13:* Fall, 1974. (Education Products Group, Digital Equipment Corp., Maynard, Mass. 01754)
8. Dwyer, T. A. and M. Critchfield. "CATALYST: CAI in a General Time-Sharing Environment," *EDUCOM Bulletin* (Winter, 1970).
9. Dwyer, T. A. and M. Critchfield, *A Computer Resource Book--Algebra,* Boston, Houghton Mifflin Co., 1975.
10. Lockard, J. David, Ed. *Seventh and Eighth Reports of the International Clearinghouse on Science and Mathematics Curricular Developments (1970, 1972),* (American Association for the Advancement of Science, Washington, D. C.) Science Teaching Center, Univ. of Md., College Park, Md.
11. Luehrmann, Arthur W. "Should the Computer Teach the Student or Vice Versa," *Proceedings of the Spring Joint Computer Conference,* Amer. Fed. of Info. Processing Societies, 1972.
12. Martin, Richard. "Sum & Substance," *Wall Street Journal,* May 31, 1973.
13. National Council of Teachers of Mathematics (Regional Orientation Conference). *The Revolution in School Mathematics, A Challenge for Administrators and Teachers,* 1961.
14. National Council of Teachers of Mathematics, *The Continuing Revolution in Mathematics,* 1968.
15. National Science Foundation. *Education Programs in Mathematics for Elementary and Secondary Schools,* April, 1974.
16. Sarason, Seymour B., *The Culture of the School and the Problem of Change,* Allyn and Bacon, Inc., Boston, 1971.
17. Springer, George (Dir.). *Report of the Conference on the K-12 Mathematics Curriculum (NSF), June 1973,* Mathematics Education Development Center, Indiana Univ., Bloomington, In. 47401.

Good Things From Oregon

Judging from the number of CC subscribers from Oregon and the tremendous number of people from Oregon at various conferences and meetings, it is certainly one of the leading states in computer education. We had hoped to have a comprehensive article about computing activities throughout the state, but apparently the people I spoke to are just too busy doing their own publications to do a piece for us. Hence, we'll just note several conspicuous examples of the good things happening in Oregon.

Oregon Computing Teacher produced by the Oregon Council for Computer Education, is an informal magazine of about 72 typewritten pages per issue which appears 4 times a year. It contains a variety of original and reprinted material of interest mainly to high school and undergraduate college faculty. (It is not aimed at students.) We're impressed with the uniformly high quality of this publication. It's available for $5.00/yr from Oregon Council for Computer Education, 4015 S. W. Canyon Road, Portland, OR 97221.

Computers in Education Resources Handbook is a comprehensive 500-page handbook about the uses of computers in education, primarily at the pre-college level. It covers both instructional and administrative uses of the computer although it is clearly stronger on the instructional side. It discusses hardware (lightly), software (moderately), applications (heavily), training, surveys, and sources of additional information (excellent). First published in 1973 it is quite current, even so a new edition is being published in early 1975. Available for $10.80 from Dept. of Computer Science, Univ. of Oregon, Eugene, OR 97403.

ECO-NET is a non-profit environmental education network emphasizing the exchange of information relating to the environment, energy, communications and, yes, even computers. A 16-page monthly newsletter is called *RAIN*. Despite its Pacific Northwest bias, it's one of the very best, ranking along side *Whole Earth Catalog, Epilog,* and *CoEvolution Quarterly.* At the moment, the price is right too. *Rain* is available free (until their grant runs out) from Environmental Education Center, Portland State Univ., P. O. Box 751, Portland, OR 97207.

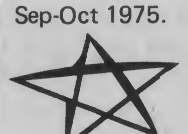
An apology: The Wizard graphic and note on pg. 25 of the Jan-Feb *Creative Computing* came from the May 1974 issue of *Oregon Computing Teacher* which we neglected to mention. Sorry.

A request: When you write for materials such as those above or from advertisers, please mention *Creative Computing.* That encourages those groups to keep us posted and/or keep advertising with us.

COMING IN CREATIVE COMPUTING!

Jul-Aug 1975. Don't look for it this year.

Sep-Oct 1975.

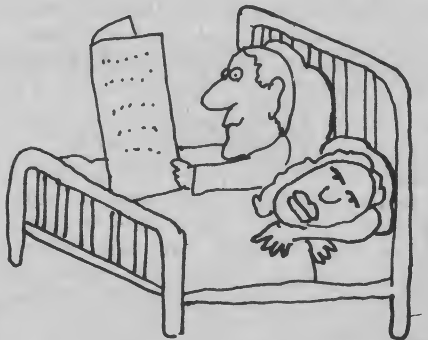


Computer Literacy
Learning and Innovation Activity.
Civil War — A tutorial program.
Test scoring by computer.
Word Scramble — A new game.
What Do You Value?

Nov-Dec 1975.

The Computer Threat to Society
Multivac — A new story by Isaac Asimov.
The computer and the rights of citizens.
Computer Crime.
Four new games and a super poster!

CREATIVE COMPUTING Reviews



Equations: The Game of Creative Mathematics, by Layman E. Allen. $6.50, WFF'N PROOF, 1490-SM South Blvd., Ann Arbor, MI 48104.

Instructional Math Play (IMP Kits: Simulations of Computer Assisted Instruction Programs, by Layman E. Allen and Jon K. Ross. $1.00 per kit, $15.00 for set of 21. WFF'N PROOF.

On-Words: The Game of Word Structures, By Layman E. Allen, Frederick L. Goodman, Doris J. Humphrey, and Joan K. Ross, $6.50, WFF 'N PROOF.

It is both convenient and natural to review these two games, and the associated instructional-simulation aids, as a single publication because they are, besides being designed by the same person or persons, very closely related in purpose, playing equipment, rules, and interest for the players.

EQUATIONS is played by two or more persons (or, as the IMP kits indicate, by a person and a computer program) with the objective of finding ways of expressing equations in simple arithmetic operations. One player defines a goal (one side of the equation) by selection of some of the numbers and operators provided by a roll of a dozen or more special dice. The players then try to come as close as possible to supplying a left-hand side, without actually doing so, by selecting one die at a time from the remaining dice. Getting too close to a solution, or preventing all solutions (by eliminating crucial dice from play) loses the game; successfully challenging an errant opponent wins.

The above description does not do full justice to the rules. Let me hasten to add that the actual rules supplied, including variants for those who find the standard rules too tame, covers forty (40) pages of printed text, so there is no way that a review can do justice to the rules. In fact, the standard rules, once understood, are not all that complex: rather, it is the presentation that is complex. This is the major problem with what are basically very interesting games: the statement of the rules is far too formal and complex.

The *IMP* kits contain a summary of the rules which is vastly easier to read and comprehend, and I would strongly recommend that anyone who buys *EQUATIONS* get some of the *IMP* kits as well. They provide solo practice as well as a clearer understanding of the rules (in Kit No. 1 only.) Once the rules have been assimilated, the game can be played by elementary school children (4th grade up) and will be enjoyed by many, I believe. The games have considerable popularity in some schools in which they are used.

My two boys (5th and 7th grades) found that they picked up new insights into arithmetic in their first attempts to play. However, they do not seem to be ready to accept the game as part of their regular selection (they are currently hung up on Cribbage.)

The play of *ON-WORDS* is similar: the goal is the length of a word, which is to be made up from a selection of the remaining cubes, which have letters on them. The general structure of the rules is identical, including, unfortunately, the complexity of the explanation. I find the game interesting and challenging, but an attempt to introduce it to a group of word-game enthusiast friends was met with furrowed eyebrows and eventual rejection. Maybe it's my poor powers of explanation, maybe they are just not ready for a game of this sophistication, but we did not get past the first game. It is a real pity that the author did not spend less time making the rules rigorous in favor of making them clear and concise.

L. D. Yarbrough
Lexington, Mass.

* * *

Learning for Tomorrow: The Role of the Future in Education, Alvin Toffler (Ed), 421 pp. $2.95. Vintage Books Div. of Random House, New York.

"All education springs from images of the future and all education creates images of the future." To support this thesis, editor Alvin Toffler (*Future Shock*) and eighteen leading psychologists, educators, futurists, social scientists, psychiatrists and humanists have joined together to put forth proposals for educational reform. With a dramatic call for "education in the future tense," they show why action learning, value clarification, racial and sexual equality, along with simulations, games, science fiction and other educational innovations need to be integrated and fused with a sense of "future – consciousness" if we are to design effective learning systems. This sense of "future – consciousness" must be developed early in the child's educational experience so that desirable futures can be planned for, and undesirable futures avoided. Teaching children to "model build", to see alternative solutions, to assume the responsibility for the implications of such alternatives, should be the primary role of education if we are to survive in a world bombarded with rapid technological innovations.

Each chapter is a self-contained unit, written by a different author on a different aspect of developing "a sense of the future". Yet, Toffler has done such a superb job of editing that the reader feels the continuity of a single authorship. The book is absorbing, developing a sense of urgency for some drastic change in our thinking on why we educate. It is also an eye opener, especially in the area of sexual and racial inequities. "Why Women See the Future Differently from Men" and "The Black Child's Image of the Future" should produce in the conscientious educator some sleepless nights.

This book is a must for all educators, instructors, and administrators alike. It provides challenging alternatives in approach to all areas of study. It has some vitally important things to say about the necessity for "real" change in our educational institutions as we encounter ever more rapid rates of technological change and accommodating changes in responding social structures.

Beginning where most proposals for education reform leave off, it demands change not merely in how, where and when we educate, but in WHY we educate.

J. Leone
THE Journal, Acton, MA

"Official" Review Policy

For the most part, reviews in *Creative Computing* come from two main sources:

1. From time to time someone writes me a letter and says, "I read such-and-such a book and would you like a review of it in the magazine." I invariably say "yes, of course," and maybe the person writes it and maybe not.

2. Lynn Yarbrough, our Reviews Editor, sends a book (or game, etc.) to one of our volunteer reviewers along with a "Guideline for Reviewers" and asks for a review by such and such a date (about 2-3 months for a book) which sometimes is met, but generally not.

Occasionally Lynn or I review a book if we feel we don't want to trust it to the vagueries of the Official Review Process. Or in fact, if we want to read the book ourselves.

Where do we get the books we review? Most publishers regard any magazine printed on newsprint a member of the Alternative Press and, therefore, Not Worthy to Receive Free Review Copies of books. Occasionally I meet a salesman or representative of a publisher at a conference and he is astounded that I am actually Human and Genuine and Serious and not a Freak even though I am publishing a magazine on (Yecch) Newsprint. And maybe they put in a good word and we get a couple of books to review (sometimes with a bill following in 10 days; in which case I usually return the book).

However, most of the books for review I actually buy because they look interesting or controversial or because I want to read them. That's why they're not all new, or strictly about computers, and include titles like *How To Survive in Your Native Land* and *Future Shock* and science fiction and adventure and other neat things.

So I guess what I'm trying to say is this. If you want to help us out, we'd love it. If you've read a good book that may be of interest to other *Creative Computing* readers, send for our review guidelines and then write a review for us. Or volunteer as a reviewer and eventually we'll send you a book to review. Or if you're a publisher (or have a publisher friend), send us a book or game for review – without a bill following. Or if you're none of the above, read our reviews and enjoy them and buy the books we recommend because you'll probably enjoy them too. –DHA

THE U.S. POSTAL SERVICE HAS LOST OVER 1/3 OF THE COMMERCIAL MAIL IT HANDLED DURING THE LAST 10 YEARS

How To Survive in Your Native Land, James Herndon, 179 pp. $1.25, Bantam Books, New York, 1971.

Contrary to the title, this book is not really a survival manual. In fact, it probably points out more pitfalls and reasons that most kids will have a hell of a time surviving in schools than it indicates solutions. The author, a junior high teacher for 10 years, found that an open approach worked for him but he's rather pessimistic whether it will be widely emulated. Indeed his own principal thinks it's maybe OK but can't really see what's wrong with Proven Establishment Methods.

I could give you all the beautiful adjectives and superlatives and reasons you should read this book whether or not you're an advocate of open education. The main reason is that there's a damn important message about the nature of schools as an institution buried in the humor and poignancy and hope and pessimism. I'll let Herndon tell you about part of it. *"In all public schools in the United States* the percentage of kids who cannot really read the social studies textbook or the science textbook or the directions in the New Math book or the explanations in the transformational grammar book is extraordinarily high. Half the kids. The school tells everyone that reading is the key to success in school, and no doubt it is, a certain kind of reading anyway. Does the school then spend time and effort teaching those kids who can't read the texts how to read the texts? Shit no, man. Why mess up a situation made to order for failure? The school's purpose is not teaching. The school's purpose is to separate sheep from goats."

Whether you're a student, teacher, or whoever – skip a day of school and read this book. You'll be better off for it.

David H. Ahl

CREATIVE COMPUTING
Feature Review
34 Books on BASIC

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

Installment #2. Continued from the Mar/Apr issue.

5. *Introduction to Computing Through The BASIC Language,* by Richard L. Nolan. Second edition pub. June 1974 (first edition pub. June 5, 1969), by Holt, Rinehart and Winston, New York, N. Y., 352 pages, 6 x 9, $9.00 (hardcover).

Some very good parts, but too disjointed, too many tangents. Rating: C

This review could have been rewritten to reflect the changes made in the recently-received second edition, but there are not many of significance, and it may be of interest to show how an author attempted to improve his text, but left the biggest fault untouched.

Although some of the changes are notable improvements, the allover effect is still the same, and so the one-line judgment and rating given for the first edition still apply to the second, except that the book might possibly rate a C+ now. The next dozen paragraphs refer to the first edition; the remaining ones delineate the changes, additions and deletions to the second.

The beginning is promising, with one of the most practical openings of all these books: the formula for calculating "the present worth of an investment for some number of years hence" is given. Then the author shows how the equivalent BASIC program line is almost the same. Four more lines are added to make the book's first program, which is then expanded upon so that several sets of constants can be used. These two programs and their explanations take up the first chapter, five pages.

But by page 13, the book begins to fall apart, with four pages that give a long table of nine BASIC definitions and twelve statements, with two or three examples of each. Too much is given in too short a space. This material should be spread out over a chapter or two, with much more text and also more examples.

Another "too much, too soon" item starts on page 19: three and a half pages of the error statements printed out by the batch-mode BASIC compiler (UWBIC) in response to 35 BASIC statements that contain one or more syntactical errors. If this is meant to show the wrong way of writing statements, there must be a better way of doing it.

Nolan goes into flowcharting early, and uses a good number of flowcharts in the text.

Page 47 starts a 69-line program, with a two-page flowchart, but there is no run to show what the program can do.

There are ten chapters: Introduction, Introduction to BASIC, BASIC Definitions, three chapters on 13 BASIC statements and the functions, Concept of a Computer (computer simulation model), Computer Hardware, Computer Software, Conclusion. There are five appendixes: time-sharing and batch-mode BASIC, techniques of flowcharting, matrices and MAT statements, additional BASIC statements (strings, computed GO TO, SGN, DEF, etc.), and some general application programs.

BASIC is covered in the first six chapters and 96 pages. There are review questions and exercises at the end of each of these chapters (and of most of the others), with full answers and solutions at the end of the book.

Chapter 4 starts with a vocabulary and dryness that do not make this an easy book to read: "The syntactical relations and grammar discussed in the previous chapter provide the basis for developing a BASIC program. In this chapter, the response elicited from the computer by the REMARK, READ, DATA, END, LET, PRINT, and GO TO statements will be explained. This will be done in the context of the logic required to . . . ." The first example of each of these statement is in words, such as "READ variable, variable . . ., variable," after which actual examples are usually given. The first program in this chapter is a slight enlargement (via REMARK statements) of the very first program, which determines present worth. Memory cells are explained with a drawing of several mailboxes. The same program is used throughout the chapter, basically unchanged, to illustrate the use of the various statements. There are 18 excellent Review Questions and Exercises, seven of them requiring programs to be written.

Chapter 5 covers IF/THEN, FOR/NEXT, DIM, and STOP. The program on page 43 is actually only the second program in the book, if one discounts the several variations on the first one. This second program seems more complex than it really is, perhaps due to the nine REMARK statements in a program that has only six active lines plus two DATA statements. Again, the language is stiff and pedagogic, with words such as "concatenated." The third program (sorting, and counting in categories) is much too long so soon, unless the author's idea is to get the reader used to long programs. The principles could be explained with one or more much shorter programs; this one is 69 lines long (20 are REMARK lines), but without a run. A grade-sorting program is so long that the flowchart takes up three pages. The chapter contains too many programs without RUNs: ten of them, and only one with an output.

Chapter 6 is on functions and GOSUB/RETURN. A "nonsensical program" illustrates four stored functions, when several shorter, meaningful ones would be much better. RND is explained in one sentence: "The function returns a six-digit, uniformly distributed pseudo-random number between zero and one." The program using RND, concerning a silver miner's "grubstake," is explained in only five short sentences, leaving many readers still largely in the dark about RND. Perhaps as an exercise for the brighter students, the chapter ends with a time-sharing simulation program that computes a historical analysis of use (average wait, longest wait), with 76 lines and a 50-box flowchart three pages long.

Chapter 7 is Concept of a Computer, subtitled Computer Simulation Model. This goes into the writing of a BASIC program that will translate "Op-codes for machine-language programming system of model computer" (in machine language), so that programs can be written directly in the "pseudo-machine-language," as required in most of the exercises accompanying this chapter. A fascinating chapter, but this elementary book is not the place for it, not right after six introductory chapters on BASIC.

Chapter 9, on Computer Software, goes into the different levels of programming languages. There are some very good sections on system programmers, application programmers, and programming systems.

Appendix B, 20 pages on Techniques of Flowcharting, includes an entire page showing a table of USASI Working Groups, X3 Series. Why? The coverage of flowcharting is very thorough, although it is nearly all taken up with symbols and their meanings, and only two flowcharts are shown.

Appendix E covers Some General Application Programs: chi-square, T-test, correlation analysis, matrix inversion, linear programming, grade analysis. Much too much in an elementary text; the linear program is almost 300 lines long.

The Selected Bibliography lists 50 books and other publications, on a variety of computer subjects, from biomedical programs to business-data programming.

The last item in the book is a 12-page glossary. The inside covers contain a synopsis of BASIC definitions, functions and statements.

The preface calls this "an integrated approach to teaching computing." Disjointed is a better word, as there are too many tangents and space-fillers. The very simple is next to the very complex, all too dizzyingly often. There are some very good parts, making this a good source book for a teacher, or perhaps as a second or third book, or for browsing. But not as a first book, either for learning about BASIC or about computers, for the solitary reader.

The second edition is essentially the same as the first, except that it is now in hardcover instead of paperback, the entire book has been reset with different typefaces, and the text reworded and expanded (from 262 to 352 pages), with two new chapters (model building, programming languages), the appendix on flowcharting turned into a chapter, the three chapters on BASIC commands reworked into four chapters on BASIC statements, and two new appendixes added.

As an example of the rewording, the first sentence of the preface in the first edition is: "Computing no longer belongs to an 'esoteric cult.'" In the second edition: "Computing is no longer the sole province of an 'esoteric cult.'" And so on, with slightly different words but the same content, in the same paragraph groupings, for the rest of the chapter.

The first chapter was previously an "introduction" and was about the advantages of using computers, with some specific applications; now it is "problem solving and algorithms" and is mainly about defining the objective, formulating a solution, and carrying it out, with several examples, such as calculating the volume of a cube, preparing Boston baked beans, and preparing a payroll.

The second chapter, on flowcharting, was previously an appendix, and this time omits the entirely superfluous chart of USASI Working Groups, X3 Series. Previously, there were mostly symbols and only two flowcharts; now there are six flowcharts, for making Boston baked beans, digging a hole, determining whether a number is prime, etc.

Chapter 3, on BASIC elements, starts almost exactly the same as chapter 2 in the first edition. The first example is on interest; the second edition inserts a few sentences showing by exactly how much the principal would increase for several years, before presenting a formula for determining interest, along with a flowchart. The same programs are presented in both editions, with basically the same text.

Chapter 4, on concepts and definitions, is almost the same as before, bewildering to the beginner, with all the definitions and statements presented at once, along with the same 35 statements containing errors.

The three chapters on BASIC "commands" are turned into four chapters, by taking GOTO from one chapter and IF-THEN from another, to make a single new chapter. INPUT is added to the first of these four chapters, along with a small drawing to illustrate a data stack.

New page 52 shows that the second edition was designed with less care than the first. It is now crowded, with a typeface that seems rather loud when compared with the simpler and more elegant type of the first edition, page 30. However, the new type is larger, and thus easier to read.

A figure has been added in the section on PRINT, to show a page divided into five print zones, each 15 columns wide.

Two pages have been added to the FOR/NEXT section, to illustrate looping in greater detail, with a program that sums five numbers, and a table to show the "contents of memory cells during execution of FOR/NEXT loop." A second illustration has been added to show legal nested loops, and there is now one to show illegal nested loops.

The coverage of DIM is increased by two pages, by rewriting and expanding the text around the same programs and flowcharts, and adding a page with a chart that shows "contents of memory cells during execution of sort program." New pages 96 to 106 run exactly parallel with old pages 62 to 72, with the same figures, and almost the same text.

The random-number function is explained much better now, instead of with only sentence; three and a half pages have been added, with a coin-flipping simulation program and flowchart, a discussion of uniform distribution, etc.

Under GOSUB/RETURN, two figures have been added to show pictorially the use of subroutines and of nested subroutines.

The time-sharing simulation program is expanded, with a longer program and more complex flowchart, although the program is not nicely indented into related groups of lines as was the old one.

The new application chapter on model building is written in a stiff and highly formal manner. Two deterministic models are presented: automobile parts economic-order-quantity (with a 13-line program) and land investment (with a 25-line program not sufficiently explained by the three sentences discussing the fairly complex program and flowchart); this last is more confusing than a helpful example would be. The stochastic simulation models involve a queuing problem (at a car wash), decision trees (introducing a product), with long programs: 77 lines for car wash, with only two sentences of explanation and no flowchart; a long, seven-page explanation of the decision-tree problem, with three trees, a long flowchart with four subroutine charts, and a 115-line program, with only a paragraph of explanation.

The chapter on the computer simulation model is expanded by "specifying in BASIC the major parts of the computer model," and lengthening the program by adding DATA statements and showing an output of the program.

The chapter on computer software adds a flowchart showing the translation process, a table giving the IBM 360/370 Operating Systems Summary, and a table with the name and type of operating system used on twelve computers from eight manufacturers, from the Burroughs B 5500 to the Xerox Sigma 5/7. Much of this chapter has been rewritten, reshuffled, and expanded. Where the first edition was about system programmers, application programmers and the computer process, the second is about language processors, operating systems, and the technical services group, and covers system and application programmers in five sentences. Some material is repeated, with changes, such as the portion on programming systems. There is a completely new section, on control programs, service programs, and on the eight types of operating systems developed "to date." There is even a note on the "unbundling of IBM."

Chapter 13, on programming languages, is an enlargement of a portion of the old chapter on software. Only one figure is from the first edition, showing the levels of programming languages. The new material consists of a page each on FORTRAN, COBOL, PL/1, BASIC, ALGOL, and RPG. However, the only examples of these are in the figure taken from the first edition, which is a program showing a simple loop operation written in binary, hexadecimal, assembly language, and four of the high-level languages. The pages on the six languages are informative as to their origins, but tell much too little about the languages themselves.

Appendix A, on matrices, is exactly the same as previously, with the addition of a footnote stating that "the introduction to matrices is based on an unpublished paper written by David L. Smith, currently a lecturer at the University of Illinois."

Appendix B, on Additional BASIC Statements, covers some of the same areas as before; two of the programs that before had no output, now have one. There are now two pages on TAB and PRINT USING, with a program that is supposed to print HI three times (but somehow prints it five times), and a page on RESTORE.

The two new appendixes are on Using Files in BASIC, with five pages on creating and using files, three programs, and five-plus pages on Using BASIC on a Mini-Computer, with an 87-line program for the PDP-8/E that simulates the landing of a lunar rocket, although without output.

The last appendix, on Package Programs, presents four of the six programs from the first edition, dropping matrix inversion and grade analysis. The linear programming example uses the same objective function, subject to the same constraints, but the program is completely different, and produces a much shorter output, half a page instead of two pages.

The old Selected Bibliography was a single list, two pages long. The new one is over three pages long, with several publications for each chapter, except for chapters 3-8, for which there is one group of five books on BASIC, by Farina, Kemeny & Kurtz, Sass, Sharpe, and Spencer.

All these changes do little to help this become a unified text, with a feeling of overall cohesiveness between its various chapters. The disjointedness persists. The biggest fault is still the sudden and overwhelming presentation of the entire BASIC repertoire of characters, definitions, statements, and error messages, after only one short program has been given, thus putting a stone wall directly in the reader's path. However, the book no longer begins to fall apart apart by page 13; because of the added material, the collapse doesn't begin until page 32.

* * *

6. *A Guide to BASIC Programming: A Time-Sharing Language*, by Donald D. Spencer. Pub. Dec. 2, 1969, by Addison-Wesley, Reading, Mass., 216 pages 6½ x 9¼, $6.95 (paperback).

One of the better books, with an easily understood text, many examples, flowcharts throughout. Rating: B+

This book has many features to recommend it, and only a few drawbacks. There are many examples of each new statement, and many flowcharts, just about one for each program. Every chapter ends in exercises, although without answers.

Spencer is the only author other than Sass (21) to provide a short history of time-sharing, starting with the 1961 CTSS at MIT. The first chapter also presents a 15-line program for finding the roots of a quadratic equation, but with no explanation other than the mechanics of putting it into the system.

The second chapter, an Introduction to BASIC, is mainly about flowcharting. The next chapter, on Elements of BASIC, does get into the language, in a slow but sure way, using only REM and LET. The fourth chapter, on Reading and Printing, contains the first complete program, six lines on determining true annual interest rate, plus five more programs.

Chapter five begins to separate the men from the boys, or rather the high-school kids from the college students, with one example that uses the summation sigma. Another example goes off on a tangent by taking up three pages to discuss in detail the Newton-Raphson method for calculating square roots. Chapter six is on loops. Chapter seven, on Arrays and Subscripted Variables, contains another digression, a confusing one on the knight's tour. This may be a practical application of tables, but in a text as elementary as this, it's too much, adding little or nothing other than confusion. The only point in including it seems to be the use of subscripted variables to indicate the path of the tour.

There is a fine chapter on matrices, 16 pages, taking the time to discuss the subject fully and carefully. Chapter 11 is Sample Programs For Study, selected from a variety of fields, with 17 problems on 45 pages, including Fibonacci numbers, coordinate geometry, greatest common divisor, compound interest, satellite orbit (two-dimensional), polynomial evaluation, generating prime numbers, maze-running, and magic-square generation. The last chapter, Problems For Reader Solution, has 23 problems, some with flowcharts (but none with solutions) including mortgage calculation, inventory turnover, number-base conversion, etc.

The section on references is unique: five pages that list 8 books on BASIC, 13 manufacturers' books on BASIC, 15 on other programming languages, 5 on programming and computers, etc.

There are five appendixes. The first is on BASIC implementations, a unique chart showing which of 96 statements are available on each of 14 different time-sharing systems. The second appendix is 7 pages on the ASR33 Teletype, followed by two pages on General Electric time-sharing commands, then a five-page glossary, and a two-page true-false quiz on BASIC.

This is one of the better books, with an easily understood text, many examples, flowcharts throughout, and many programs. The drawbacks are few: no answers to the exercises, and wandering off twice (Newton-Raphson, knight's tour). The use of the summation sigma may actually bother only a few readers who haven't gotten that far in mathematics.

The typography is distracting, as the book is not well designed. There are too many typefaces; page 33, for instance, contains five different ones, making it a very busy page, with different fonts for the text, section headings, sample program lines, an actual program, and italics for formula constants.

* * *

7. *Problem-Solving With the Computer*, by Edwin R. Sage. Pub. 1969 by Entelek, Newburyport, Mass., 244 pages, 7 x 10, $4.95 (paperback).

Very slowly paced, aimed at secondary-school students. Rating: B

The first thing one might notice in this book is the very wide margin, almost three inches, which is used 47 times alongside programs and flowcharts. If the margin were reduced to just under an inch, the book could be made almost 30% narrower.

This is a high-school text, for grades 8 through 12. It teaches by giving a problem and then discussing the solution in detail. There is a flowchart for every demonstration problem.

There are eight chapters: BASIC Skills I, BASIC Skills II, A New Look at Numbers (rounding off, primes, random numbers, etc.), Algebra and the Computer, Geometry and the Computer, Data (FOR-NEXT, subscripts), Determinants, Approximations.

Sage is one of the very few to discuss debugging with the use of PRINT statements, and also by longhand, and in detail. He is also the only one to use the phrase "fall through," and to explain this highly important principle.

There are exercises after the introduction of every new idea, but without answers. These problems are all purely mathematical, and show little imagination. The one place where some imagination is used is in introducing flowcharts, where this is done for the steps involved in making a telephone call.

Some items are covered only as "Additional Facts" following the summaries at the ends of the chapters. This includes SQR and ABS, covered in one sentence each.

In the chapter on Geometry and the Computer, many problems are examined at length and in great detail. This is fine for the student who is weak in geometry, although it would bore one who isn't.

Although the beginning of the book is extremely slow, the last chapter, on approximations, is not slow and easy at all, and will be understood only by the bright students. The approximations are for sine and cosine, natural logarithms, slope of tangent line, and limits.

* * *

8. *Introduction to Programming: A BASIC Approach*, by Van Court Hare, Jr. Pub. May 25, 1970, by Harcourt, Brace & Jovanovich, New York, N. Y., 436 pages, 6 x 9, $10.95 (hardcover).

The only book to go extensively into computer hardware (120 pages) and also FORTRAN (60 pages). Fairly well done, with many interesting features. Rating: for the entire book: B+; for the BASIC portion only: B

The book gets a higher rating for its entirety than for the BASIC portion alone because of its uniqueness as a three-part text: hardware, BASIC, and FORTRAN.

The beginning chapters, on "the history and economics of computer development, the parts of a computer system, and the way in which computers handle data," go into just about the right amount of detail: enough to be informative, not so much as to be confusing and overly technical. The author goes into much more than technical developments; he tells how, for instance, Remington Rand had the "initial lead in manufacturing large-scale machines," but lost out to IBM.

The brief section on the development of programming languages is interesting and the only one in all these books. Hare is also the only author to quote from the Bible: "But let your communication be Yea, yea, Nay, nay; for whatsoever is more than these cometh of evil" (from Matthew 5:37, part of the Sermon on the Mount).

Hare is worldly as well as colloquial: he uses *Playboy* as an example of a publication, and writes of cores as "small donuts of magnetic material." And he has some interesting comments, such as on reliability: "If our automobiles and home television sets worked one millionth as reliably as computers do, there would be no local electronic or garage mechanic who could pay his rent; they would all be out of business."

Each chapter ends with problems, without answers.

Some of the sentences in the hardware portion of the book are so terse as to be confusing to the novice, such as "The output of the computer is often in excess of printing capability, and a number of printers may be used . . ." Another sentence that could use some more explanation is "The light pen is an outgrowth of friend-or-foe radar developments." An inquiring mind might want to know a little more than just that.

Around page 100 the author begins to slide slowly into BASIC, in a chapter on Programming Essentials, without going into any details of the language. The first exposure to a BASIC program is on page 128, with a five-liner on net pay, and three pages of explanation. Although many short programs are presented in the following three dozen pages, there is not one single run, not in the entire chapter on END, PRINT, READ, and DATA, nor in the chapter on LET and stored functions. Is this to get the reader to try these short programs on a terminal?

Not until page 164 is there a program of any substance: eight lines on summing the numbers from 1 to 10. Yet by page 174 there is a 32-line program, which, although simple, might not be understood, with so little preparation up to this point, except by the brighter readers and students. The program on page 176 is also presented without adequate preparation, as are several subsequent programs.

Sorting is covered only in problems at the end of a chapter, not in the preceding text. These are not problems, actually, but rather are presentations of three sorting programs.

Seven pages are devoted to matrices, with only one program on matrix operations, in a chapter on BASIC extensions, called Adult BASIC. The definition of matrix inversion is quite murky except to an expert on the subject, or to someone who has just taken a course on it.

Although the errors in the book are minor, they seem to stand out, perhaps because most of them are so obvious, such as (on page 289), "octal 4 is equal to 010." A few pages earlier, the text is careful to point out that in the preceding program, there is a leading space before a string variable in quotes. Yet in the program itself, there is no leading space in the referenced line.

Many of the programs are interesting, but many of them have too little explanation for a good understanding. An example is the program on page 296 for right-justifying the output, with only five explanatory sentences; enough for an experienced programmer, but too little for a beginner.

The portion on BASIC ends with a chapter of selected computer problems. They are all long and complex, much too much for the little preparation so far, on computer ciphering and deciphering, dating game, mazes and labyrinths. Very nice, but too hard, and with too little explanation. The author seems to assume top students who will dig hard into the problems and figure them out as a challenge.

Starting on page 233, Hare begins to work toward FORTRAN, saying it is more flexible, in a footnote. (This book has more footnotes than many scholarly monographs: 96 of them.) Hare seems more interested in the nitty-gritty of FORTRAN than of BASIC, where he seems more interested in applications. A FORTRAN program on cross-tabulation is explained in far greater detail than any of the BASIC programs.

There is some nice detail on the importance of rounding off in affecting close decisions, such as credit being accepted or rejected. However, FORTRAN is somehow made to seem hellishly complex, which to some it may well be.

There are 18 chapters: From Loom to Electron; Bistable Devices and Binary Codes; Input/Output Devices; Memory Devices; Data and Programs in Memory; Programming Essentials; Getting the Computer to Work; END, PRINT, READ, and DATA; LET and Stored Functions; REM, GO TO, IF-THEN, and INPUT; FOR-NEXT, Subscripted Variables, and DIM; Subroutines and Their Use; A Baker's Dozen (13 problems with computer solutions); Extensions of the BASIC Language; Selected Computer Problems; Extending What You Have Learned (data format, introduction to FORTRAN); and two chapters on Thirteen FORTRAN Translations (of the BASIC programs in chapter 13).

The first appendix is unique: a side-by-side comparison of the individual features and statements of BASIC and FORTRAN, nine pages worth.

Hare has the longest glossary of all these authors: 16 pages. Nolan (5) has 11½, Sass (21) has 8, and Spencer (6) has 5 pages.

The author goes into great detail in some places, such as explaining why most systems require RUBOUT at the end of each line on paper tape when punching, and also the reason for typing TAPE before inputting tape; no other author explains these two things. Yet when it comes to the chapter involving PRINT, there are no examples of the various PRINT options. Nor is there a single printout in the chapter on LET and stored functions. The reader finds himself saying "Show me!" Thus this is an uneven book, with too much detail in many places where it isn't really necessary, and not enough in all too many instances.

The book is set in a sans-serif type. Some programs are set in boldface, others are Teletype originals in various reductions from full-size to quite small; the mixture produces many odd-looking pages.

This book may be suitable for class use where the problems will be worked, but not for reading only. The author seems to be writing on a programmer-to-programmer level.

To be continued next issue.

Notices, etc.

DON'T MISS THESE ISSUES

Sep-Oct "Computer Literacy" including lots of games for young grades (many of which don't need a computer).

Nov-Dec "Computer Threat to Society" including a new story by Isaac Asimov, "The Life and Times of Multivac."

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

BACK ISSUES AND POSTERS

We have a very limited quantity of Mar-Apr '75 issue (computer careers) available for $1.50 each prepaid. The first two issues are completely sold out.

We have Dr. Spock posters from this issue available for 30¢ or 3 stamps. Posters from the previous issues are all gone.

CAREER ARTICLE REPRINTS

Reprints of "A Computer Career for You" by Judy Edwards from the Mar-Apr 1975 issue are available for $16 per 100. Single copy 30¢ or 3 stamps.

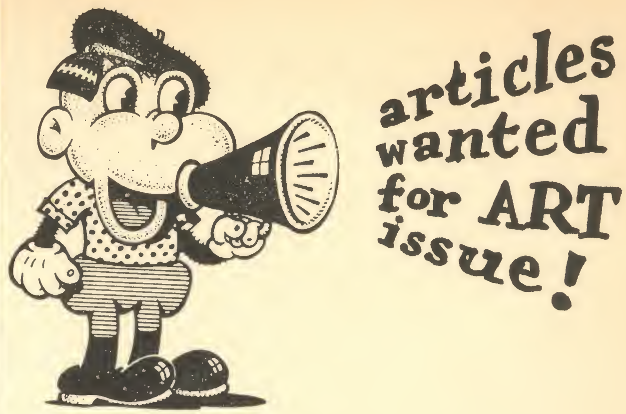
ELSEWHERE

Subscription blank on Page 4. T-Shirt Design Contest on Page 7. Programming Contest on Page 35.

WHERE ARE THEY?

"Reality Interaction Game." We promised it to you in this issue but we just didn't get it polished up in time. But it's coming — and that's a promise.

"Interview Activity." Results from the questionnaire which appeared in the Nov-Dec 1974 issue are still coming in. They will be summarized in a future issue.



Are you using the computer in art work? If so, please note that *Creative Computing* has a special art issue coming up next spring (Mar-Apr 1976). Why not contribute to this issue? Contributions should be 250 to 1500 words — or more if you have a lot to say! Typed, double-spaced. Please consider the questions below in preparing your article.

Get your material in EARLY. Absolute, final, last cut-off date is October 15, 1975 but don't wait 'till then. Also, early material has a much higher probability getting a good spot in the issue. DO IT TODAY!

How/why did you become involved with the computer (in producing art)?

What is your art background?

What role does the computer play for you . . . simulation, tool, etc.? What is your role?

Are your computer works related to non-computer art?

Do you have a final image in mind when work begins?

Could your work be done without the aid of a computer? If yes, why use the computer?

To what extent are you involved in the technical production of your work, for example, in programming?

Do you feel art work created with a computer has now or will have an impact on art as a whole in the future?

Do you intend to continue using the computer to create art pieces?

Do you recommend the use of the computer for others in creating works of art?

Along with your article, opinion, or other good words we would like illustrations, graphics, and photos of your work. Reproduction quality please (sharp B & W artwork, glossy B & W photos 5 x 7 or larger, preferably 8 x 10).

READERS AND WRITERS!! Please submit additional questions you'd like us to focus on.

Please send all material, artwork, responses, questions, etc. direct to the *Creative Computing* art issue guest editor:

Ruth Leavitt
5315 Dupont Ave. South
Minneapolis, MN 55419
(612) 825-9005

CREATIVE COMPUTING
P. O. Box 789-M
Morristown, New Jersey 07960

BULK RATE
U. S. POSTAGE
PAID
PERMIT No. 153
Morristown, NJ