Second Class Postage Paid at Menlo Park, CA
Address Correction Requested

93065JOHNSB09 D1 2251

BYRON JOHNSON MY81
356 LAGUNA TERR
SIMI VALLEY, CA 93065

$2.00 In Canada $2.50

Recreational COMPUTING

FOR THE IMAGINATIVE SMALL COMPUTER USER!

VOL. 8 NO. 5 ISSUE 44 MARCH - APRIL 1980

Editors' Notes

Things at PCC are changing. *Recreational Computing* has a new slick format; we hope you find it an improvement. There are personnel changes too. This is the last issue to be edited by the four-headed Dragon (Bob Albrecht, Tracy Deliman, Don Inman, and Ramon Zamora). Beginning next issue, Dennis Allison, long-time PCC President, will take over. The Dragonfolk will continue as contributing editors; look for them in the ComuterTown USA project, and as authors on these pages and elsewhere.

Changing Directions

Recreational Computing is a magazine of the future. Even in this age of electronic communication and computers, printed periodicals are one of the best ways to disseminate timely information . When PCC first started in 1972, *PCC Newspaper* was the only publication devoted to personal and recreational computing. Today everyone has trouble keeping up with the proliferation of computer magazines. Most magazines have followed the model we established; often they are more effective than we have been at PCC. They have had the advantage of greater resources (PCC operates as a non-profit and has always been a bootstrap operation). We don't have the subscription base or the advertising revenues of our competition. But we do have a vision of the future.

How will *Recreational Computing* change; Mostly it will be a change of focus and an expansion of coverage. We want to answer the question, "*What would I use a personal computer to do?*". We believe in the innovative and creative ability of humankind; all too often the other computer magazines (and we too, sad to say) have avoided the hard problems. Recreations and amusements will continue to be *RC*'s main thrust, but recreations will be interpreted very broadly. We will try to show how computers can be used in an effective and innovative way to improve the quality of life. We view the computer as a tool, the key to maintaining individuality in an increasingly technological society. We want to provide windows into the technology to allow everyone access to the tool. A personal computer should be more than an expensive paperweight.

You Can Help

We need your ideas. We need your articles, literate articles on all aspects of computing and computer recreations. While we at PCC will be developing some of the material published in *Recreational Computing* it is you, our readership, which will provide the most interesting and thought-provoking articles. We need your serious articles and your off-the-wall comments.

Even if you are not a potential author, we still want your input. What kind of articles should we be publishing? What is important now and what will be important in the future? You are our biggest resource.

We also need your financial support. To make *Recreational Computing* a continuing success we must have a larger subscription base. Urge

Continued on pg. 43

Recreational COMPUTING

March/April 1980
Volume 8, Issue 5

SPECIAL FEATURES

GAMES

SOFTWARE

DEPARTMENTS

Cover by Sahnta Pannuti

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

FOREIGN DISTRIBUTORS OF *RECREATIONAL COMPUTING*: **LP Enterprises**, 8/11 Cambridge House, Cambridge Rd., Barking Essex 1G11 8NT, UK; **Electronic Concepts PTY Ltd.**, 52-58 Clarence St., Sydney, NSW 2000, AUSTRALIA; **RS-232**, 186 Queen St., W. Toronto, Ontario M5V1Z1, CANADA; **ASCII Publishing**, 305 HI TORIO, 5-6-7 Minami Aoyama, Minato-Ku, Tokyo 107, JAPAN; **Ing. W. Hofacker**, D815 Holzkirchen, Lindenstr. 8, WEST GERMANY; **Hobby Data**, S200 12 Malmo, SWEDEN; **Computer Store**, P.O. Box 31-261, 22B Milford Rd., Milford, Auckland, NEW ZEALAND; **Systems Formulate Corp.**, Shin-Makicho Bldg. 1-8-17, Yaesu, Chuo-Ky, Tokyo 103, JAPAN.

Recreational Apple II Hi-Res Graphics

Lines, Triangles, and Other Shapes

BY ARTHUR WELLS, JR.

Arthur Wells sends a steady flow of great programs for the Apple. Here is his latest offering--an easy-to-use shape generator and Hi-Res graphics package. Arthur encourages everyone to use the program and talk with him about it. Write him at 1171 Cragmont Ave., Berkeley, CA 94708.—RZ

The program permits anyone to draw simple or elaborate pictures on the 280 X 160 dot high-resolution screen available with the Apple II. The user never has to know or learn anything about programming, grids or coordinates. The program lets the user draw dots, lines, solid rectangles, triangles and parallelograms of any size, any place on the screen. Unwanted lines and shapes can be erased. Color can be selected. Cursor coordinates can be identified if desired. Pictures can be saved and recalled, and a recalled picture can be modified.

THE APPLE II

Several features of the Apple II permit this program to operate with ease. The first, of course, is that the computer produces high resolution graphics. The Apple sets aside a specific area of memory which, with proper calls to hardware functions, produces a 280 X 160 grid for graphics display. Second, the Apple comes with a Hi-Res demonstration program. This program contains machine language software that will produce dots and lines, as well as perform other necessary graphics chores. Lastly, the Apple provides a set of two "paddles"—analog devices that let you change the input to the computer by rotating the dials on the paddles. Using the paddles for movement and a flashing dot as a cursor, the user can put a display on the screen without arithmetic or mathematics.

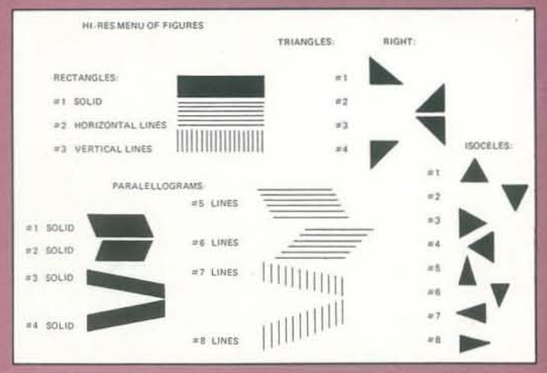
The Hi-Res display mode permits the use of up to four lines of text at the bottom of the screen. One serious fault of the graphics mode is that alpha-numeric characters cannot be displayed on the Hi-Res screen. This program corrects that defect by incorporating a routine which, upon request, goes to a HEX table, gets the desired letters and numbers, and puts them wherever the user wants them on the screen.

GETTING THE PROGRAM UP

This program works best if you have a disc drive. If you do, load the Hi-Res machine language subroutines at C00 to FFF. If you do not care about printing on the Hi-Res screen, save the subroutines on the same disc as the BASIC program with the command "BSAVE HIRES.SKETCH.BIN, A$C00, L$400". Line 110 of the program will automatically load the subroutines each time you RUN.

If you want to print on the screen you must load the Hi-Res subroutines, then go into the monitor and enter the HEX table set out in this article, starting at address F00. (Yes, this will overwrite some of the subroutines.) Then save the subroutines and the table with the command "BSAVE HIRES . SKETCH . BIN , A$C00 , L$6B0". Load the BASIC program and RUN.

If you have only tape, go to the monitor and load the subroutines C00 . FFF. If you use the HEX table, the space required is C00 . 12B0. Then go to BASIC and load the BASIC program. Delete line 110. If you only have 16K of memory, you must set HIMEM:8192.



WORKING THE PROGRAM

Dots and Lines: After you type RUN, the name of the program and a little blurb will appear. Hit any key. The screen will go black. Push the button on paddle 0. You will see a blinking dot as long as you hold the button down. Paddle 0 moves the dot up and down; paddle 1 moves it right and left. Find the place where you want to start a line. Then push "D". This fixes the point from which a line will be drawn.

Nothing further will happen until you push the Space Bar. You must *always* push the Space Bar after each time you fix a point or draw something. (It will be easy to remember this. Every time you can't get anything in the program to work, it means you forgot to push the Space Bar.)

Push the Space Bar. The dot on the screen may disappear. This is because the blinking dot subroutine erased it. However, the location of the dot is still POSNed as the start of the line even if the dot is not visible. If you want the dot to remain visible, move one of the paddles a little before you push the Space Bar.

Now move the blinking dot to another location where you wish the end of the line to be. Push "L". A line should appear from the first dot location to the present blinking dot location. Move the blinking dot again and draw another line by pushing "L" again. (*Remember the Space Bar.*) Notice that the line will start from the location where you POSNed a dot with "D" and not from the end of the last line you drew. Thus if you want to draw a line from the end of the last line you drew, you must re-POSN the dot by using "D". The drill would be "D", Space Bar, move the dot, "L", Space Bar, "D", Space Bar, move the dot, "L", Space Bar, and so on.

Now make another line. Right after you make the line by pushing "L", hit the Space Bar, and then push "E". This will erase the line. So the "E" subroutine is the opposite of the "L" subroutine: it erases a line from the last dot POSNed by "D" to the present position of the dot. To erase everything and start all over, push #.

Lastly, if you want to make a record, or for any other reason you want to know the coordinates of the location of the blinking dot, push the button on paddle 1. The coordinate will appear as long as the button is depressed.

Shapes: This article contains a menu of shapes the program will draw. (The shapes are my choice and you could put other shapes in the program instead.) To call up a shape, position the blinking dot where you want the shape to start. Then hit "R" if you want a rectangle, "P" if you want a parallelogram, or "T" if you want a triangle. The program will ask you for dimensions of a rectangle or parallelogram (enter horizontal dimension, comma, vertical dimension) or size of a triangle (enter one number). Because the program takes this information as input, you must enter it by pushing "RETURN". This is the only time "RETURN" is used. After sizing the shape, you must select the specific shape by number

upon request of the program. The triangle subroutine will also ask if you want a right or isoceles triangle.

Remember that after each shape is printed you must push the Space Bar to continue.

You can erase any figure, or a part of any figure, by using the button on paddle 1. As long as the button is depressed, the shape will erase when it would otherwise draw. To erase the whole shape, depress the button before you push the number of the shape you want and hold it down. For example, to erase a triangle, position the dot, push "T", enter size, select "R" or "I", push the button on paddle 1, and select the number of the triangle.

You can choose the color of the shape: after Space Bar push "C", then push "B" for blue or "G" for green, call the shape and proceed. The color reverts to white after each shape is drawn.

There is a block erase subroutine which will erase without regard to the position of the blinking dot. To see how this works, fill the screen with a rectangle (by moving the dot to (0,0), call Rectangle and use dimensions (279, 159), then push "M", and enter coordinates as requested.

You can combine the draw and erase functions to get all sorts of odd shapes. Print a triangle, then go over it with rectangle #2 or #3 in the erase mode. This will give you a triangle of lines instead of a solid one. By using the erase function creatively, you can cut out almost any portion of any shape. A little planning helps.

Alphanumerics: To get characters on the Hi-Res screen push "W", then enter what you want printed, push RETURN, enter coordinates where the characters are to be printed, and push RETURN again. Space Bar gets you back to the rest of the program. I changed several of the characters: "!" will print a multiplication sign "X"; the quote sign will print a division sign ":"; "&" will print a degree sign "°". These might be useful for creating math problems for children.

SAVING THE DISPLAY

If you have a disc you can easily save and recall pictures you are working on or have finished.

If you want to save a picture, push "H". Then name the picture and push RETURN. The program will save the picture and tell you when it's through. Push Space Bar to continue.

If you want to bring back a picture for more work or for any other reason, push "S" and name the picture. Then push RETURN and the picture will appear. Push Space Bar to continue.

The picture is moved and loaded at $2000, the primary high resolution page. If the program blows up in the middle of your drawing, you can save what you have with the command "BSAVE [name], A$2000, L$2000". If you must save to tape, go to the monitor and save the picture with 2000.3FFFW.

EXPANDING THE PROGRAM

This program is by no means complete. There are several things that can be done to improve it.

1. I wrote this program in Integer BASIC before disc drives and Applesoft ROM cards were available. If the program were in Applesoft we could
   a) prevent the program from crashing under most predictable circumstances by using ONERR GOTO, a command unavailable in Integer BASIC, and
   b) have the program draw circles and curves, because Applesoft has floating point arithmetic. Someone should definitely do this.
2. There are already programs which permit the user to rotate the Hi-Res graphics page up and down, left and right, and which cause the picture to become a negative of itself (white to black, black to white). Another program allows you to merge two pictures, if you have 48K of memory. These programs could be incorporated into this one. (I did not write these programs, but I will see that they are published herein if there is any interest as I know the author will be happy to share them.) People can probably think of many more ways to improve and expand this program. I would be delighted to see some attempts and their results.

Listing

```
 1 GOSUB 32500
 2 DIM D$(1),PIC$(25),B$(36)
 3 PRINT "NOMONC,I,O"
 4 TEXT
 5 CALL -936
 6 D$="": REM  CONTROL D
10 VTAB 8
20 TAB 14
30 PRINT "HIRES SKETCH"
40 VTAB 12
50 TAB 12
60 PRINT "COPYRIGHT 1978 BY"
70 TAB 12
80 PRINT "ARTHUR WELLS, JR."
100 FOR T=1 TO 1500: NEXT T
110 PRINT D$;"BLOADHIRES.SKETCH.BIN"
115 CALL -936
120 VTAB 7
130 PRINT "DOCUMENTATION FOR THIS PROGRAM CONSISTS OF SEVERAL PAGES OF INSTRUCTIONS AND A"
140 PRINT "MENU OF SHAPES AS WELL AS INSTRUCTIONS  FOR SAVING AND RECALLING PICTURES."
150 PRINT
160 PRINT "THE DOCUMENTATION IS AVAILABLE FROM THE APPLE CORE, P.O. BOX 4816, SAN FRANCISCOC
    RNIA, 94101."
170 PRINT
180 PRINT "OR WRITE TO ME DIRECTLY AT 428-13TH     STREET, OAKLAND, CALIFORNIA, 94612."
190 PRINT : PRINT "HIT ANY KEY TO START."
195 X= PEEK (-16384): IF X>127 THEN 199: GOTO 195
199 POKE -16368,0
200 P=19999
201 POKE -16368,0
```

```
205 L=29999
210 UNP=21000
215 PN=25000
220 READ=32000
225 DRAW=31000
227 PT=26000
229 Q=255
230 CALL INIT
240 CALL -912: CALL -912: CALL -912
250 GOSUB READ
260 D= PEEK (-16384): IF D=196 THEN GOTO PT
261 IF D=195 THEN GOSUB 31400
262 IF D=205 THEN 31200
263 IF D=200 THEN GOSUB 401
264 IF D=211 THEN GOSUB 500
265 IF D=204 THEN GOSUB 31000: IF D=204 THEN GOTO 275
266 IF D=197 THEN GOSUB 31100: IF D=197 THEN 275
267 IF D=163 THEN 230
268 IF D=210 THEN 10000
269 IF D=212 THEN 11000
270 IF D=208 THEN 13999
271 IF D=215 THEN 600
274 GOTO 250
275 POKE -16384,0
276 Q=255
280 IF PEEK (-16384)=160 THEN 300
290 GOTO 280
300 POKE -16368,0: GOTO 250
400 END
401 POKE -16368,0
405 D$="": REM  CONTROL D
410 PRINT "NAME OF PICTURE YOU WANT SAVED:"
420 INPUT PIC$
430 PRINT D$;"BSAVE";PIC$;",A$2000,L$2000"
440 PRINT "YOUR PICTURE IS SAVED."
450 GOTO 280
499 END
500 POKE -16368,0
510 D$="": REM  CONTROL D
520 INPUT "NAME OF PICTURE YOU WANT:",PIC$
540 PRINT D$;"BLOAD";PIC$;",A$2000"
560 GOTO 280
599 END
600 POKE -16368,0
610 POKE 28,255: POKE 812,255: POKE 806,1: POKE 807,0
620 INPUT "PRINT:",B$
630 INPUT "WHERE",X1,Y1
640 GOSUB 700: GOTO 280
700 X=X1:Y=Y1
710 FOR L=1 TO LEN(B$)
720 K= ASC(B$(L,L))
730 POKE 800,X MOD 256: POKE 801,X>255: POKE 802,Y: CALL POSN
740 M=(K-160)*15:N=M/256+15: POKE 804,M MOD 256: POKE 805,N: CALL SHAPE
750 X=X+7
760 NEXT L: RETURN
10000 POKE -16368,0:Q= PDL (0):I= PDL (1)
10005 INPUT "DIMENSIONS",L,W: PRINT "WHICH RECTANGLE?"
10010 A= PEEK (-16384)
10020 IF A>179 OR A<177 THEN 10010: GOTO (A-176)*100+10000
10100 FOR Z=I TO I+W
10120 X1=0:Y1=Z:X=0+L:Y=Z
10125 A= PEEK (-16286): IF A>127 THEN GOSUB 31100: IF A>127 THEN 10140
10130 GOSUB DRAW
10140 NEXT Z
10150 GOTO 275
10200 FOR Z=I TO I+W STEP 2:X1=0:Y1=Z:X=0+L:Y=Z
10210 A= PEEK (-16286): IF A>127 THEN GOSUB 31100: IF A>127 THEN 10230
10220 GOSUB DRAW
10230 NEXT Z: GOTO 275
10300 FOR Z=0 TO 0+L STEP 3
10305 X1=Z:Y1=I:X=Z:Y=I+W
10310 A= PEEK (-16286): IF A>127 THEN GOSUB 31100: IF A>127 THEN 10330
10320 GOSUB DRAW
10330 NEXT Z: GOTO 275
11000 POKE -16368,0:Q= PDL (0):I= PDL (1)
11010 INPUT "SIZE",S
11020 PRINT "RIGHT OR ISOCELES?"
11030 A= PEEK (-16384)
11040 IF A=210 THEN 11060: IF A=201 THEN 11080: GOTO 11030
11050 IF A=197 THEN 13100
11060 POKE -16368,0: PRINT "WHICH TRIANGLE?"
11065 A= PEEK (-16384)
11070 IF A>180 OR A<177 THEN 11065
11075 GOTO 11100
11080 POKE -16368,0: PRINT "WHICH TRIANGLE?"
11085 A= PEEK (-16384)
11090 IF A>184 OR A<177 THEN 11085
11095 GOTO 11300
11100 FOR Z=1 TO S
11110 GOSUB (A-176)*100+16000
11235 B= PEEK (-16286): IF B>127 THEN GOSUB 31100: IF B>127 THEN 11250
11240 GOSUB DRAW
11250 NEXT Z
11260 GOTO 275
11300 FOR Z=1 TO S
11310 GOSUB (A-176)*100+17000
11320 B= PEEK (-16286): IF B>127 THEN GOSUB 31100: IF B>127 THEN 11340
11330 GOSUB DRAW
11340 NEXT Z: GOTO 275
13999 POKE -16368,0:Q= PDL (0):I= PDL (1)
14000 INPUT "DIMENSIONS",L,W: PRINT "WHICH PARALLELOGRAM?"
14010 A= PEEK (-16384)
14020 IF A>184 OR A<177 THEN 14010: GOTO (A-176)*100+14010
14110 FOR Z=0 TO W:X1=0+Z:Y1=I+Z:X=0+Z+L:Y=I+Z
14115 A= PEEK (-16286): IF A>127 THEN GOSUB 31100: IF A>127 THEN 14130
14120 GOSUB DRAW
14130 NEXT Z: GOTO 275
14210 FOR Z=0 TO W:X1=0-Z:Y1=I+Z:X=0-Z-L:Y=I+Z
14215 A= PEEK (-16286): IF A>127 THEN GOSUB 31100: IF A>127 THEN 14230
14220 GOSUB DRAW
14230 NEXT Z: GOTO 275
14310 FOR Z=0 TO L:X1=0+Z:Y1=I+Z:X=0+Z:Y=I+Z+W
14315 A= PEEK (-16286): IF A>127 THEN GOSUB 31100: IF A>127 THEN 14330
14320 GOSUB DRAW
14330 NEXT Z: GOTO 275
14410 FOR Z=0 TO L:X1=0-Z:Y1=I+Z:X=0-Z:Y=I+Z+W
14415 A= PEEK (-16286): IF A>127 THEN GOSUB 31100: IF A>127 THEN 14430
14420 GOSUB DRAW
14430 NEXT Z: GOTO 275
14510 FOR Z=0 TO W STEP 2:X1=0+Z:Y1=I+Z:X=0+Z+L:Y=I+Z
14515 A= PEEK (-16286): IF A>127 THEN GOSUB 31100: IF A>127 THEN 14530
14520 GOSUB DRAW
14530 NEXT Z: GOTO 275
14610 FOR Z=0 TO W STEP 2:X1=0-Z:Y1=I+Z:X=0-Z-L:Y=I+Z
14615 A= PEEK (-16286): IF A>127 THEN GOSUB 31100: IF A>127 THEN 14630
14620 GOSUB DRAW
14630 NEXT Z: GOTO 275
14710 FOR Z=0 TO L STEP 3:X1=0+Z:Y1=I+Z:X=0+Z:Y=I+Z+W
14715 A= PEEK (-16286): IF A>127 THEN GOSUB 31100: IF A>127 THEN 14730
14720 GOSUB DRAW
14730 NEXT Z: GOTO 275
14810 FOR Z=0 TO L STEP 3:X1=0-Z:Y1=I+Z:X=0-Z:Y=I+Z+W
```

Hi-Res On Screen Characters

```
14815 A= PEEK (-16286): IF A>127 THEN GOSUB 31100: IF A>127 THEN 14830
14820 GOSUB DRAW
14830 NEXT Z: GOTO 275
16100 X1=O:Y1=I+Z:X=O+Z:Y=I+Z
16101 RETURN
16200 X1=O:Y1=I+Z:X=O-Z:Y=I+Z
16201 RETURN
16300 X1=O:Y1=I-Z:X=O-Z:Y=I-Z
16301 RETURN
16400 X1=O:Y1=I-Z:X=O+Z:Y=I-Z
16401 RETURN
17100 X1=O-Z:Y1=I+Z:X=O+Z:Y=I+Z
17101 RETURN
17200 X1=O-Z:Y1=I-Z:X=O+Z:Y=I-Z
17201 RETURN
17300 X1=O+Z:Y1=I-Z:X=O+Z:Y=I+Z
17301 RETURN
17400 X1=O-Z:Y1=I-Z:X=O-Z:Y=I+Z
17500 X1=O-Z/2:Y1=I+Z:X=O+Z/2:Y=I+Z
17501 RETURN
17600 X1=O-Z/2:Y1=I-Z:X=O+Z/2:Y=I-Z
17601 RETURN
17700 X1=O+Z:Y1=I-Z/2:X=O+Z:Y=I+Z/2
17701 RETURN
17800 X1=O-Z:Y1=I-Z/2:X=O-Z:Y=I+Z/2
17801 RETURN
19999 POKE 812,255
20000 POKE 800,X MOD 256: POKE 801,X>256: POKE 802,Y: CALL PLOT
20001 RETURN
21000 POKE 812,0: GOTO 20000
25000 X1=X:Y1=Y
25010 POKE 800,X1 MOD 256: POKE 801,X1>256: POKE 802,Y1
25020 CALL POSN
25030 RETURN
26000 GOSUB 19999
26005 GOSUB 25000
26010 POKE -16368,0
26020 GOTO 280
29999 POKE 812,255
30000 POKE 800,X MOD 256: POKE 801,X>256: POKE 802,Y: CALL LINE
30001 RETURN
31000 POKE 812,0
31001 POKE 800,X1 MOD 256: POKE 801,X1>256: POKE 802,Y1
31002 CALL POSN
31010 POKE 800,X MOD 256: POKE 801,X>256: POKE 802,Y
31020 CALL LINE
31040 RETURN
31100 POKE 812,0: GOTO 31001
31200 POKE -16368,0: INPUT "HORIZONTAL COORINATES",H1,H2
31210 INPUT "VERTICAL COORDINATES",V1,V2
31220 FOR Z=0 TO V2-V1:X1=H1:Y1=V1+Z:X=H2:Y=V1+Z
31230 GOSUB 31100: NEXT Z: GOTO 275
31400 POKE -16368,0
31410 A= PEEK (-16384): IF A=194 THEN Q=85: IF A=194 THEN RETURN
31420 IF A=199 THEN Q=170: IF A=199 THEN RETURN
31430 IF A=215 THEN Q=255: IF A=215 THEN RETURN
31440 GOTO 31410
32000 X= PDL (0):Y= PDL (1): IF PDL (1)>159 THEN Y=159
32010 L= PEEK (-16286): IF L>127 THEN PRINT "X=";X,"Y=";Y
32015 IF L<128 THEN PRINT "                              "
32020 K= PEEK (-16287): IF K>127 THEN GOSUB P: GOSUB UNP
32030 RETURN
32500 INIT=3072:CLEAR=3086:POSN=3761:PLOT=3780:LINE=3786:SHAPE=3805
32505 RETURN
```

```
0EFF- FF
0F00- 49 92 00 00 00 00 00 00
0F08- 00 FF FF FF FF FF FF 0A
0F10- 0E 0E 0E 0E DE 1B 28 28
0F18- 28 28 28 00 FF FF 49 72
0F20- 11 3F 3F 4E 32 00 00 00
0F28- FF FF FF FF FF FF FF FF
0F30- FF FF FF FF FF FF FF FF
0F38- FF FF FF FF 09 0D 3A 3F
0F40- 1E 0E 2D 0E 1E 3F 37 09
0F48- 05 00 FF 2E 6C 11 17 17
0F50- 17 17 4D 35 3F 00 FF FF
0F58- FF FF 31 25 05 00 FF FF
0F60- FF FF FF FF FF FF FF FF
0F68- FF 09 36 06 00 FF FF FF
0F70- FF FF FF FF FF FF FF FF
0F78- 49 1E 1E 36 15 15 05 00
0F80- FF FF FF FF FF FF FF 49
0F88- 0E 0E 36 17 17 07 00 DF
0F90- FF FF FF FF FF FF 09 0D
0F98- 3A 3B 73 2D 1E 17 2D 15
0FA0- 1F 1F 0E 29 00 89 FE 2A
0FA8- 2D F5 33 06 00 00 FF DF
0FB0- FF FF FF FF 09 92 32 17
0FB8- 07 00 FF FF FF FF FF DF
0FC0- FF FF FF 92 29 2D 2D 00
0FC8- FF FF FF FF FF FF FF DF
0FD0- FF FF 09 92 92 06 00 FF
0FD8- FF FF FF FF FF FF FF DF
0FE0- FF 49 11 1E 1E 1E 1E 05
0FE8- 00 FF FF FF FF FF FF DF
0FF0- 92 21 2C 28 75 36 36 17
0FF8- 3F 1C 2C 0C 0C 04 00 4A
1000- 0C 36 36 BE 2D 04 00 00
1008- 00 00 00 00 00 00 2A 28
1010- 75 BE F7 1E 2E 2D 25 00
1018- 00 00 00 00 00 29 2D 35
1020- 17 17 75 3E 3A E7 04 00
1028- 00 00 00 00 49 31 36 2E
1030- D6 24 3F 27 0C 0C 04 00
1038- 00 00 00 0A 2C 2D 35 1A
1040- 3F 37 12 15 2D 0C 24 04
1048- 00 05 49 2D 96 32 1E 3F
1050- 38 24 2C B0 2D 05 00 00
1058- 00 2D 2D 3E 3A 3A 32 36
1060- 00 00 00 00 00 00 00 00
1068- 09 2D 15 B6 BE 3F 1C 24
1070- 20 95 2D 04 00 00 00 52
1078- 24 29 AD 36 BE 17 3F 04
1080- 08 28 2D 00 00 00 09 12
1088- 16 06 00 00 00 00 00 00
1090- 00 00 00 00 00 09 12 16
1098- 3E 3A 00 00 00 00 00 00
10A0- 00 00 00 00 00 00 00 00
10A8- 00 00 00 00 00 00 00 00
10B0- 00 00 00 52 2D 2D 16 3F
10B8- 3F 07 00 00 00 00 00 00
10C0- 00 00 00 00 00 00 00 00
10C8- 00 00 00 00 00 00 00 00
10D0- 00 2A 28 75 1E 1E B6 06
10D8- 00 00 00 00 00 00 00 00
10E0- 00 00 00 00 00 00 00 00
10E8- 00 00 00 00 00 00 00 11
10F0- 1E 36 F5 6E 09 FC 28 25
10F8- 34 38 60 1B 07 00 36 36
1100- 36 2D 25 21 04 30 20 1C
1108- 3F 96 2D 05 00 29 75 96
1110- BA 3F 1C 24 24 04 00 00
1118- 00 00 00 00 2D 75 36 36
1120- 17 3F 27 24 24 05 00 00
1128- 00 00 00 2D 2D 96 92 3F
1130- 3F 24 24 AC 2A 05 05 00
1138- 00 00 2D 2D 96 3B B7 1A
1140- 24 24 24 05 00 00 00 00
1148- 00 29 2D DE 1B 36 36 15
1150- 2D 25 3C 07 00 00 00 00
1158- 36 36 36 4D 21 24 24 24
1160- 93 12 3F 07 00 00 00 29
1168- F5 36 36 15 3F 07 00 00
1170- 00 00 00 00 00 00 49 31
1178- 36 36 17 3F 1C 04 00 00
1180- 00 00 00 00 00 36 36 36
1188- 4D E1 1C 1C 0C 0C 0C 04
1190- 00 00 00 00 36 36 36 2D
1198- 2D 05 00 00 00 00 00 00
11A0- 00 00 00 36 36 36 4D 21
11A8- 24 24 3C B2 23 1C 04 00
11B0- 04 00 36 36 36 4D 21 24
11B8- 24 3C 92 3A 38 20 00 00
11C0- 00 29 75 36 36 17 3F 1C
11C8- 24 24 04 00 00 00 00 00
11D0- 2D 75 BE 3F C4 36 36 36
11D8- 00 FF FF FF FF FF FF 29
11E0- 75 36 36 3A E0 16 3F 20
11E8- 24 24 00 FF FF FF 2D 75
11F0- BE 3F C4 36 36 2E 49 1C
11F8- 1C 07 00 FF FF 29 75 1E
1200- 2A 32 17 3F 07 28 08 38
1208- 1C 24 00 FF 2D 2D DE 36
1210- 36 36 00 FF FF FF FF FF
1218- FF FF FF 36 36 AE 2D 0C
1220- 24 24 24 00 FF FF FF FF
1228- FF FF 36 36 15 15 05 28
1230- 20 24 24 00 00 FF FF FF
1238- FF 36 36 36 05 28 20 15
1240- 32 21 24 24 24 00 FF FF
1248- 2E AA 17 17 2E 09 21 07
1250- 20 28 20 04 00 FF FF AE
1258- 96 0A 24 2C 28 20 04 00
1260- FF FF FF FF FF FF 2D 2D
1268- BE 17 17 17 2E 2D 2D 00
1270- FF FF FF FF FF FF FF FF
1278- FF FF FF FF FF FF FF FF
1280- FF FF FF FF FF FF FF FF
1288- FF FF FF FF FF FF FF FF
1290- FF FF FF FF FF FF FF FF
1298- FF FF FF FF FF FF FF FF
12A0- FF FF 89 AA 1F 17 4D 31
12A8- 00 FF A5 8C A4 8D 85 AB
```


GAMES CATALOGUE

Some of the most popular games of this season and last, for TRS-80, Apple and PET, are listed here. The vendor's name and address and prices are included. Those highlighted with an asterisk have received favorable reviews by Jim Barter, Mark Cross and David Stone, using a game review plan developed by Marin Computer Center in San Rafael. Favorable means that these programs generally rate high in overall value, make good use of the computer's special features, have clear instructions, are creative, interesting or challenging. These lists and reviews are excerpted from Robert Elliot Purser's Software Directory, Winter 1980 (Post Box 466, El Dorado, CA 95623). —TD

TRS-80 SOFTWARE

Adventure International
Scott Adams
Box 3435
Longwood, FL 32750
(305) 862-6917

*TRS2 Adventure 1 "ADVENTURELAND"	$ 14.95
*TRS2 Adventure 2 "PIRATES ADVENTURE"	14.95
*TRS2 Adventure 3 "MISSION IMPOSSIBLE"	14.95
*TRS2 Adventure 4 "VOODOO CASTLE"	14.95
*TRS2 Adventure 5 "THE COUNT"	14.95
*TRS2 Adventure 6 "STRANGE ODYSSEY"	14.95
*TRS2 Adventure 7 "MYSTERY FUN HOUSE"	14.95
*TRS2 Adventure 8 "PYRAMID OF DOOM"	14.95
*TRS2 Galactic Empire	7.95
TRS2 Backgammon	7.95
TRSD Any 1 Adventure	19.95
TRSD Any 2 Adventures	24.95
TRSD Any 3 Adventures	39.95

(Tape versions require 16K, Disk 32K.)

****

Automated Simulations
P.O. Box 4232
Mountain View, CA 94040

TRS2 Starfleet Orion	$ 19.95
*TRS2 Invasion Orion	19.95
TRS2 Temple of Apshai	24.95
TRSD Temple of Apshai	24.95
TRS2 The Datestones of Ryn	14.95
TRSD The Datestones of Ryn	14.95

Creative Computing Software
P.O. Box 789-M
Morristown, NJ 07960
(800) 631-8112
In NJ (201) 540-0445

TRS2 Board Games-1 (six programs)	$ 7.95
TRS2 Space Games-3 (four programs)	7.95
TRS2 Ecology Simulations-1 (four programs)	24.95
TRS2 Air Traffic Controller	7.95
TRS2 Strategy Games	7.95
TRS2 Pursuit Games	7.95
TRS2 Graphic Package	7.95

(Adventure programs from Scott Adams; also disk versions of same programs available)

****

Cybernautics
Art Confil
Box 40132
San Francisco, CA 94140

TRS2 TAIPAN The China Trade of the 1800's	$ 9.98
TRS2 TREK 78 Spacetrek with graphics	9.98
TRS2 1000 MILES Roadrace game vs. TRS-80	11.98
TRS2 GAMESETTE 1 Awari, Concentration, etc.	11.98

****

Instant Software
Peterborough, NH 03458
(603) 924-7296

TRS2 Basic/Intermediate Lunar Lander (0001) $	7.95
TRS2 Space Trek II (0002)	7.95
TRS2 Robot Chase/Horse Race/Knight Quest(0003)	7.95
TRS2 Beginner's Backgammon/Keno (004)	7.95
TRS2 Golf/Cross-Out (009)	7.95
TRS2 Car Race/Rat Trap/Anti-Aircraft (0011)	7.95
TRS2 Cave Exploring/Yacht Memory (0010)	7.95
* TRS2 Destroy All Subs/Gunboats/Bomber(0021)	7.95
TRS2 Oil Tycoon (0023)	7.95
TRS2 Demo I (0020)	7.95
TRS2 Bowling (0033)	9.95
TRS2 Ramron Patrol/Tie Fighter/Klingon Capture	7.95
TRS2 Doodles and Displays I (0030)	7.95
TRS2 Space Trek III (0031)	7.95
TRS2 Space Trek IV (0034)	7.95
TRS2 Fun Package I (0037)	7.95
TRS2 Doodles Display II (0043)	7.95
* TRS2 Santa Paravia Fiumoccio (0043)	7.95

****

Programma International, Inc.
3400 Wilshire Boulevard
Los Angeles, CA 90010
(213) 384-0579

TRS2 COMPUTER MINDREADER 16K	$ 6.95
TRS2 CONCENTRATION 16K	6.95
TRSD DBM/5 32K	49.95
TRSD DIR 32K	15.95
TRS2 DOOBLE 16K	6.95
TRS2 D.O.F. 16K	9.95
TRS2 FIFTEEN NUMBERS 16K	6.95
TRS2 FIVE SPEED RACER 16K	6.95
TRSD HUNT 32K	24.95
TRS2 HUNT 16K	19.95
TRS2 MAZE-80 16K	9.95
TRS2 MINEFIELD 16K	6.95
TRS2 NIM 16K	6.95
TRS2 PACHINKO 16K	6.95
* TRS2 PENTOMINOES 16K	9.95
TRS2 PIE text editor 16K	19.95
TRS2 QUIZMACHINE 16K	6.95
TRS2 ROULETTE 16K	6.95
TRS2 TANK 16K	6.95
TRS2 THE I CHING THING 16K	6.95
TRS2 TRIBBLE TRAP II 16K	6.95

(Prices do not include postage)

****

The Software Stockpile
9434 Ironwood
Des Plains, IL 60016

TRS2 Gone Fishing	$ 6.00
TRS2 Concentration	6.00
TRS2 Craps	6.00
TRS2 Slot Machine	6.00
TRS2 Starship	6.00
TRS2 Sherlock Holmes	6.00
TRS2 Number Guess	6.00
TRS2 Dice Roll	6.00
TRS2 In-Between	6.00
TRS2 Shell Game	6.00
TRS2 Safari	6.00
TRS2 Starship 2	6.00
TRS2 Mortar Battle	6.00
TRS2 Puzzle	6.00
TRS2 Teasers	6.00
TRS2 Mouse	6.00
TRS2 PT Boat	6.00
TRS2 Turtle Race	6.00
TRS2 Chek-Chess	6.00
TRS2 Starship 3	6.00
TRS2 Association	6.00

****

TRS-80 Software Exchange
Roger W. Robitaille
P.O. Box 68
Milford, NH 03055
(603) 673-5144

GAMES

TRS2 Warfare I, 16K	$ 7.95
TRS2 Backgammon, 16K	7.95
TRS2 X-Wing Fighter II, 16K	9.95
TRS2 Taipan, 16K	9.95
TRS2 Cribbage, 16K	7.95
TRS2 Concentration, 16K	7.95
TRS2 Amazing Mazes, 16K	7.95
TRS2 Time Bomb, 16K	4.95
TRS2 Balloon Race, 16K	9.95
TRS2 Robot/Breakaway, 4K	7.95
TRS2 Tycoon, 16K	7.95
TRS2 Slalom, 16K	7.95
TRS2 Ten Pin, 16K	7.95
TRS2 All Star Baseball, 16K	7.95
TRS2 End Zone, 16K	7.95
TRS2 Space Battles, 16K tape	14.95
TRS2 Space Battles, 32K disk	19.95
TRS2 Star Trek III.3, 16K	14.95
TRS2 Dog Star Adventure, 16K	9.95
TRS2 Safari, 16K	7.95
TRS2 Treasure Hunt, 16K	7.95
TRS2 Round the Horn, 16K	9.95
TRS2 Pork Barrel, 16K	9.95
TRS2 Kamikaze, 16K	7.95
TRS2 All Star Baseball, 16K	7.95
TRS2 Pentominoes, 16K	9.95
TRS2 Journey to the Center of the Earth, 16K	7.95

****

APPLE SOFTWARE

Apple Computer, Inc.
10260 Bandley Drive
Cupertino, CA 95014

Apple Apple Trek/Space War	$ 15.00
Apple Color Demo/Brick Out	7.50
Apple Blackjack/Slot Machine	7.50
Apple Biorhythm/Mastermind	7.50
Apple Wari/Othello/Dragon maze	7.50

Apple Hammurabi, Pinball, Sink the Ship, Catch, Curves, Seven, Towers of Hanoi, Nightmare #6, Othello, Blackjack, 23 Bricks, Yahtzee.

****

Apple Pi Library
Terry N. Taylor
12319 East Bates Circle
Aurora, CO 80014
(303) 750-5814

Over 2,000 (Programs for trade at either Colorado School of Mines meetings or by mail with other groups and individuals via disc, cassette or listing).

****

Automated Simulations
P.O. Box 4232
Mountain View, CA 94040
(415) 964-8021

Apple Starfleet Orion	$ 19.95
Apple Temple of Apshai	24.95

****

Creative Computing Software
P.O. Box 789-M
Morristown, NJ 07960
(800) 631-8112
IN NJ (201) 540-0445

Apple Space Games—I (four programs)	$ 7.95
Apple Sports Games—I (four programs)	7.95
Apple Strategy Games—I (five programs)	7.95

GAMES CATALOGUE Continued

Apple Space and Sports Games	14.95
Apple Strategy and Brain Games	14.95

****

Programma International, Inc.
3400 Wilshire Boulevard
Los Angeles, CA 90010
(213) 384-0579

Apple Alien Encounters 32K Cass A	$ 6.95
Apple Alien Invasion 16K Cass	9.95
Apple Alley 16K Cass I	6.95
Apple Trivia Box 48K Disk I	19.95
Apple Baseball 16K Cass	15.95
Apple Basketball 8K Cass	9.95
Apple Battlefield 48K Cass	9.95
* Apple Battlestar I, 32K Cass I	15.95
Apple Blockade 16K Cass I	9.95
* Apple Breakthru 16K Cass	9.95
Apple Canter Downs 16K Cass I	15.95
Apple Chase 8K Cass I	9.95
Apple Color Startrek 8K Disk I	6.95
Apple Coney Island 16K Cass I	9.95
Apple Country Driver 8K Cass I	6.95
Apple Cross Chase 8K Cass I	6.95
Apple Death Race 16K Cass I	15.95
Apple Death Charge 16K Cass I	15.95
Apple Earthquest 36K Cass I	19.95
Apple Echo 32K Disk I	6.95
Apple Echo – Hires 16K Cass I	9.95
Apple Flyswatter 8K Cass I	9.95
* Apple Fool's Spool 8K Cass I	6.95
Apple Frustration 16K Cass I	9.95
Apple Galactic Battle 16K Cass I	9.95
Apple Gunfight 8K Cass I	6.95
Apple I Ching 24K Cass A	15.95
Apple I Ching 32K Disk A	19.95
Apple Jumpout/Shooting Stars 8K Cass 1	6.95
Apple Kaleidoscope 16K Cass I	9.95
Apple King 32K Cass A	6.95
Apple Laser Blast 16K Cass I	9.95
Apple Laser Turret 8K Cass I	6.95
Apple Leapfrog 16K Cass I	6.95
Apple Lunar Lander 16K Cass I	9.95
Apple Lunarscape 8K Cass I	6.95
Apple Magic Squares 32K Cass I	9.95
Apple Match Wits 16K Cass I	9.95
Apple Motocross 16K Cass I	9.95
Apple Mouse Hole 8K Cass I	6.95
Apple Peg Jump 16K Cass	9.95
* Apple Phaser Zap 16K Cass I	15.95
Apple Pirates! 8K Cass I	9.95
* Apple Planets 48K Cass A	15.95
Apple Ricochette 16K Cass I	9.95
Apple Saucer War 32K Cass A	15.95
Apple Scramble 16K Cass I	9.95
Apple Space Wars 16K Cass I	9.95
Apple Speedway 16K Cass I	15.95
Apple Star Dodger 8K Cass I	6.95
Apple Star Voyager 32K Cass I	15.95
Apple Stunt Cycle 16K Cass I	15.95
* Apple Super Dungeon 48K Disk I	24.95
Apple Othello 16K Cass	15.95
* Apple Starwars 32K Cass I	15.95
* Apple Tarot Cards 16K Cass I	15.95

****

Softape Software Exchange
10756 Vanowen St.
North Hollywood, CA 91605
(213) 985-5763

Apple Bomber	$ 9.95
Apple Best of Bishop – disk	39.95
Apple Rocket Pilot, Saucer Invasion	12.95
Apple Star Wars, Space Maze	12.95
Apple Forte	19.95
Apple Coney Island	12.95
Apple Crazy 8's	12.95
Apple Craps	14.95
Apple Fighter Pilot	12.95
Apple Gomoku	14.95
Apple Journey	19.95
Apple Jupiter Express	9.95
Apple Microgammon	14.95

****

Speakeasy Software Ltd.
Box 1220
Kemptville Ontario
KOG 1 J0, Canada
(613) 258-2451

Apple Bulls & Bears	$ 12.00
* Apple Warlords	12.00
Apple Kidstuff	12.00
Apple Pigskin Playoffs	12.00
Apple Wheelers Dealers	37.50

****

PET SOFTWARE

Adventure International
Scott Adams
Box 3435
Longwood, FL 32750
(305) 862-6917

PET Adventureland	$ 7.95
PET Pirates Adventure	7.95

(All require 24K)

****

Automated Simulations
P.O. Box 4232
Mountain View, CA 94040

PET Starfleet Orion	$ 19.95
PET Temple of Apshai – 32K	24.95
PET Datestones of Ryn – 16K	14.95
PET Invasion of Orion – 16K	19.95

****

Commodore PET (tm)
(Pet is the trademark of
Commodore International, Ltd.)
Commodore Systems
3330 Scott
Santa Clara, CA 95054
(408) 727-2260 for repairs

PET Target Pong and Off the Wall	$ 9.95
PET A Treasure Trove of Games	9.95
PET Galaxy Games	9.95
PET Black Jack	9.95
PET Space Flight	9.95
PET Spacetrek	9.95

****

Creative Computing Software
P.O. Box 789-M
Morristown NJ 07960
(800) 631-8112
In NJ (201) 540-0445

PET Logic Games-1, (six programs)	$ 7.95
PET Number Games-1, (six programs)	7.95
PET Sensational Simulations-1, (five programs)	7.95
PET Logic Games-2, (six programs)	7.95
* PET Graphics Games-1, (five programs)	7.95
* PET Graphics Games-2, (six programs)	7.95
PET Conversational Games-1, (five programs)	7.95
PET Board Games-1, (seven programs)	7.95
PET Action Games (six programs)	7.95

****

Creative Intricacies
P.O. Box 3663
Orange, California 92665
(714) 637-1320

PET Demolition Derby – Crash into numbers!	$ 4.00
PET Frogs – How many bugs can you eat?	4.00
PET Maze – Try to find your way out!	4.00
PET Slalom Skiing – Ski down slalom course	4.00
PET Collision – Don't get hit by the car!	4.00
PET William Tell – Shoot the apple and win!	5.00
PET Blockade II – Try to block your opponent	5.00
PET Five Card Draw – Play against PET!	5.00
PET Smog – Kill the dragon and escape	5.00
PET World War II – Destroy your opponent	5.00
PET Galaxy Invaders (mach. lang.) – Fast action!	8.00
PET Enterprise-NCC-Destroy the Klingons!	10.00

(The above PET software can be used with new or old PET'S! Make checks payable to BRENT KLINCHUCH.)

****

Creative Software
P.O. Bix 4030
Mountain View, CA 94040

* PET Breakout (machine language)	$ 10.00
* PET Seawolf (machine language)	10.00
PET Star Wars Shootout (req. Fairchild Joy.)	10.00
PET Space War II (req. Fairchild Joy.)	10.00
PET Road Race (req. Dual Joystick)	10.00
PET Tag (req. Dual Joystick)	10.00

****

Fantasy Games Software
P.O. Box 1683
Madison, WI 53701

PET Swordquest	$ 12.95
PET Escape from the Death Planet	12.95

(both for $21.95)

****

Instant Software, Inc.
Peterborough, NH 03458
(603) 924-7296

* PET Casino II-(0015)	$ 7.95
* PET Checkers/Baccarat (0022)	7.95
* PET Trek-X(0032)	7.95
PET Tangle/Super-Trap (0029)	7.95
PET Qubic-4/Go-Maker (0038)	7.95
PET Echo (0039)	7.95
PET Penny Arcade (0044)	7.95
PET Baseball Manager (0062)	14.95
* PET Dungeon of Death (0064)	7.95

****

PET User Group
P.O. Box 371
Montgomeryville, PA 18936

PET Racetrack, Othello, Bagels, Star Trek, Blackjack, Do-All, Trap, Super Mastermind, Lunar Lander, Nim, Qubic, King Breakout, Swatplot, Wumpus, Shark Bait, Flea Race, Awari, Chase, Star Lanes, Kaleidoscope, Hex Dec, Doodler, Gnip Gnop, Pong, Curfit, Battleship, Moonlander, Crypto, Jotto, Hammurabi, Slot Machine, Craps, Breakout – sound, Pop Shot, Stars, Pounce, Star Wars, Audo-Doodle, Match Game, Brain Strain, Snake, Yahtzee, Hangman, Maze, Backgammon, P Pong, Othello/2, Tanks, Starwars, Keyboard, Marks, Trace, Checkers, Simon, Klingon Capture

****

PETWARE
Peter Ruetz
368 Albion Ave.
Woodside CA 94062

PET Star Battle – graphics and sound	$ 7.95
PET Mars Landing – graphics and sound	7.95
PET Football – graphics and sound	7.95
PET Trader-Raider – for sophisticated players	14.95
* PET Breakout	

(orders under $15, add $1 s & h)

****

Programma International, Inc.
3400 Wilshire Blvd.
Los Angeles, CA 90010
(213) 384-0579

PET Banner Plus (Banner+)	$ 9.95
PET Battleship	9.95
PET Blockade	9.95
PET Craps/Tic-Tac-Toe	9.95
PET Depth Charge	9.95
PET Dominoes	9.95
PET Drone	6.95
PET Echo (PET Echo)	6.95
PET Hunt	19.95
PET Life	9.95
PET Pong	9.95
PET Quickdraw/Toughdraw	9.95
PET Random	6.95
PET Roulette (PET Roulette)	6.95
PET Scramble	6.95
PET Shooting Gallery	9.95
PET Slot Machine	9.95
PET Space Wars	9.95
PET Startrek	9.95
PET SupeReverse	6.95
PET Thousand Miles (1000 Miles)	9.95
PET Three of a Kind	6.95
PET Word Search	6.95
PET Z-A-P	6.95

PET GAMES *Reviewed by Len Lindsay*

PROGRAMMA INTERNATIONAL

Programma International at 3400 Wilshire Blvd., Los Angeles, CA 90010, has quite a library of PET programs. Most will work on either OLD or NEW PETs, but some have two versions, one for each model of PET. The cassette and any written instructions come nicely packaged in a plastic pouch, suitable for display in a store.

Blockade – This is a starship battle game. $9.95
Startrek – This is a version of the classic Startrek. $9.95
Quickdraw – Good graphics show a shoot out between you and the PET. Get him before you are shot. $9.95
Depth Charge – Your mission is to sink the subs. $9.95
Slot Machine – Realistic graphics make this one of the best slot machine programs. $6.95
Dominoes – You against the PET in a game of Dominoes. $9.95
Shooting Gallery – Shoot at both stationary and moving targets. $9.95
Three of a Kind – You against the PET in this strategy game. $6.95
Letter Squares – This is the standard puzzle of 4 by 4 blocks with one missing. You must slide the blocks around to get them in the right order. $6.95
Head Start & Ad/Ims – Test your memory with Head Start and watch Ad/Ims twist around some popular sayings. $9.95
Briefing Charts – Use this is place of the large paper sheets when doing a briefing – can do graphics or lower case. $19.95
Craps & Tic Tac Toe – Play the gambling dice game or challenge the PET to tic tac toe. $9.95
Bug & Scissors/Rock/Paper – Bug (also known as Cootie) is played with you against the PET. You and the PET play the scissors/rock/paper game with graphics. $9.95
Super Reverse – This is a good version of the reverse puzzle. You try to get the 9 numbers in correct order. $6.95
Trace A Word – How many words can you discover inside a 4x4 matrix of letters. $9.95
Banner+ – Create banners of letters on your printer of extra large letters. $9.95
Scramble – Can you unscramble this word? eslambrc. $6.95
1000 Miles – This is a good version of the card game 'Mille Bornes' pitting you against the PET. Don't get a flat tire. $9.95
Z-A-P – Move around the screen to hit the targets, but watch out for decoys. $6.95
Drone – Challenge the PET in this air to ground combat game. $6.95
Sign System – Have a moving billboard on the screen. $9.95
PET Routlette – Here is a graphically displayed roulette game for the PET. $6.95
Word Search – How many words can you make from the letters in this word: computing. $6.95
Space War – A Star Wars battle against the tie fighters. $9.95
PET Echo – Use with or without sound – match the tones and graphics the PET generates. $6.95
Pong – For 0, 1, or 2 players. $9.95
Hunt – Invent your own ADVENTURE games without having to learn BASIC. Or use the sample game that comes with it. 2 programs and data files. $19.95.

CREATIVE COMPUTING TAPES

Creative Computing (PO Box 789-M, Morristown, NJ 07960) has a series of programs on tape for the PET. Each tape has about 5 or 6 programs each recorded twice, once on each side of the cassette and sells for $7.95. A small manual is included with each tape, but instructions usually are also part of the program so it is OK if you lose the written instructions. The games I have played from these tapes seem to be very well done, highly user oriented with good graphics. Recommended.

Awari – This is an ancient African game of strategy. It is you against the PET, and the PET gets better the more it plays.
Hexpawn – You vs. the PET on a 3 by 3 chess board using 3 pawns each. This is another game where the PET gets better the more it plays.
Bagels – Similar to Mastermind but with only 3 digits in the secret code. You try to guess the computers code.
Chomp – This game is for two or more players taking bites out of a cookie. You lose when forced to bite the poison square.
Flip-Flop – This is a puzzle where you try to flip a row of 10 X's to O's.
Hi-Q – This is the classic solitaire puzzle in which you jump pegs and try to end up with only one peg in the center hole.

Action Games
Splat – Try a parachute jump, but don't go splat when you hit ground.
Car Race – Race your car around a jagged maze-like track.
Breakout – Here is the classic video game on the PET. Knock out all the bricks with a bounding ball using a small paddle controlled by the keyboard.
Bowling – Bowling for up to 4 players.
Tank – For two players. Try to blow up your opponent before he blows you off the map (including walls, trees and lakes).
Subs – Drop depth charges from your ship at subs while they fire missiles at you. Many levels of play by changing speed, missiles, etc.

Sensational Simulations
Animal – This has to be one of the best games for children of all ages (from 2 to 80). Instead of you guessing what the PET is thinking of, the PET will try to guess what animal you are thinking of by asking you yes/no questions. Any time it can't guess an animal it will add the new one to its data base. Thus it continually adds to the animals it can identify to the limit of your PET's memory.
Fur Trader – Watch out for the Indians as you try to get the best prices for your furs.
Hammurabi – Also known as Kingdom, this classic computer simulation puts you in charge on an ancient country, Sumeria. You decide how to feed your people, cultivate the land and trade with neighboring states. See if you can survive 10 years.
Stock Market – Start with $10,000 and see how well you do buying and selling stocks.
Word – Guess what word the computer is thinking of. This is not HANGMAN, though in some ways similar.

Graphics Games 1
Chase – This is another classic game for two players chasing each other about a maze of obstacles.
Escape – This unusual game puts you in prison. You must escape through 11 passageways patrolled by robots.
Dart – See how quickly and accurately you can answer an arithmetic problem.
Snoopy – This is a cute educational math game, teaching use of the number line, both positive and negative. Curse you RED BARON!
Sweep – Similar to the DEFLECTION game, you try to hit 9 targets in the correct order with your rollerball.

Graphic Games 2
Artillery – This is a great 2 player game.
Checkers – You against the PET. Best for beginning players since the PET only looks one move ahead.
Dodgem – An interesting strategy game.

Board Games
Yahtzee – Up to 4 players can play.
Blackjack – PET is dealer for up to 4 players.
Backgammon – A good graphic version.
Trek3 – Another Startrek game.
One Check – This puzzle involved trying to remove all checkers from a grid.

Number Games
Guess – A version of Hi-Lo.
Letter – Alphabetic version of Guess.
Number – Only 1 guess per round.
Trap – Trap the computer's number between your two.
Stars – Similar to Guess but the PET tells you how close, but not in which direction.
23 Matches – Similar to Nim.

Conversational Games
Eliza – This is the classic psychoanalyst program.
Hurkle – You have only 5 moves to find the Hurkle on a 10 x 10 grid.
Hangman – Another classic.
Hexletter – An intriguing strategy game against either the PET or a friend.
Haiku – The PET composes Haiku for you.

Logic Games
Rotate – The classic 4x4 puzzle.
Strike 9 – A fun dice game.
Nim – Another classic.
Even Wins – You against the PET in this Nim-like game with 1 pile of beans.
Not One – A dice game against the PET or a friend.
Batnum – Similar to Nim.

HAYDEN PROGRAM TAPES

Hayden programs come on cassette with a manual neatly packaged in a heavy plastic pouch. They all loded first time every time for me. Programs seem to be of high quality. Order from Hayden Book Company, 50 Essex Street, Rochelle Park, NJ 07662.

Backgammon – This is a good graphic version of Backgammon. It is you against the PET. The PET seems to play a good game. $10.95
Crossbow – Here is an excellent example of a fun yet educational computer game. There are several levels of play. This game will help you learn fractions, or give you plenty of practice using fractions. The fun part is shooting arrows at the target. Recommended for teaching fractions. $9.95
Mayday – If you are into small plane aviation you may enjoy this simulation of landing your plane when your engines just quit operating. The simulation is good and uses the PET graphics for its display. $9.95
Batter Up, Microbaseball Game – This baseball game has the best graphics of all the PET baseball games I have seen. This game is for two players, alternating pitching and batting. As batter you have two kinds of swings to make and your timing is important (you see the ball coming across the screen). $10.95
Soon to come are a PET Checkers game and SARGON II for the PET (a chess playing program).

HELP EXPAND THIS LIST

Please send your comments, ideas and suggestions to me at this address: Len Lindsay, 1929 Northport Dr., Room 6, Madison, WI 53704.

A BIT OF GALAXY - II

BY BRETT A. TONDREAU

The play-by-mail computer game seems to be gaining popularity, as there are increasing numbers of enthusiasts. This is a particularly innovative version. Players figure out which activities they want to undertake in Galaxy-II then their turns are run through the programs monthly. Looks like a new genre in computer games has begun!

If you would like additional information, write Galaxy-II at P.O. Box 7968, Van Nuys, CA 91409. Please send a quarter for postage and handling and please don't forget to include your address!
TD

Have you ever wanted to conquer an entire Galaxy? Or name yourself 'Emperor of a hundred solar systems'? Or simply yearned for command of the potent force of a 450 mile long Andromeda class starship? Well, I did. These ideas motivated me to create a great space game. I call it Galaxy-II.

Galaxy-II is played by mail by people all over the United States and Canada. Each game is run monthly when we take turns of all the players in the game, keypunch them, and run them through the Galaxy-II programs on an IBM 3033 mainframe computer. The programs produce laser printed reports detailing every activity of every ship captain, starship, asteroidal port and planetary colony under your command.

The rules to Galaxy-II were originally written in 1975 but have been updated five times. The rules explain how combat is calculated, the physics of the ships, and communication procedures with your units as well as with the other players. The rules also explain the equations used to determine the population growth rates, the efficiency of factories and the dreadful circumstances under which colonies choose to rebel.

Galaxy-II was designed to improve with use. As suggestions for improvement from the players are implemented, G-II acquires an experimental flavor, a history and a future, something like the quality of evolution in nature.

Three reports that help explain Galaxy II are included in this article. They come from the current version of the sample game used to test the programs. The first report, the Turn Interpretation, is the actual turn submitted for the race of Idiots (a race name chosen to exemplify dumb techniques). The other reports document only the second and third flight phases for the Idiot's new starship Recon. Included also is the computer-generated map for the game.

The Turn Interpretation reports back to the player the instructions he entered and tells him when they will be executed by the units they instruct. Line 14 is the header for a new starship, Recon, being built at location 3-34-2 (the second orbit within the solar system located at row 3 column 34 on the map). The captain is called 'A Nurd' and the ship is designated 'AA' (attack anything). This is the most hostile designation of the Barbarian racial group.

While line 14 actually has the vessel built, the receive orders on lines 15 through 23 (R) give the vessel its composition. Recon thus consists of: 13 Matter-Antimatter drives (MA), 13 hull units (HU) (classifying the ship as a Medium Explorer Light or MEL), 132 beams and shields (BE & SH), 100 units of life support (LS), 60 units of power support (PS) and 90 industrial units that represent 900,000 colonists and their colony foundation equipment. The colonists are placed into 'cold sleep' to reduce the quantity of life support needed while in transit to their new home.

In the First Flight phase (Phase = 2 on the listing) the Recon makes her maiden voyage, flying to 3-43 orbit 1. Items 25 through 27 are conducted at 3-43-1. Item 25 has an error. Recon has received only 90 IUs and in this instruction the player is ordering A Nurd to found colony (FC) New Glop with 900. The program will check the number of colonists on board and report a shortage to the player. The colony will then be founded with the available colonists—the 90 received earlier. The other item on line 25, the 'DB', indicates that the colony is designated as defensive barbarian (if fired on it'll return fire).

In the Second Flight phase the Recon is ordered to 2-24-1. While there, the ship receives still more power and life support while mining the three mineral groups (G 03), receiving matter-antimatter drives and hulls, and visually probing (V PB) the planet. This tactic is interesting, because the colony is inhabited by the Infoser rebellious members of a barbarian race. These rebels are the most hostile group within Galaxy-II; they fire on anything except a fellow rebel.

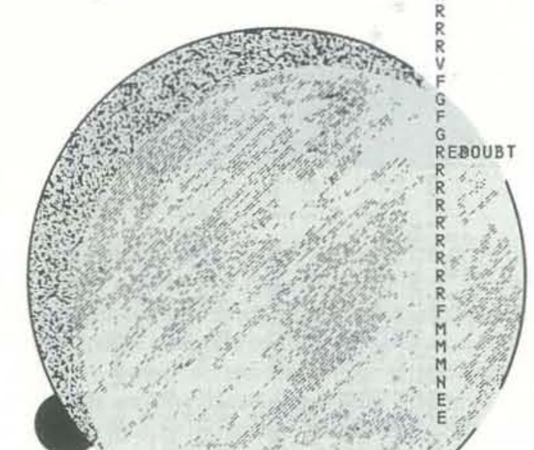
In this phase the probe report gives a picture of a planet with ample mineral resources (I, II, III), no beams or shields, six spacecraft in orbit and a population of 10^9 industrial units. Overpopulation is probably the reason for rebellion.

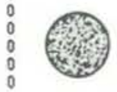
The first of the five rebel vessels, all called Revolt, opens fire on Recon with 550 beams. Luckily, Recon is better powered and is able to withstand the attack, although it loses 107 of its 130 shields. The Recon then opens fire on the Revolt and expends all 130 beams, damaging the Revolt but not destroying it. The Recon is now on the defensive since it no longer has weaponry (beams). Recon then encounters the remainder of the rebel fleet, and again, luckily, the rebel vessels are unable to harm the Recon. None of the rebel ships has enought power to fire beams. Captain A Nurd gains a great deal of experience from the encounters, progressing from a previous experience level of 21 to 306, while each of the rebel commanders gains 50 units of experience. While the rebels hold their colony and ships, the Recon fills its tanks of power and life support, receives 390 units of the colony's minerals, and A Nurd gains more experience than any of the rebel leaders.

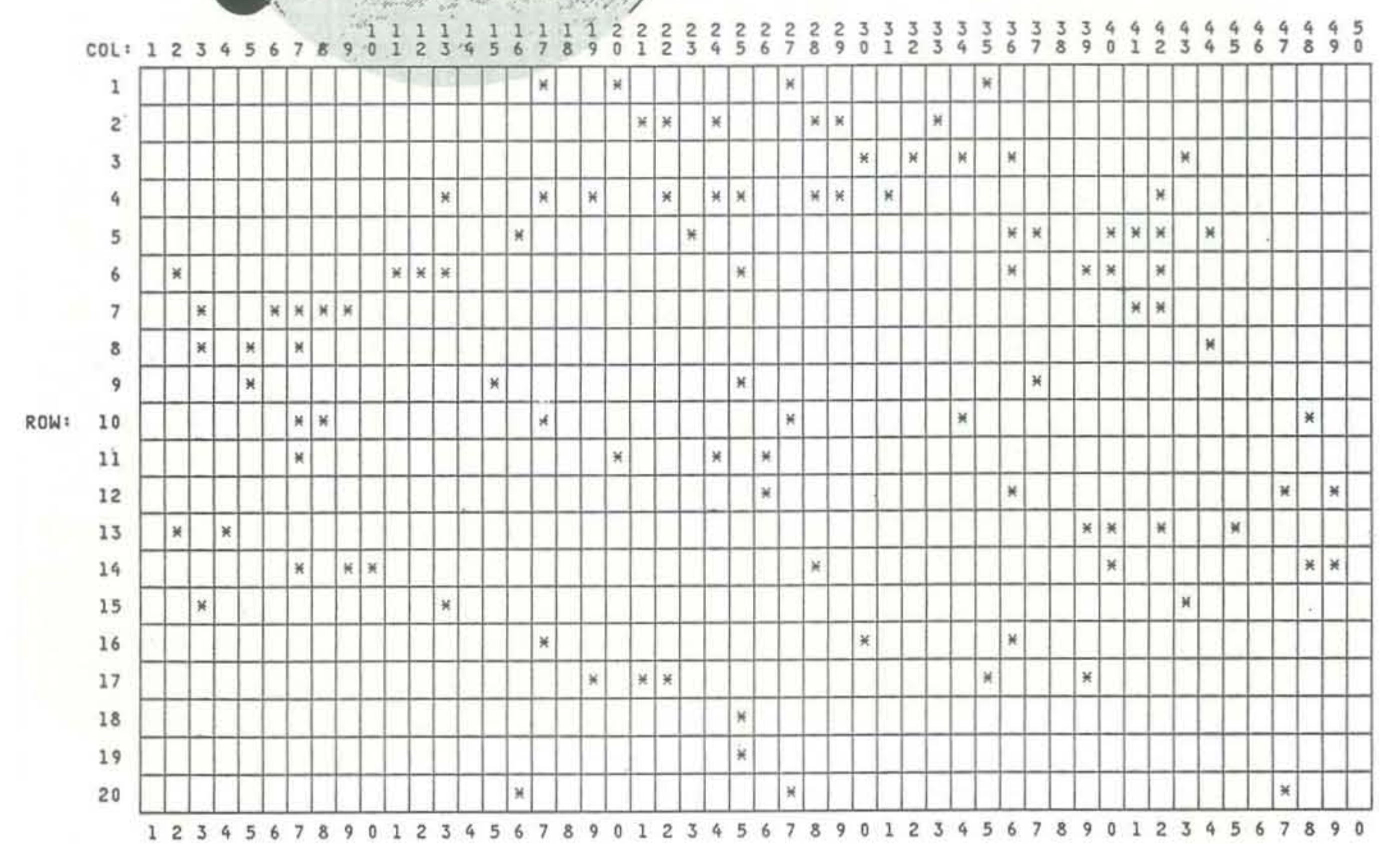
The Third Flight phase of the Recon repeats the orders of the previous flight phase, ordering the Recon to fly to an alien, barbarian orbit and take minerals. This time the Recon encounters the Infoser planet Keep Off. The Keep Off is a loyal colony of the Infoser with relatively sparce minerals. This is obvious when the Recon is unable to receive 130 units of minerals I and III. Notice that after the colony is mined, it identifies the hostile designation of the Recon, opens fire, and vaporizes our ship! The colony wins. The governor of the colony picks up all the experience of the Recon's Captain A Nurd along with a hundred or so for the extremely rare (by Galaxy-II terms) victory of a colony over a starship! All of the minerals on board the Recon eventually reenter the atmosphere of the planet and become available once more to the colony.

This covers many aspects of the game called Galaxy-II. There are many items left entirely untouched, but you now have the flavor of the game. Galaxy-II is played for fun and lasts about a year. If you play, you will be given a certificate of participation. If you win, you'll receive a certificate of victory and a handcast trophy or plaque documenting your rare achievement. Winning is not easy, but the challenge is a lot of fun.

```
IDIOTS
# CARDS:  69
# PL'S :   2
# SHIPS:   2

IDIOTS        0 RC     0                       PHASE =1    1
THIRD PL   3343 PL     0                       PHASE =1    2
B            25 PB     0                       PHASE =1    3
G          2331 02     4                       PHASE =1    4
G          3321 02     4                       PHASE =1    5
G          4311 02     4                       PHASE =1    6
G          1351 02     4                       PHASE =1    7
G          3361 02     4                       PHASE =1    8
G          5361 02     4                       PHASE =1    9
SECONDPL   3342 PL     0                       PHASE =1   10
G            26 01     0                       PHASE =1   11
G            26 01     0                       PHASE =1   12
G            26 01     0                       PHASE =1   13
RECON      3342 VE N   0 A NURD    AA          PHASE =1   14
R            13 MA     0                       PHASE =1   15
R            13 HU     0                       PHASE =1   16
R           132 BE     0                       PHASE =1   17
R           132 SH     0                       PHASE =1   18
R            65 PB     0                       PHASE =1   19
R            13 SC     0                       PHASE =1   20
R           100 LS     0                       PHASE =1   21
R            60 PS     0                       PHASE =1   22
R            90 IU     0                       PHASE =1   23
F          3431        0                       PHASE =2   24
M           900 FC     0 NEW GLOP DB           PHASE =2   25
R           100 LS     0                       PHASE =2   26
R            10 PS     0                       PHASE =2   27
F          2241        0                       PHASE =3   28
G             0 03     0                       PHASE =3   29
R           100 PS     0                       PHASE =3   30
R           100 LS     0                       PHASE =3   31
R            10 HU     0                       PHASE =3   32
R            10 MA     0                       PHASE =3   33
V             0 PB     0                       PHASE =3   34
F          2242        0                       PHASE =4   35
G             0 03     0                       PHASE =4   36
F          3341        0                       PHASE =5   37
G          1000 03 M   0                       PHASE =5   38
REBOUBT    3342 VE N   0 A BLANK   AA          PHASE =1   39
R            13 MA     0                       PHASE =1   40
R            13 HU     0                       PHASE =1   41
R           132 BE     0                       PHASE =1   42
R           132 SH     0                       PHASE =1   43
R            13 SC     0                       PHASE =1   44
R           100 LS     0                       PHASE =1   45
R            60 PS     0                       PHASE =1   46
R           900 IU     0                       PHASE =1   47
F          1351        0                       PHASE =2   48
M           300 FC     0 BANAL     DB          PHASE =2   49
M             2 SC     0                       PHASE =2   50
M             2 MA     0                       PHASE =2   51
N             0 BB     0 TRIER                 PHASE =2   52
E             0 BB     0           GV          PHASE =2   53
E             0 CP     0           GV          PHASE =2   54
```

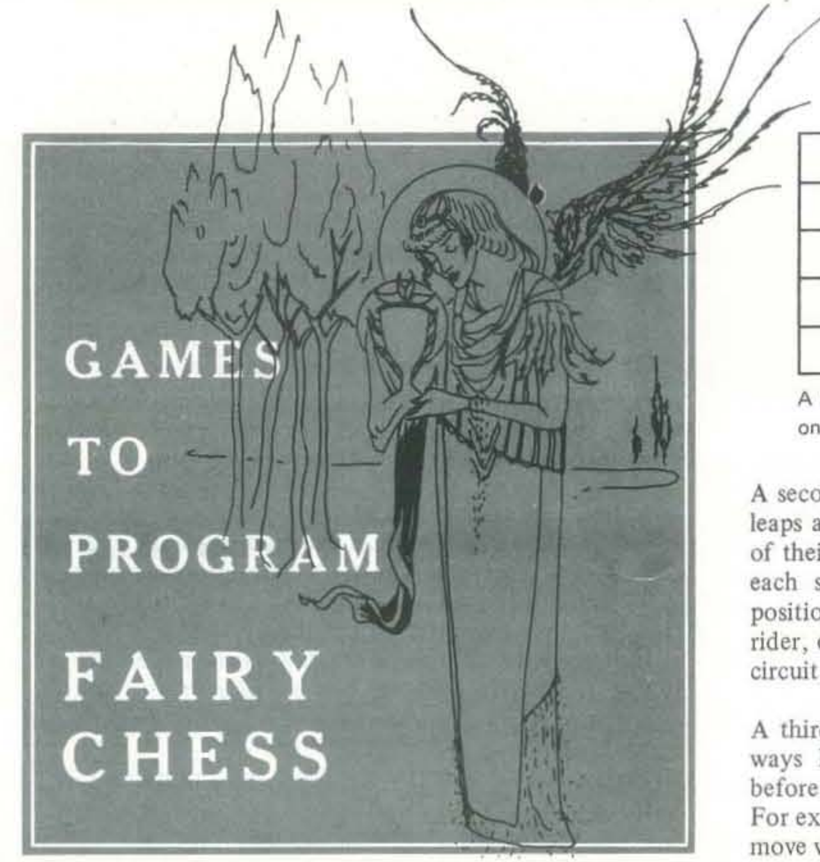






```
SHIP'S LOG OF THE VESSEL RECON    CLASS: MEL      SECOND FLIGHT PHASE

FLIGHT FROM  3-43-1 TO  2-24-1 WILL CONSUME  14.5340 UNITS OF  60.5262 UNITS PS AVAILABLE. -CAPT. A NURD
FLIGHT EXECUTED.  PRESENT LOCATION OF RECON     IS  2-24-1.
 ORIGIN  LOCATION STATUS   BEAMS  SHIELDS  DESIG      I      II     III       CRAFT   INDUSTRY               PROBE REPORT
 2-24-1   2-24-1              0        0    RB      933    918    823          6           9    NO SCREENS
WE OF THE RECON    HAVE ENCOUNTERED THE  REVOLT , A LSS  VESSEL, DESIGNATED RB, IDENTIFIED AS THE RACE OF INFOSER
VESSEL  REVOLT  HAS FIRED ON US.  IT HAS EXPENDED   550 BEAMS.  OUR SHIELDS ARE HOLDING.
THE CREW OF THE RECON    WILL ENDEAVOR TO WIN THROUGH THE DEFENSES OF THE INFOSER  VESSEL  REVOLT  IN THE NAME OF RULER A DUMMY
WE WERE UNABLE TO DESTROY THE DEFENSES OF THE STARSHIP  REVOLT
WE OF THE RECON    HAVE ENCOUNTERED THE  REVOLT , A  P   VESSEL, DESIGNATED RB, IDENTIFIED AS THE RACE OF INFOSER
THE VESSEL  REVOLT  IS HOSTILE YET HAS INSUFFICIENT POWER TO HARM US.
COMPLETE LACK OF BEAMS WILL PREVENT US FROM FIRING.
WE OF THE RECON    HAVE ENCOUNTERED THE  REVOLT , A  P   VESSEL, DESIGNATED RB, IDENTIFIED AS THE RACE OF INFOSER
THE VESSEL  REVOLT  IS HOSTILE YET HAS INSUFFICIENT POWER TO HARM US.
COMPLETE LACK OF BEAMS WILL PREVENT US FROM FIRING.
WE OF THE RECON    HAVE ENCOUNTERED THE  REVOLT , A  P   VESSEL, DESIGNATED RB, IDENTIFIED AS THE RACE OF INFOSER
THE VESSEL  REVOLT  IS HOSTILE YET HAS INSUFFICIENT POWER TO HARM US.
COMPLETE LACK OF BEAMS WILL PREVENT US FROM FIRING.
WE OF THE RECON    HAVE ENCOUNTERED THE  REVOLT , A  P   VESSEL, DESIGNATED RB, IDENTIFIED AS THE RACE OF INFOSER
THE VESSEL  REVOLT  IS HOSTILE YET HAS INSUFFICIENT POWER TO HARM US.
COMPLETE LACK OF BEAMS WILL PREVENT US FROM FIRING.
CAPTAIN A NURD   REPORTING ALL HOSTILE ALIEN VESSELS  ARE NOW CLEARED FROM THIS ORBIT.
THE COLONY BELOW IS DESIGNATED RB THE INHABITANTS BELONG TO THE RACE OF INFOSER
BY YOUR ORDERS THE VESSEL RECON    WILL ATTACK THE COLONY NAMED  REVOLT   INHABITED BY MEMBERS OF THE RACE OF INFOSER
OUR LACK OF WEAPONS PREVENTS US FROM FIRING ON THE COLONY  REVOLT
THIS COLONY APPEARS HOSTILE AND READY TO FIGHT BUT IT SEEMS WEAPONLESS!
VESSEL   LOCATION  CAPTAIN   BEAMS SHIELDS  PROBES SCREENS   MADS   SECURITY    TRIBUTE  POWER SUPPORT  MINERAL CARGO  MINERAL-II
RECON      2-24-1  A NURD        0      23      65      13     13     ( )              0      99.9999                       130
                EXPERIENCE  PASSENGER & EXPERIENCE  CLASS  HULLS  DESIGNATION  INDUNITS  LIFE SUPPORT     MINERAL-I  MINERAL-III
                       306                       0   MEL      13       AA             0      99.9899            130        130

SOLAR SYSTEM REPORT AT  2-24 (RACE,DG,COND.,COLONY): (INFOSER  RB    REVOLT ) (INFOSER  AA   KEEP OFF) (ASTEROID AS **ASTEROID)
AND  7 VESSELS (BY ORB,HULL,DG,C,&RACE): ORBIT 0: ORBIT 1: RECON    AA  MEL  IDIOTS    REVOLT  RB  LSS  INFOSER   REVOLT  RB   P
     INFOSER   REVOLT  RB   P   INFOSER   REVOLT  RB   P   INFOSER   REVOLT  RB   P   INFOSER  ORBIT 2: ORBIT 3: FOIBLE   B   LSL
     INFOSER  ORBIT 4: ORBIT 5: ORBIT 6: ORBIT 7: ORBIT 8: ORBIT 9:
```

```
SHIP'S LOG OF THE VESSEL RECON     CLASS: MEL      THIRD FLIGHT PHASE

FLIGHT EXECUTED.  PRESENT LOCATION OF RECON     IS  2-24-2.
WE WERE UNABLE TO RECEIVE MORE THAN THE PLANETS TOTAL SUPPLY OF        11 M1'S.
WE WERE UNABLE TO RECEIVE MORE THAN THE PLANETS TOTAL SUPPLY OF       126 M3'S.
CAPTAIN A NURD   REPORTING ALL HOSTILE ALIEN VESSELS  ARE NOW CLEARED FROM THIS ORBIT.
THE COLONY BELOW IS DESIGNATED AA THE INHABITANTS BELONG TO THE RACE OF INFOSER
BY YOUR ORDERS THE VESSEL RECON    WILL ATTACK THE COLONY NAMED KEEP OFF  INHABITED BY MEMBERS OF THE RACE OF INFOSER
THE INHABITANTS ARE SO COMPLETELY HOSTILE IN THEIR BARBARISM THAT THEY WOULD FIRE UPON US. WE ATTACK AND OFFER NO QUARTER.
OUR LACK OF WEAPONS PREVENTS US FROM FIRING ON THE COLONY KEEP OFF
WE ARE ENGAGED IN A FIERCE ATTACK ON OUR VESSEL BY THE PLANET KEEP OFF
MAYDAY, OUR SHIELDS FAIL .....
VESSEL   LOCATION  CAPTAIN   BEAMS SHIELDS  PROBES SCREENS   MADS   SECURITY    TRIBUTE  POWER SUPPORT  MINERAL CARGO  MINERAL-II
RECON      2-24-2  A NURD        0       0      65      13     13     ( )              0      99.9998                       260
                EXPERIENCE  PASSENGER & EXPERIENCE  CLASS  HULLS  DESIGNATION  INDUNITS  LIFE SUPPORT     MINERAL-I  MINERAL-III
                       351                       0   MEL      13       AA             0      99.9799            141        256

SOLAR SYSTEM REPORT AT  2-24 (RACE,DG,COND.,COLONY): (INFOSER  RB    REVOLT ) (INFOSER  AA   KEEP OFF) (ASTEROID AS **ASTEROID)
AND  7 VESSELS (BY ORB,HULL,DG,C,&RACE): ORBIT 0: ORBIT 1:  REVOLT  RB   P   INFOSER   REVOLT  RB   P   INFOSER   REVOLT  RB   P
     INFOSER   REVOLT  RB   P   INFOSER  ORBIT 2: RECON    AA  **   IDIOTS   ORBIT 3:  REVOLT  RB  **   INFOSER  FOIBLE   B   LSL
     INFOSER  ORBIT 4: ORBIT 5: ORBIT 6: ORBIT 7: ORBIT 8: ORBIT 9:
-----------------------------------------------------------------------------------

SHIP'S LOG OF THE VESSEL REDOUBT   CLASS: MEL      THIRD FLIGHT PHASE

FLIGHT EXECUTED.  PRESENT LOCATION OF REDOUBT  IS  1-35-3.
CAPTAIN TRIER    REPORTING A SUCCESSFUL FOUNDATION OF COLONY TRIVIAL  AT  1-35-3 WITH  300 INDUSTRIAL UNITS.
COLONY TRIVIAL  REPORTS SUCCESSFUL ADAPTATION.  INITIAL CONDITIONS INDICATE THE CAPACITY OF PLANET AS 0.34 . -CAPT. TRIER
WE HAVE TRANSFERRED         1 SCS TO THE COLONY TRIVIAL
WE HAVE TRANSFERRED         1 MAS TO THE COLONY TRIVIAL
VESSEL   LOCATION  CAPTAIN   BEAMS SHIELDS  PROBES SCREENS   MADS   SECURITY    TRIBUTE  POWER SUPPORT  MINERAL CARGO  MINERAL-II
REDOUBT    1-35-3  TRIER       132     132       0       0     10     ( )              0      45.0986                         0
                EXPERIENCE  PASSENGER & EXPERIENCE  CLASS  HULLS  DESIGNATION  INDUNITS  LIFE SUPPORT     MINERAL-I  MINERAL-III
                        21                       0   MEL      13       AA             0      84.9950              0          0

SOLAR SYSTEM REPORT AT  1-35 (RACE,DG,COND.,COLONY): (IDIOTS   DB   BANAL   ) (IDIOTS   DB   TRITE   ) (IDIOTS   AA   TRIVIAL )
AND  1 VESSELS (BY ORB,HULL,DG,C,&RACE): ORBIT 0: ORBIT 1: ORBIT 2: ORBIT 3: REDOUBT  AA  MEL  IDIOTS   ORBIT 4: ORBIT 5: ORBIT 6:
     ORBIT 7: ORBIT 8: ORBIT 9:
```

GAMES TO PROGRAM FAIRY CHESS

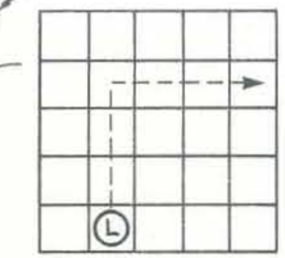
BY HERBERT KOHL

Fairy chess could equally well be called fantastic chess or inventive chess. It is not a game but rather a concept of all the different chess-like games that could be invented and played. Imagine a rook that could only move three squares at a time, a bishop whose move power shrank each time it was moved so that it started out moving along the whole diagonal and after 8 moves shrank into nothingness. Think also of a game where the object is to trap the queen or force your opponent to capture your king. The board could be any size from a 5 x 5 square on up. All these are fairy chess variations.

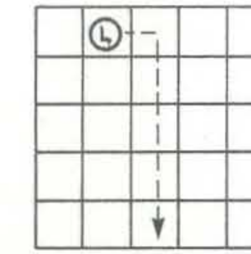
The idea of studying all variations of chess was developed by Henry Tate of Melbourne, Australia in 1914. Tate coined the name fairy chess. The classic books on fairy chess were written by T.R. Dawson. Dover has printed five of these under the single title *Five Classics of Fairy Chess.*

Dawson distinguishes a number of different types of chess-like pieces. He calls one group *leapers* since they can leap from one point of the board to another, over occupied squares. In Chinese chess, a cannon can fire (leap) across the Yellow River which divides the board, and thus can recapture one of its own pieces. More generally, leaper's moves can be described by the coordinates (m, n) where m is a number indicating vertical moves following the lateral moves.

A knight is a (2,1) or (1,2) leaper. Leapers with greater range would be (3,3), (2,4), etc. Even more powerful leapers would be described in the following way: (x,y) where x ranges from 1 to 5 and y ranges from 1 to 5. A leaper with the characteristics (x, y) where x and y can take any value from 1 to 7 could move to any square on a standard chess board at any time. A very powerful creature!



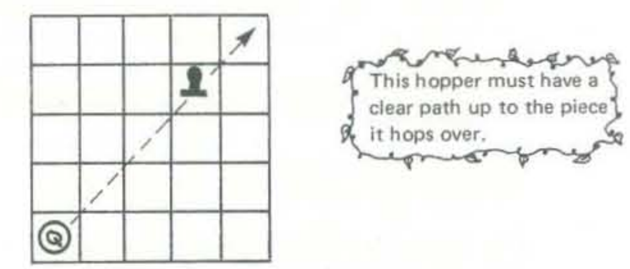
A (3,3) Leaper Move on a 5 X 5 board



A (1,4) Leaper Move on a 5 X 5 board

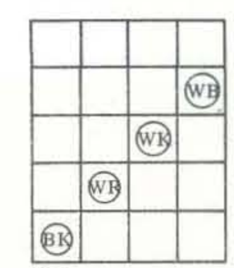
A second type of piece is called a *rider*. Riders make successive leaps as long as all the squares they could land on in the path of their leap are unoccupied. A bishop is a (1,1) rider—that is, each step on its path is diagonally related to the original position. Night-rider is the name given to a knight turned rider, one that cannot leap but can only travel an unoccupied circuit.

A third type of piece is the *hopper*. Hoppers move in fixed ways like other pieces, only they must hop over a piece before they can land on a square right beyond the piece. For example, the Grasshopper moves like a queen but can only move when it can hop:

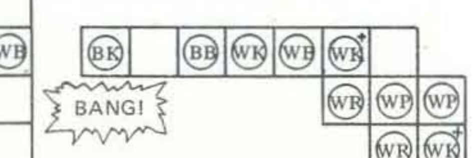


Dawson lists other pieces he calls *composite men* because their moves are a combination of the moves of other standard chess pieces. For example, the queen's move is a composite of the rook's and bishop's moves. The *Empress'* moves are a composite of the rook's and the knight's, while the *Princess'* are a composite of the knight's and the bishop's.

In addition to changing the moves of pieces it is also possible to change the shape of the chess board. T.R. Dawson presents a few complex chess puzzles on modified boards using ordinary chess moves:



White to move, mate in 3



Revolver Practice
white to move, mate in 21

It is also possible to change the rules of chess. Dawson calls these changes "limitations" and states that "In my classified collection of fairy problems I have systematically analyzed

the more obvious types of limitation in 10,000 classes." Some limitations are

1. Captureless chess—the object of both players is to prevent capture
2. Checkless chess
3. Self-mating—the goal is to be the first to mate your own king.
4. Helpmates—both players work together to mate either black or white king, and they also collaborate on ways to avoid the mate.

I can imagine other limitations. For example, it is possible to give numerical values to all the pieces and give a certain number of points to each player and let him choose his pieces. He sets them up in any way he desires on the first two ranks of each side of the board. For example:

Queen 25
Bishop 5
Knight 3
Rook 7
Pawn 1

Each player starts with the king placed on the usual square. He also has 60 points and can use those points to choose any combination of pieces adding up to sixty.. For example, in one game the two sides can choose as follows:

Black		White	
2 queens =	50	1 queen =	25
1 rook =	7	3 rooks =	21
1 knight =	3	2 bishops =	10
	60	4 pawns =	4
			60

Each player sets up his side and the game commences. Of course there can be interesting limitations on how the game is set up. The players may

a) take turns placing their pieces on the board,
or
b) decide how their pieces are to be set up, draw a diagram indicating their choices and then place all pieces on the board. This is similar to the simultaneous move system used in some strategy wargames.

It is probably possible to modify existing chess programs so that fairy chess versions can be attempted. The following questions might be considered:

(1) How can boards be modified and the number of pieces reduced so that many different variations can be tested in order to discover challenging games? (2) What are the criteria for challenging games? (3) What interesting goals can be set up for game variations? (4) Is it possible to play a game that results in both players ending with no legitimate move (such a game might be called Traffic Jam)? (5) Is it possible to play so that pieces have magic powers of appearing at different times during the game or control not only their own moves but also those of other pieces? The possibilities are greater than Dawson's 10,000 limitations, and it would be interesting to see what you readers can come up with.

For the readers who are interested in more information on Fairy Chess, there is an excellent section in Martin Gardner's book, New Mathematical Diversions From Scientific American, *New York: Simon & Schuster, pages 74-75, 1966. Within that and earlier sections of the above, Mr. Gardner discusses many variations of chess and chess-like games. He mentions Edgar Rice Burroughs' game of Martian Chess as chronicled in* The Chessman of Mars. *Mr. Gardner also points the reader to the science-fiction novel* The Fairy Chessman, *by Lewis Padgett. In this last book, the protagonist successfully wins a war because of his experiences with the game of Fairy Chess.*

In the bibliography section of Diversions, *there is a short list of references to chess variation games, including this one on Fairy Chess:*

Chess Eccentricities. Major George Hope Verney. London: Longmans, Green and Co., 1885. The best reference in English.

"Fairy Chess." Maurice Kraitchik in *Mathematical Recreations*, pages 276-279. Dover, 1953.

"Variations on Chess." V. R. Parton in *The New Scientist* (an English weekly), page 607, May 27, 1965.

Les Jeux d'Echecs Non Orthodoxes. Joseph Boyer. Published by the author, Paris, 1951.

Nouveaux Jeux d'Echecs Non Orthodoxes. Joseph Boyer. Published by the author, Paris, 1954.

Les Jeux de Dames Non Orthodoxes. Joseph Boyer. Published by the author, Paris, 1956.

Cryptarithms

BY JOHN DAVENPORT CREHORE (Nine Hex)

Jack Crehore (Nine Hex) is on the move. He now lives in Virginia. To speed up the process of getting your solutions to him, we want you puzzlers to send your solutions to: Nine Hex, P.O. Box 96, Charlotte Court House, VA 23923.

Solutions for the 1979 Cryptarithms, through the Sept.-Oct. issue, appeared in the Letters section of the Jan.-Feb. 1980 RC. *Check there to see how well your solutions matched up with Jack's for last year's puzzles.* —*RZ*

NEW PUZZLES

Puzzle 17 (Novices)

```
  YMA HKM FJT
  TMN HMY JMT
M MYT KJA NTF
```

Hints: Read samples and examples. Also, look at hints in the Comments from Nine Hex on the next page.

Puzzle 18 (Adepts)

```
        BU
MHU_/CRTBK
    CBMT
     CCBK
     KABN
      URN
```

Hints: Work with zero, nine, a squared digit, and two digits taken as a pair.

Puzzle 19 (Geniuses)

List No. 1	List No. 2
RB	RB
TU	UT
NR	NR
KU	KU
KUK	RKM

Hints: An adding machine list with an error—corrected with the rule of nines. Turn on your logic for this one.

Puzzle 20 (Computers)

```
MANBTK
UAUMTK
NAUYTK
NAWBTK
CAWMTB
```

Hints: Simple computer additions—have your machine chew on this one for a while.

SAMPLES AND EXAMPLES

Puzzle 1 *Puzzle 1 Solution*

```
  B   5
  B   5
  B   5
  ---
 CB  15
```

The number 5 is the *only* digit that works.

Puzzle 2 *Puzzle 2 Solution*

```
  R   9
  R   9
  T   1
  ---
 TR  19
```

The number 9 is the *only* digit that works.

Puzzle 3 *Puzzle 3 Solution*

```
  AC   32
 xAC  x32
 ----  --
  FH   64
  JF   96
 ----  --
YBCH 1024
```

The letter A can *only* have the value 3. If A were more than 3, three letters would be needed in place of JF; if A were less than 3, the total would not reach 1,000.

Puzzle 4 *Puzzle 4 Solution*

```
  A, ABB, FCB    8,844,534
+ H, KCF, FKB   +1,235,524
 -------------  ----------
 HM, MAM, MFA   10,080,058
```

In column one, A *must* be an *even* digit. Why? The H must be a 1. Right? So HM is easy. That solves for B in column one; then column four; then column three . . . Get it? Stay with it a while and the logic starts jumping off the page.

COMMENTS FROM NINE HEX (JACK CREHORE)

Well, fans and puzzlers! Your letters are marvelous—keep them coming. Your "educated guessing" shows a rare talent. Nobody solved (without a hint) Puzzle 11 in the last issue of *RC*. Everyone insisted D could be *zero*, thus indicating either a typing error or a mathematical irrationality. There is no error! Two of the exponents are positively identifiable by logic. One factor identifies both its digit and exponent. There is a letter that is zero, but it is *not* D. D might be a *one*. Try that! The typography may have thrown some of you: two letters without a space stood for a two-digit number. Remember: the puzzle was for geniuses!

Here are some tutorial hints for Puzzle 17 on the previous page: In column 10, M can't be anything but *one*. This gives the result in column 9, also a *one*, and indicates that the Y in column 8 is either *two* or *three*.

For column 9, M is *one*; Y is *two* or *three*—how big is T? T must be either *nine* or *eight*. Now the fun begins. Think fast!

In column 7, A+N results in T. The sum of two different digits is at most 17 (9+8), but T is either 9 or 8 so A+N is less than 17 (in fact, less than 10), and there is no carry. So, Y is a *two*!

Back to column 9. If Y is a *two*, T is a *nine*. Now look at column 1. How much is F? F is an *eight*. There is a carry to column 2. J turns out to be a *seven*. Now from column 1 to column 10 each position is solvable with letter values found in the previous columns. How's that for fun?

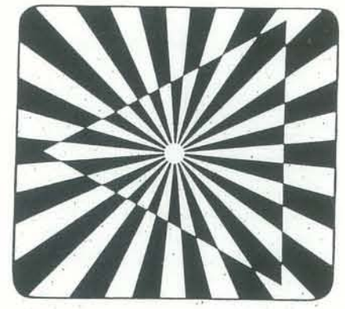
Polarbaer: Smart research by you for another modulus for Computer Puzzle 12! Error *TUL* for *TAL* frustrated you and the others. You discovered the error by analysis. Great work!!

Crazyman: You also did a beautiful research piece on Number 12. Puzzles 9 and 10 you belittled. I must explain. *Recreational Computing's* introductory pledge was to help in education. We *have* to give some step-by-step encouragement to beginners.

Puzzle 10 was a bit easy for adepts. We will make the adept category more of a challenge—beginning with this issue.

David Hubbard: Your contribution in *RC* 41 drew comment from Polarbaer and others who found the starter of triple T good, as I did too. Send more! We can't promise comments on each submission. *RC* cannot check the accuracy of contributions; so, *Puzzlers Beware*! It's all fun though. Contributors, welcome!

THE KID: Your letters are so good! Read about Puzzles 11 and 12 above. Nobody got No. 11; 12 had a typo error, there is no *need* for the *sign*! Now you've given me the chance to remind everyone that *in Puzzledom* "All's fair in love and war!" Common courtesy would require "not base 10" perhaps; but the use of *twelve* letters would relieve the *need*.
—Nine Hex



PUZZLERS' SOLVING RECORD

Puzzlers' Current Issue

Number	#42, Nov-Dec '79	Puzzle No. 13	14	15	16
16	K8VDU	●	●	●	●
27	Gedasm	●	●	●	●
28	Polar Baer	—	●	●	●
29	Touriguy	—	●	●	X
31	Black Vulther	●	●	—	—
32	Metatron	●	●		

		In #42	Total
1	Dab		6
2	David Hubbard		4
3	TWW		4
4	Spock		2
5	C.W. Moore		4
6	Scott		4
7	Willz		4
8	Judy		4
9	Hobo		4
10	GCG #3		4
11	The Kid		6
12	Bobby Baum		4
13	Richard HA		1
14	SR McEntee		4
15	Dr. Dave Marquis		4
16	K8VDU	4	10
17	Charles I. Goldman		4
18	Mark		0
19	Crazy Man		6
20	Daddidwjm		4
21	Michael Richter		1
22	Prof. J. Householder		0
23	Keith B. Lewis		4
24	Joseph W. Kmoch		4
25	Jack Dushey		2
26	Bob C. Washburne		4
27	Gedasm	4	8
28	Polar Baer	3	8
29	Touriguy, Mr. & Mrs. Paul	3	4
30	Ctein		4
31	Black Vulther	4	10
32	Metatron	2	6

PROGRAMMING PROBLEMS

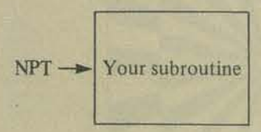
BY BOB ALBRECHT AND DON ALBERS

PROBLEM #5 FAST PRIME NUMBER TESTER

Do this one. You will need the solution in future programming problems. Write a *subroutine* to determine whether a positive integer NPT is a prime number.

- If NPT is not a prime number, set PPT = 0 and set FPT equal to a factor of NPT *other than* 1 or NPT (1<FPT< NPT), then RETURN.
- If NPT is a prime number, set PPT = 1 and set FPT = 1, then RETURN.

Your subroutine has one input (NPT) and two outputs (PPT and FPT).



→PPT = 0 if NPT is not prime
1 if NPT is prime.

→FPT = Factor between 1 and NPT if NPT is not prime.
1 if NPT is prime

If you write your subroutine in Microsoft™ BASIC, use the following program to test it.

```
100 REM***PROBLEM NO. 5 PRIME NUMBER TESTER
110 REM***RECREATIONAL COMPUTING, MAR-APR 1980

200 REM***GET AN INTEGER, 1 TO 999999
210 PRINT : INPUT "INTEGER, PLEASE (1 TO 999999)" ; N
220 IF N <> INT(N) OR N<1 OR N>999999 THEN 210

300 REM***TELL WHETHER N IS PRIME OR NOT PRIME
310 NPT = N : GOSUB 510
320 IF PPT = 1 PRINT N "IS A PRIME NUMBER."
330 IF PPT = 0 PRINT N "IS NOT A PRIME NUMBER." FPT "IS A
    FACTOR."

400 REM***GO TO 'GET AN INTEGER'
410 GO TO 210

500 REM***YOUR SUBROUTINE (CALLED IN LINE 310)
510 and so on
```

Wait! Before you plunge in, we want to add two small constraints.

- We want your subroutine to be FAST!
- You may **not** use arrays. Especially you may **not** store a bunch of primes, then use them as divisors to test NPT. Really! That's just brute force; eventually you will run out of memory. We can always pick a number too big to be tested by a bunch of primes stored in memory, especially if we remove the 999999 restriction in line 210.

Now for some good news! You may assume, in your subroutine, that NPT has been checked out and is a legitimate customer. You may assume that NPT is a positive integer in the range 1 to 999999. If your BASIC can handle more than six digits, modify line 220 of our tryout program and go for more.

PROBLEM #6 SUM OF DIVISORS

Below is a dreadfully slow program to compute and print the sum of the divisors of a positive integer. This program is in Microsoft™ BASIC for the TRS-80.

```
100 REM***PROBLEM #6 SUM OF DIVISORS
110 CLS

200 REM***GET A POSITIVE INTEGER, N
210 PRINT : INPUT "POSITIVE INTEGER, PLEASE" ; N
220 IF N<>INT(N) OR N<=0 THEN 210

300 REM***COMPUTE SUM OF DIVISORS OF N
310 SUM = 0
320 FOR D = 1 TO N
330 IF N/D = INT(N/D) THEN SUM = SUM + D
340 NEXT D

500 REM***PRINT SUM OF DIVISORS. GO BACK FOR NEW N
510 PRINT "SUM OF DIVISORS =" SUM
520 GOTO 210
```

The above program is OK for small positive integer values of N but, alas, seems to take forever for larger values.

Your job: Rewrite lines 310 on—make the program *fast*. Send your solutions. Later, we might publish hints (send some hints) on how to build a fast sum of divisors program. Hmmm . . . is it merely coincidence that this is problem number 6?

PROBLEM #7 GAMEMASTER'S DICE

If you play *Dungeons and Dragons, Runequest, Tunnels and Trolls* or *Chivalry and Sorcery*, you know about gamemaster's dice. They are tetrahedrons, hexahedrons (cubes), octahedrons, dodecahedrons and icosahedrons. If you are a teacher and haven't heard about the above games, ask your students about them. In the unlikely event that they can't fill you in, read "What Is All This Stuff? Beginner's Guide to Fantasy Role-Playing" in the May-June 1979 issue of *RC*.

The world of fantasy role playing games uses the following dice to simulate events in the nonreal world.

D4	Tetrahedron, numbered 1 through 4.
D6	Cube, numbered 1 through 6.
D8	Octahedron, numbered 1 through 8.
D10	Icosahedron, numbered 1 through 10, with each number appearing twice.
D12	Dodecahedron, numbered 1 through 12.
D20	Icosahedron, numbered 1 through 20.
D 100	Two icosahedrons. Each die is numbered 0 through 9. One die is the tens digit; the other is the ones digit. A roll of 00 is interpreted as 100. This pair of dice is used to roll percentages from 1 to 100.
D3	A roll of a D6 with results as follows: 1 or 2 = 1, 3 or 4 = 2, 5 or 6 = 3

In fantasy role playing games, events are sometimes determined by rolling 2 or more of the same type of dice.

3D6	Roll three six-sided dice (cubes). The result of the rolls is the sum of the individual dice (3 to 18).
5D8	Roll 5 eight-sided dice (octahedrons). The possible results range from 5 to 40.

We want a program to simulate gamemaster's dice, as described above. Here is a sample RUN of the program that we have in mind.

```
ROLL? D6                    <-- We ask for one six-sided die
 3                              The computer rolls a 3.
ROLL? 3D6                   <-- We ask for 3 six-sided dice
 14                             Good roll!
ROLL? 2D12                  <-- We ask for 2 twelve-sided dice.
 13
ROLL? D100                  <-- We ask for percentage dice.
 100                            Hmmm . . . they came up 0,0.
ROLL? 3D6                   <-- Computer ignores spaces.
 11
ROLL? #3%7                  <-- Complete garbage!
I DON'T UNDERSTAND
ROLL? #3D%6
I ASSUME YOU MEANT 3D6      <-- Clever computer!
 11
ROLL? 2D7                   <-- These aren't on our list of dice.
I DON'T HAVE A D7
ROLL? 0D4                   <-- Roll zero dice?
 0                              Of course!
```

We hope you get the idea. Any reasonable request for dice should be accepted and the computer should "roll the dice" and print the result. Put yourself in the place of the gamemaster. He or she is concentrating on running the game; the computer is waiting patiently to help. The computer should provide help as *helpfully* as possible.

Solutions to this problem will be useful in many future problems. Solve this one, then tuck the solution away where you can find it and apply to future problems in the world of fantasy role playing games.

PROBLEM #8 RECURSIVE NUMBER PATTERNS

Each of the following number patterns is a sequence of integers.

(1) 1, 2, 3,. . .	(2) 2, 5, 8,. . .
(3) 2, 4, 8,. . .	(4) 3, 12, 48,. . .
(5) 1, 11, 111,. . .	(6) 9, 99, 999,. . .
(7) 43, 433, 4333,. . .	(8) 32, 332, 3332,. . .
(9) 34, 334, 3334,. . .	(10) 12, 102, 1002,. . .

Each sequence is of the form: $s_1, s_2, s_3, \ldots$
Each sequence can be defined recursively, as follows.

s_1 is given
$s_k = f(s_{k-1})$ for k = 2, 3, 4,. . .

For example,

(2) $s_1 = 2$ $s_k = s_{k-1} + 3$ for k = 2, 3, 4,. . .
(6) $s_1 = 9$ $s_k = 10s_{k-1} + 9$ for k = 2, 3, 4,. . .

In fact, every sequence above (and many others!) can be defined as follows.

s_1 is given.
$s_k = f(s_{k-1}, m, a, b)$ for k = 2, 3, 4,. . .
where m, a, and b are constants.

Sequences (5) through (10), and many similar sequences, appear in elementary school textbooks. They all produce interesting patterns when each term of the sequence is squared.

Now to our problem. Complete the following program to compute and show the first N terms of a sequence. The program first asks for the number of terms desired (N), the first term (S) and the constants M, A and B which define the sequence.

```
100 REM***PROBLEM #8 NUMBER PATTERNS
110 REM***RECREATIONAL COMPUTING, MAR/APRIL 1980
120 REM***VARIABLES
120 REM*** N = NUMBER OF SEQUENCE NUMBERS TO SHOW
130 REM*** S = FIRST NUMBER IN SEQUENCE
140 REM*** SK = NEXT NUMBER IN SEQUENCE
150 REM*** M, A, B = CONSTANTS USED TO COMPUTE NEXT
          NUMBER

200 REM***ENTER NUMBERS TO DEFINE SEQUENCE
210 PRINT:PRINT
220 INPUT "HOW MANY TERMS"; N
230 INPUT "FIRST TERM"; S
240 INPUT "M = "; M
250 INPUT "A = "; A
260 INPUT "B = "; B

300 REM***INITIALIZE: START KTH TERM (SK) AT FIRST
          TERM (S)
310 SK = S

400 REM***COMPUTE AND SHOW N NUMBERS OF SEQUENCE
410 CLS
420 FOR K = 1 TO N
```

430 PRINT SK,

440 SK = _____

450 NEXT K

500 REM***GO BACK FOR A NEW SEQUENCE
510 GOTO 210

Also, please show the values of S, M, A, and B which you would enter for each sequence (1) through (10).

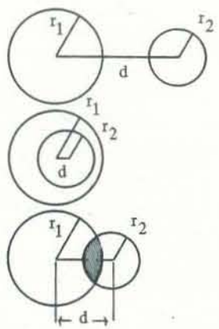
SEQUENCE	S	M	A	B
(1)				
(2)				
(3)				
(4)				
(5)				
(6)				
(7)				
(8)				
(9)				
(10)				

PROBLEM # 9 TRICKY AREA NO. 1

The problem about Newett Awl and the Goat (*RC*, July-August 1979) reminded us of this problem, which first appeared in *Advanced Problems for Computer Mathematics*, published by Digital Equipment Corporation.

Write a program to compute the area of the *intersection* of two circles, given the radius of each circle and the distance between the centers of the two circles.

Here are some possibilities.



Under control of *your* program, the computer should ask for values of r_1, r_2, and d, then compute and print the area.

PROBLEM # 10 1980

This one is easy. Complete line 20 in the following BASIC program so that the computer will print the number of this year, 1980.

```
10  CLEAR SCREEN
20  YEAR = ___
30  PRINT YEAR
40  END
```

Wait! There are a few rules.

(1) Your BASIC expression must include every nonzero digit: 1 2 3 4 5 6 7 8 9
(2) Every nonzero digit must appear once, and only once, in line 20.
(3) You may not add additional lines to the program.

It can be done. Here are some examples.

```
20  YEAR = 4 * 5 * 9 * (8 + 3) * (7 + 1) / (6 + 2)
20  YEAR = 45 * (89 - 67) * (3 + 1) / 2
20  YEAR = 198 * (7 + 3) * (4 + 5 - 2 - 6)
```

The above examples are quite mundane. Dwellers of Xanth and other magical places will use the full power of BASIC (or even PASCAL) to generate the number 1980.

PROBLEM # 11 WORD'S WORTH NUMBER ONE

This is the first in a series of problems about computing numbers from words. For the first few problems, we assign number scores to the letter A through Z, as follows:

A = 1	G = 7	M = 13	S = 19	Y = 25
B = 2	H = 8	N = 14	T = 20	Z = 26
C = 3	I = 9	O = 15	U = 21	
D = 4	J = 10	P = 16	V = 22	
E = 5	K = 11	Q = 17	W = 23	
F = 6	L = 12	R = 18	X = 24	

A word's worth is its numerical value, obtained by adding the values of the letters in the word. For example, HOBBIT is worth 56 points, DRAGON is worth 59 points and WIZARD is worth 81 points.

Write a program to compute a word's worth. A RUN of your program might look like this:

```
YOUR WORD?  WIZARD
YOUR WORD IS WORTH 81 POINTS

YOUR WORD?  ISN'T
YOUR WORD IS WORTH 62 POINTS

YOUR WORD? FLIP-FLOP
YOUR WORD IS WORTH 92 POINTS

YOUR WORD? ABCD
YOUR WORD IS WORTH 10 POINTS

YOUR WORD?  3#AB%*Z
I DON'T UNDERSTAND

YOUR WORD?  and so on...
```

Your program should compute the score, or worth, of any word or even any string of letters, even if it isn't a word.

Contractions and hyphenated words are OK.

Your program should accept strings which include:

- Letters, A through Z
- Apostrophes (')
- Hyphens (-)
- Spaces

This problem is the beginning of a bunch of problems. So, *do* solve this one. If you want more information, all you have to do is ASCII us (Oops! Sorry about that) by sending a Self-Addressed Stamped Envelope (SASE) to the DRAGONS, P.O. Box 310, Menlo Park, CA 94025.

Dragonsmoke

- THIRTEEN is worth 99 points. Are there numbers whose English name has a word's worth equal to the number? If so, what is the smallest such number?

- AHA is worth 10 points, AHA is a palindrome, but . . . sigh . . . 10 is not a palindrome. Are there words which are palindromes for which the word's worth is also a palindrome? Hmmm . . . if not in base 10, perhaps in some other base. See Problem # 2 Palindrome Numerals in *RC*, Jan/Feb 1980, page 57.

PROBLEM # 12 SCRABBLE SCORES

In the game of SCRABBLE, each letter has a number score, as shown below.

A = 1	G = 2	M = 3	S = 1	Y = 4
B = 3	H = 4	N = 1	T = 1	Z = 10
C = 3	I = 1	O = 1	U = 1	
D = 2	J = 8	P = 3	V = 4	
E = 1	K = 5	Q = 10	W = 4	
F = 4	L = 1	R = 1	X = 8	

A word score is the sum of the scores of the letters in the word. For example, HOBBIT is worth 13 points and WIZARD is worth 19 points.

Write a program to compute the score for a word, or string of letters, using the letter scores shown above. A RUN of your program might look like this:

```
YOUR WORD?  HOBBIT
YOUR SCORE IS 13

YOUR WORD?  WIZARD
YOUR SCORE IS 19

YOUR WORD?  3#AB%
I DON'T UNDERSTAND
```

And so on. The computer should accept any "word" consisting of a string of letters, even if it is not a real dictionary word, but reject any string containing characters other than the letters A through Z.

PROBLEM # 13 SCRABBLED PRIME NUMBERS

In the game of SCRABBLE, each letter has a number score, as shown below.

A = 1	G = 2	M = 3	S = 1	Y = 4
B = 3	H = 4	N = 1	T = 1	Z = 10
C = 3	I = 1	O = 1	U = 1	
D = 2	J = 8	P = 3	V = 4	
E = 1	K = 5	Q = 10	W = 4	
F = 4	L = 1	R = 1	X = 8	

A word score is the sum of the scores of the letters in the word. For example, PLAY is worth 10 points and KANGAROO is worth 13 points.

Next, we bring you the first 25 prime numbers:

2	3	5	7	11	13	17	19	23	29
31	37	41	43	47	53	59	61	67	71
73	79	83	89	97					

Perhaps you have already guessed the problem. For each of the first 25 prime numbers, find a word for which the word score is equal to the prime number. You may use any single unabbreviated word in *Webster's Third New International Dictionary* (Unabridged) even if it is not acceptable in the game of

If you have solved Problem # 12 Scrabble Scores, you may wish to modify your program to help you in this problem.

PROBLEM # 14 EXOTIC NAMES IN FANTASY ADVENTURELAND

Suppose you are creating a fantasy adventure such as a *Dungeons and Dragons, Runequest*, or *Tunnels and Trolls*. You may wish to use unusual names for your heroes, wizards, monsters and other creatures. You could, of course, borrow names from fantasy adventure books such as *Lord of the Rings*. But, perhaps you prefer to invent your own.

Why not use your computer to help you invent names? Sounds OK, but how do we get the computer to print names that are pronounceable (or almost so) and seem to be unusual, exotic or even fantastic?

That's the problem. Write a program to generate and print or display random names that might be used in a fantasy or science fiction game or story.

Yes, this does sound more like a *project* than a *problem*. Good! We hope you will project your best efforts into this problem . . . er, project . . . and send us some cunningly contrived programs to generate fantastic names. Let's see now —how many orcs would have to type for how many years to write *Lord of the Rings?*

From the Gourmet Math Kitchens of San Francisco
Introducing

Delicious Functions #1

BY RITA LIFF LEVINSON & TED KAHN

With special thanks to Diane Resek, Dept. of Mathematics, S. F. State University, and
From Nearby Berkeley
A Special Guest Appearance by
The Lawrence Hall of Science
FUNCTION MACHINE

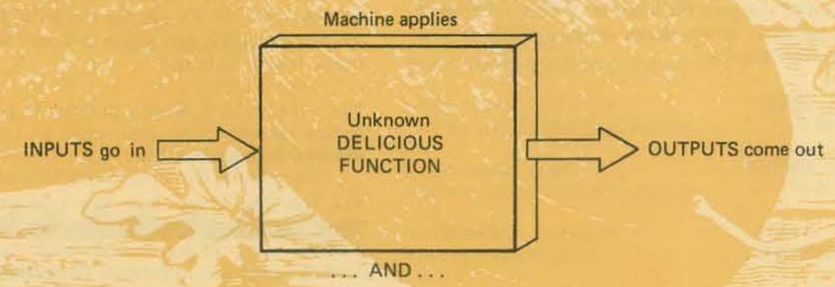
RC welcomes you to the wonderful world of *Delicious Functions!* Here you will find delights for the mind, delicacies which defy description, pleasant pastries for the problem-solving palate . . . all made in the secret depths of a Magical, Mystery *Function Machine!*

And WHAT, you may ask, is a Delicious Function? Or, for that matter, what is a Function Machine? Pretend there is a special machine, painted in spectacular colors, which has a space for INPUTS and a space for OUTPUTS.

(Use your imagination and add your own features to this simple Function Machine. Send your designs to Recreational Computing *and we'll publish the best of them!)*

Once turned on, our Function Machine waits for INPUTS, which it then takes in and converts—by means of some unkown Delicious function (Rule)—into mind-watering OUTPUTS! Having carefully sealed our Function Machine shut, we must use our *minds* to discover what the Delicious Function is, since we cannot see what goes on inside (Let's hope our machine never needs repair)!

So, what we have in simple form is:



Let's do a very simple example. Below is a chart, with corresponding INPUTS and OUTPUTS. Fill in the missing OUTPUTS and state the rule (that is, the Delicious Function) which causes each input to be converted into its corresponding output:

DF # 1

INPUTS →	OUTPUTS
2	2
4	2
37	2
1	2
150	----
−7	----
0	----

you fill in here

In your own words, write what *you* think the rule for DF #1 is.

DF # 1 is an example of the "constant function," because no matter what we give the machine for input, it always gives us the same number (2) for output.

Now, let's try some others. These won't be quite so easy, so *look for patterns!*

DF # 2

INPUTS →	OUTPUTS
5	9
2	3
25	49
−3	−7
100	199
30	----
−6	----
4	----

PATTERNS

In your own words, write what *you* think the rule for DF # 2 is

Or, how about this one?

DF # 3

INPUTS →	OUTPUTS
0	1
1	1
2	1
−1	REJECT!
30	2
125	3
99	2
375	----
10	----
1234	----

This is starting to look like . . .
Whoops! Don't guess too soon!
(This input can't be accepted by DF # 3)

What's my pattern?!

What's the rule for DF # 3?

Note that in Delicious Function # 3, "– 1" caused the machine some difficulty. Not all inputs will work for all functions! Do you understand what happened?

Now let's look at a slightly different kind of function (or is it really different).

DF #4
Yum!

IN	OUT
bat	3
ball	4
salt	4
green	5
fun	3
computer	8 (wow)
jump	----
aha	----
think	----

HINT: Look at DF #3 . . . Can you see a similarity?

And now, two for you to think about for next time . . .

DF #5 (Hint: It's similar to DF #4, but with a twist!)

IN	OUT
cherish	10
glad	4
once	4
village	10
green	6
glove	6
sing	4
was	2
to	----
program	----
think	----

Why?

AND **DF #6** HMM . . .

IN	OUT
strange	r
place	c
candy	d
flake	k
father	h
another	h
govern	r
girl	l
wash	h
bit	REJECT!
blade	d
winter	----
spring	----
summer	----
fall	----

Why?

What do all these outputs have in common?

Please submit your own delicious functions and next time, we will begin to publish programs in BASIC for different microcomputers so you can program your own.

DELICIOUS FUNCTION ANSWERS

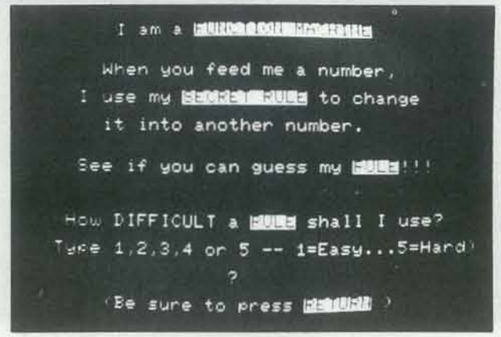
DF #	RULE
2	2x – 1, where x is input number.
3	Number of digits in x, where x ⩾ 0.
4	Number of letters in each word.
5 & 6	These answers next time!

Have fun figuring these out!!

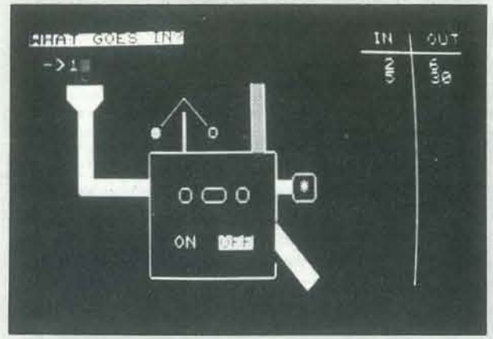
THE FUNCTION MACHINE IN ACTION

Pictures courtesy of University of California Regents, © 1979.

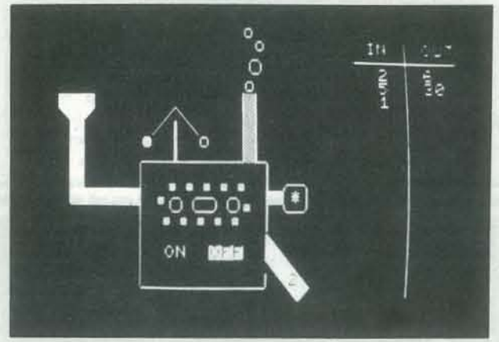
People Sampling a Few Delicious Functions



Starting Up the Function Machine



In goes a 1!



Out comes a 2!
What's the Rule?

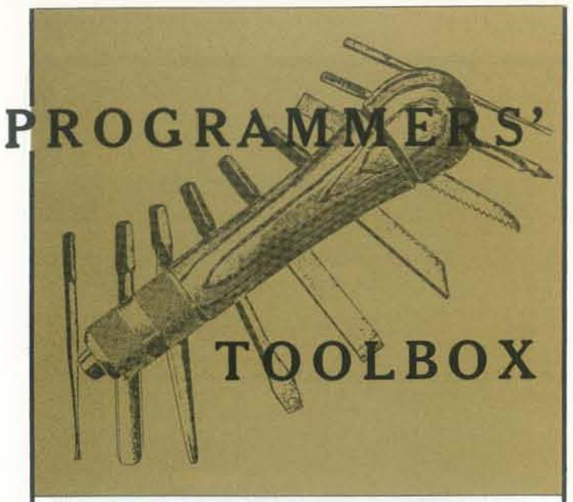
RITA LIFF LEVINSON

Rita Liff Levinson has taught at Lawrence Hall of Science, Mills College, and San Francisco State University. She has conducted mathematics and computer workshops for teachers. She is especially interested in motivating and working with math and computer avoiders.

Rita is currently a software engineer with TAK Components in Burlingame, CA. She is an active organizer of conferences that introduce women to science related careers. On March 15, 1980, through the Math/Science Network, 15 conferences will be held nationwide. The Math/Science Network started in the San Francisco Bay Area about four years ago, and now has over 500 members.

TED M. KAHN

Ted M. Kahn has been interested in educational alchemy and mysticism for nearly a decade, having replaced "earth, air, fire, and water" with "kids, games, computers, and creativity" (in no particular order). He has worked in the Computer Education Project at the Lawrence Hall of Science, lived in the sands and hills of the Middle East for two years, experienced a war, six different jobs (one for each day of the week, with one day of rest), and done graduate studies in the psychology of creative thinking and problem-solving. For the past three years, he has been in residence with the Learning Research Group at Xerox Palo Alto Research Center where he has been researching the use of computer games to teach complex problem solving skills. He is currently serving as Educational Consultant for personal computers with the Consumer Marketing Division of ATARI, Inc.



PT 16: A SIMILARITY COMPARATOR FOR STRINGS

If a computer is directed to search a file looking for a particular string of characters, a simple typographical error will cause the computer to report that no match has been found; even though there was something very close in the file. The statement "IF A$ = B$ THEN . . . " is taken literally by the computer; even the slightest difference is not tolerated.

Wouldn't it be better if a computer, finding no exact match, would report the *best* match, or the 5 best matches listed in order of closeness of match? That is what the routine illustrated here does; it computes a similarity index on a scale of 0 thru 100 percent.

The routine is written in Ohio Scientific Instruments 8 K BASIC, Version 1, and was run on a Challenger II system. (*The program will run, as is, on the Apple II with floating point BASIC, and the PET. Eds.)*

```
 10  LET  T= 0
 20  LET  P= 3
 30  PRINT "FIRST WORD";
 40  INPUT A$
 50  LET A= LEN (A$)
 60  PRINT "SECOND WORD";
 70  INPUT B$
 75  IF A$= B$ THEN PRINT "EXACT MATCH"
 80  LET B= LEN(B$)
 90  IF A > B THEN LET B= A
100  FORM = 1 TO B
110  LET C = 0
120  FOR I = 1 TO M
130  LET K$= MID$ (A$, B – M + I, 1)
140  LET L$= MID$ (B$, I, 1)
150  IF K$= L$ THEN LET C= C + 1
160  NEXT I
170  LET C= CIP
180  LET T= T + C
190  NEXT M
200  FOR M= B + 1 TO 2*B – 1
210  LET C= 0
220  FOR I = 1 TO 2*B – M
230  LET K$ = MID$9A$,I,1)
240  LET L$= MID$ (B$, M – B + I, 1)
250  IF K$= L$ THEN LET C= C + 1
260  NEXT I
270  LET C= CIP
280  LET T = T + C
290  NEXT M
300  LET S= 100*T/BfP        The up arrow indicates
310  PRINT S; "%"            exponentiation.
320  LET T = 0
330  GOTO 70
340  END
```

Sample Run — **Comments**

```
RUN
FIRST WORD? POOL
SECOND WORD? POOL
EXACT MATCH
   103.1%                  > 100% because of double letter
?  POOR                    3 letter pattern "POO" matches.
   45.3%
?  COOL                    Still a 3 letter pattern.
   45.3%
?  POO                     Same match, because nonmatching
   45.3%                   characters do not count.
?  POLO
   28.1%                   Two 2 letter matches, "PO" and
?  LOOP                    "OL", do not count as much
   18.7%                   as one 3 letter match.
?  PAIL
   12.5%                   Only 2 isolated letters.
?  POOL ROOM
   10.4%
?  MAIL ROOM
   /.5%
?  PO/OL                   Presence of extra random
   14.4%                   character reduces match.
?  0000                    Repeated letters result
   40.6%                   in unexpectedly high match.
?
OK
RUN
FIRST WORD? T. C. O'HAVER 710 HILLSBORO DR. SILVER
SPRING MD. SECOND WORD? TOM O'HAVER 710 HILLS-
BORO DR. SILVER SPRING MD.
   82.9%
?  R.D. O'HAVER 710 HILLSBOROUGH RD. SILVER
SPRINGS FL.
   10.3%
?
OK
```

BY T.C. O'HAVER

PT 17: BINGO BOARDS

This program prints bingo boards on the TRS-80's video screen. If you have a printer, you can change the program to list the boards on paper. The current program simply displays the boards on the screen. This program *does not* keep track of each board's status during the game. Someone out there with a little free time could easily make that modification. (Hint: load a board's numbers into a 2-dimensional array; change numbers to negative when 'called'; check for the 13 possible Bingos after each call.)

The idea for this routine came from Karen Chepko's Bingo "Number Calling" program Nov.-Dec. 1979 *RC*. I designed the numbering of statements so that my routine can be typed right in with Ms. Chepko's, effectively turning two programs into one. Like Ms. Chepko's routine, this one is also written in Level II BASIC and requires less than 1K.

```
10  REM *** WAYNE F. CUMMINGS      ***
12  REM *** LA CROSSE, WI.         ***
15  REM *** NOV. 22, 1979          ***
20  CLS: RANDOM: INPUT"HOW MANY BOARDS DO YOU
      WANT PRINTED"; P
25  PRINT "TOUCH 'ENTER' AFTER COPYING EACH":
      FORX = 1 TO 1000: NEXT
30  L$(1) = "B": L$ (2) = "I": L$(3) = "N" : L$ (4) = "G":
      L$ (5) = "0"
35  FOR G = 1 TO P: CLS: PRINT: PRINTTAB(3)
40  FOR K = 0 TO 60 STEP 15
45  IF K = 0 THEN 50 ELSE PRINTSTRING$ (3,13);:
      PRINTTAB(3)
50  FOR X = 1 TO 5
55  N = RND (15) + K: A(X) = N:FORB = 0TO X-1: IFA(X) =
      A(B) THEN 55ELSENEXTB
60  PRINTUSING "##" ; N, : PRINT "      "; : NEXT X
65  PRINTTAB (45) : PRINT L$ (K/15+1) ; : NEXT K
70  FORX=15360TO16320STEP192:FORY=XTOX+40:
      POKEY,131:NEXTY,X
75  FORX=15360TO15400STEP8:FORY=XTOX+959
      STEP64:POKEY,191:NEXTY,X
80  PRINTTAB (50): INPUT A: NEXT G
```

For questions on BINGO Boards, write to me at: 3218 Lauderdale Ct., La Crosse, WI 54601.

BY WAYNE F. CUMMINGS

PT 18: SORTED RANDOM NUMBERS

Have you ever needed several random numbers in a game, and wanted them to be sorted? Here is a short routine that will produce an ordered list of uniform random numbers.

```
10   REM THIS ROUTINE GENERATES SORTED UNIFORM RANDOM NUMBERS USING AN
11   REM ALGORITHM OF JON LOUIS BENTLEY AND JAMES B. SAXE PUBLISHED
12   REM AS A CMU COMPUTER SCIENCE DEPARTMENT REPORT CS-79-113
13   REM DATED 6 MARCH 1979.
14   REM
15   REM
50   RANDOMIZE
100  INPUT N
200  GOSUB 1000
300  FOR J = 1 TO N
400     GOSUB 2000
500  PRINT J, Z
600  NEXT J
700  GOTO 100
1000 L = 0.0                          } Initialization
1010 I = N                            }
1020 RETURN                           }
2000 IF I <= 0 THEN 2100              }
2010 L = L + LN(1-RND)/I              } actual routine
2020 I = I - 1                        }
2030 Z = EXP(L)                       }
2040 RETURN                           }
2100 PRINT "error: too many calls"    }
2110 STOP                             }
9999 END                              }
```

SAMPLE RUN—10 NUMBERS

```
? 10
  1          0.999781
  2          0.803244
  3          0.636599
  4          0.518174
  5          0.505454
  6          0.31189
  7          0.242318
  8          0.149439
  9          4.55302E-3
  10         1.11596E-3
```

BY DENNIS ALLISON

AMINAL-A Computer Game

BY KATHY BURK, AGE 10 & RACHEL WASSERMAN, AGE 9

Part 2: A Solution

In the Nov.-Dec. issue of RC, *Kathy and Rachel posed a challenge to other young programmers 10-years old and under. They presented the readers with a sample run of a program, and asked what others thought the program would look like.*

Ashley Henshaw, a 9-year old at Ludlow Elementary School, Shaker Heights, Ohio, sent in this solution. Ashley and her classmates have taken 8-week courses in computer literacy in both the fourth and fifth grades. Now it's your turn! Try Ashley's program on your computer, and create your own aminals. *–RZ*

Ashley Henshaw's AMINAL Program

```
 9   PRINT "TYPE YOUR NAME, PLEASE"
19   INPUT A$
29   PRINT "IN THIS GAME YOU MAKE YOUR OWN ANIMAL,"; A$
39   PRINT "WHAT DO YOU WANT YOUR ANIMAL TO BE,"; A$
49   INPUT B$
59   PRINT "WHAT SIZE DO YOU WANT YOUR ANIMAL TO BE,"; A$
69   INPUT C$
79   PRINT "NOW YOU HAVE A", C$ " "; B$ " . "; A$
88   PRINT "WHAT COLOR DO YOU WANT IT TO BE,"; A$
99   INPUT D$
119  PRINT "WHAT COLOR EYES DO YOU WANT IT TO HAVE"
129  INPUT E$
139  PRINT "NOW YOU HAVE A "; C$ " "; D$ " "; B$; "WITH"; E$; "EYES"
149  PRINT "HOW MANY LEGS DO YOU WANT IT TO HAVE,"; A$
159  INPUT F$
169  PRINT "NOW YOU HAVE A "; C$ " "; D$ " "; B$; "WITH"; F$; "LEG(S) AND ";
       E$; "EYES"
179  PRINT "WHERE DO YOU WANT IT TO LIVE"
189  INPUT G$
199  PRINT "NOW YOU HAVE A "; C$ " "; D$; " "; B$; "WITH "; F$; "LEG(S)$ AND ";
       E$; "EYES THAT LIVES IN A "; G$
219  PRINT "WELL, BY,"; A$
229  END
```

a New Algorithm for Chess

PART VI HUMAN VS. COMPUTER

BY DAVID CHELBERG & DAVID WATTERS

The final test of a computer program for chess is, of course, to play it against a human. Chelberg and Watters give the details of such a contest here, thus testing out their program. This article is the last in the series by this author team, but I'm sure we'll be hearing from them again in RC. *Lists of the program are available from the authors. They can be reached at P.O. Box 10952, Stanford, CA 94305 –TD*

The first five articles in the series explained the development and implementation of our algorithm. In our concluding article we show the success of our approach. To do this, we shall analyze a sample game, one played by a human against the computer.

The game chosen is fairly representative of the computer's major strengths and weaknesses. It is also sufficiently short to allow a detailed presentation. The human, although unranked, has been playing for about ten years and has had previous experience against our program. He is familiar with the computer's approach. This is important since a knowledge of the computer's general strategy allows its weaknesses to be accentuated and used to advantage. In the game, the computer is White and the human is Black.

OPENING MOVES

White	Black
1. P-K4	.....

When the computer moves first, it has the option of choosing either P-K4 or P-Q4. In this case, P-K4 was the random selection.

1.	P-K4
2. N-KB3	N-QB3
3. P-Q4	P-Q3

Until now, the program had been operating from its book of openings. At this point, however, Black chose to make an unorthodox move which did not appear in the book of openings. The correct move to continue the Petroff Defense is PxP. Although P-Q3 is a weak move, it serves to get the program off of the opening and on to the middle strategy where the computer has a greater chance to blunder its initial development. So White is on its own, relying solely on the developmental factors in the middle game strategy to guide its opening moves.

4. B-KN5	B-K2
5. B-K3	.....

This seemingly strange sequence of moves by White serves two purposes: By moving B-KN5, the computer attacks the queen. The human's response, although successfully attacking the bishop, is not the best. A better move is QN-K2 which stops the attack as well as avoids the complications of PQ-5. B-K3 is the only acceptable retreating move, as it increases White's center control by protecting the queen pawn.

5.	N-KB3
6. N-QB3	0-0
7. B-Q3	.....

In this case, the second choice for the computer, B-QB4 was .0006 less than B-Q3. It is our opinion that B-QB4 has a greater attacking potential and is less defensive. However, White cannot see any clear attacks, so in its opinion B-QB4 is questionable. White prefers B-Q3 as a defensive move since it increases the king-pawn protection as well as cementing its structural alignment (see Figure 1).

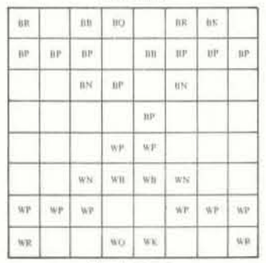


Figure 1

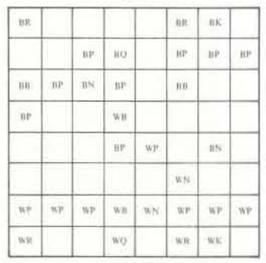


Figure 2

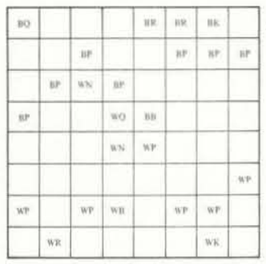


Figure 3

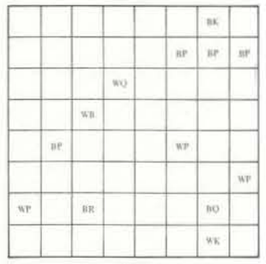


Figure 4

7.	N-KN5
8. 0-0	.....

The computer took this opportunity to castle. Positionally, castling queen-side is preferred in this situation. However, we have no castling analysis at this time. White chose to castle because it had no better move; it could find no solution to Black's threat of NxB which creates doubled king pawns. Although this could be defended by Q-Q2, it is not necessarily good to move the queen as far into the open as K3 this early in the game. The bishop itself is not free to move since it is vital to the protection of the queen pawn. Probably the best move in this situation is N-Q5. This would look even better if 7. were B-QB4.

8.	P-QN3

This is a poor move for Black since it allows the computer to pin the knight to the rook. Apparently, the human was trying to develop his bishop as well as his pawns. A better sequential approach might have been: P-QR3, P-QN4, B-N2. Yet the best possibilities for Black seem to lie in PxP.

9. B-QN5	B-N2
10. B-Q2?	.....

The computer makes its first blunder! This move results in the loss of a pawn and is due to a shortcoming in the vital protection routine. In its evaluation of the situation, white saw 3 defenders and 2 attackers. Following the capture sequence through, it deduced that only 2 defenders were needed and that any 2 of the 3 present would suffice. This meant that all 3 pieces could have complete move liberty since their absence would still leave 2 protectors. White failed to notice, however, that the bishop move not only removed the bishop's protection, but blocked the queen's as well. It moved the bishop because of the threat of the knight. A better move here is Q-K2.

10.	PxP
11. N-K2	B-B3
12. B-Q3	P-QR4
13. B-QB4	.....

White shows indecisiveness in moving the bishop. Nevertheless, White still maintains the offensive advantage. Instead of B-Q3, the computer should have considered P-KR3: if N-KR3, then BxN regains the queen pawn and destroys Black's pawn structure. If N-K4, the lost pawn is still regained. B-Q3 is good in the sense that it blocks P-Q6; however, moving B-QB4 wastes that move.

13.	B-R3
14. B-Q5	Q-Q2

Black's Q-A2 is a weak move that causes problems later (see Figure 2). A better move is B-N2. The computer is now ready to begin its assault.

15. N(K2)xP	BxR?

BxR gives White the opportunity to gain a knight advantage. Again B-N2 is preferred.

16. QxB?	.....

Had the computer moved BxN, it would have yielded a greater material gain. The computer thinks that the knight being pinned to the pawn prohibits that knight from protecting the bishop in the event of the capture.

16.	R-K1!?

This move seems good in the sense that it attacks the knight by relieving the pin. It is poor, however, because it replaces the pin with a hurdle. Better perhaps is R-QB1.

17. NxN	.....

White still does not see the knight as protecting the bishop, so it moves the knight because it is attacked. If it does not move BxN, NxN is second best.

17.	BxP
18. R-N1	B-K4
19. P-KR3	N-B3
20. Q-N5	.....

This is a preparatory attacking move. Since the queen is protected, Black's queen can be attached via NxB threatening B-B6.

20.	Q-B1
21. KN-Q4	NxB
22. QxN	.....

QxN is preferred to PxN because PxN gives Black an open rank.

22.	Q-R1

This is Black's preparatory move that led to the mate 10 moves later. (see Figure 3).

23. P-KB4	BxN ch
24. NxB	Rxp
25. Q-QB4?	.....

This move loses a knight for White. The correct move, R-N5, is given a value of 10.24 as opposed to 10.48 for Q-QB4. The computer prefers the move it made because it attacks the human's unprotected bishop pawn.

25.	P-B4
26. Q-N5	.....

Again White chooses to attack the pawn. It prefers this move to RxP because it relieves the pin on the lost knight. Best here is P-B3.

26.	RxN
27. B-K3	R-N5
28. RxR	.....

Better is Q-B1, however, the computer would rather take the free pawn.

Continued on pg. 36

TRS-80: TINY TREK

Klingons in Less Than 4K

BY MILAN D. CHEPKO

Milan, our master of small recreations, presents a Trek game for the TRS-80 that, with some modification, should work on other computers. The game takes up less than 4K of memory, and is one of the most compact versions we have seen. Blast off!! – *RZ*

With pollution, inflation, and rising gas prices, what this country really needs is another StarTrek program, right? I already have over a dozen versions of this popular game on disk, and unused listings for several more. So why this one now?

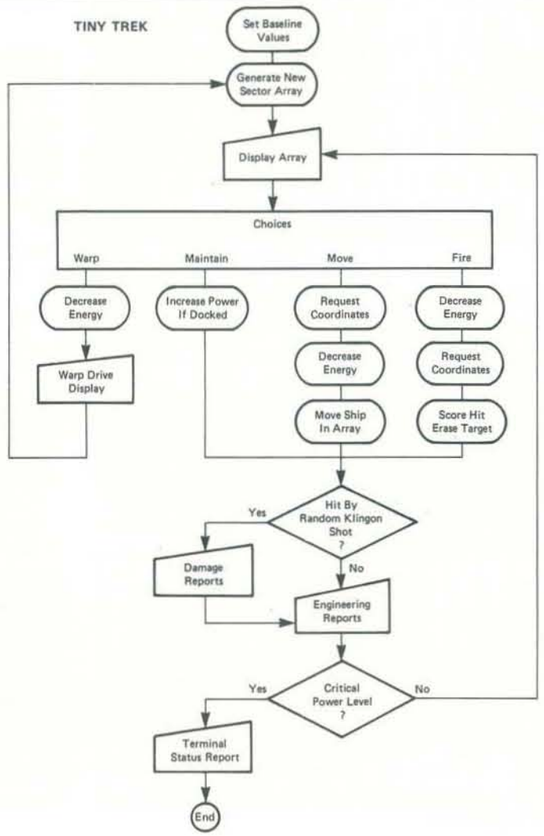
First of all, Trek is one of my early attempts at programming in TINY BASIC for my homebuilt system that I later converted to the TRS-80. The program should work on almost any system having about 4K of RAM. Second, Trek makes use of the Warp Drive subroutine that appeared in Programmer's Toolbox™ a few months ago. Finally, the program provides a relatively smooth running game, with an occasional surprise.

The flow chart and REM statements illustrate the general pattern of play, but a few additional comments might be helpful. The first 49 elements of the array store the locations of the Enterprise and Klingon ships, stars, and bases; the Enterprise is placed at the center after each warp jump, while the others are randomly distributed. Line 210 assigns a starbase to one of the 65 array elements, so there will be some fields that don't contain a reachable starbase. Since the whole game is based on available power, you can make it more or less difficult by changing the size of the array in line 130 and the RND statement in line 210. Line 890 allows for random shots from the remaining Klingons. If you have at least 1000 units of power, there will be no damage. But as the power drops, the amount of damage increases. In line 820, power can be increased by docking alongside a starbase and maintaining position, although it might be wise to eliminate any Klingons in the area first. Timing loops in lines 1000 and 1100 determine how long the status reports remain on the screen. You may wish to change these as well.

This program would also work with a teletype or line printer as the output device by replacing the warp-drive display subroutine in lines 490-670 with the cryptic message "WARP DRIVE ENGAGED."

VARIABLES USED

I, J, N	Counters
A(1)-A(49) .	Sector array
K	#of Klingons in display sector
P	Power level
D	Your decision
X, Y	Coordinates in warp-drive subroutine; coordinates in display for moving or firing




```
100 CLS:PRINTTAB(20)"STARTREK":PRINT
110 'BY MILAN D. CHEPKO, M.D.
120 'THIEF RIVER FALLS, MN  56701
130 RANDOM:DEFINT A-Z:DIM A(65):P=1000:GOSUB1190
140 '*** SET UP RANDOM STARS AND KLINGONS ***
150 CLS:FORI=1TO50:A(I)=0
160 X=RND(15)
170 IFX=3 A(I)=3
180 IFX=7 A(I)=7
190 NEXTI
200 '*** POSITION A STAR BASE AT RANDOM ***
210 X=RND(65):A(X)=2:A(25)=1:L=25
220 '*** DISPLAY LOCATIONS ***
230 CLS:I=0:J=0:K=0:X=1
240 PRINT"
250 PRINT%0,";
260 I=I+1
270 IFA(I)=0#PRINT"  ";:GOTO320
280 IFA(I)=1#PRINT" E ";:GOTO320
290 IFA(I)=2#PRINT" B ";:GOTO320
300 IFA(I)=3#PRINT" * ";:GOTO320
310 IFA(I)=7 K=K+1:PRINT" K ";
320 J=J+1:IFJ>6GOTO340
330 GOTO260
340 X=X+1:J=0
350 IFX=2#PRINT"  !  POWER LEVEL=";P
360 IFX=3#PRINT" !"
370 IFX=4#PRINT"  !  HERE ARE YOUR CHOICES:"
380 IFX=5#PRINT"  !    1=MOVE WITHIN SECTOR"
390 IFX=6#PRINT"  !    2=FIRE WEAPONS"
400 IFX=7#PRINT"  !    3=WARP AWAY"
410 IFX>8GOTO250
420 PRINT"  !    4=MAINTAIN POSITION"
430 PRINT"  !  ----------"
440 PRINT"     1  2  3  4  5  6  7" :PRINT
450 INPUT"WHAT IS YOUR DECISION";D:IFD>4GOTO450
460 PRINT
470 ON D GOTO 690 ,740 ,490 ,790
480 '*** WARP DRIVE SIMULATION ***
490 IFP<400#PRINT"ENERGY TOO LOW!":GOTO450
500 P=P-200
510 CLS:FOR I=50 TO 60:A(I)=RND(128):NEXT I
520 FOR X=0 TO 127:SET(X,0):SET(X,47):NEXT X
530 FOR Y=0 TO 47:SET(0,Y):SET(127,Y):NEXT Y
540 FOR N=0 TO 50
550 FOR I=51 TO 55
560 X=R(I):Y=R(I+5)
570 IF X>1 AND X<126 AND Y>0 AND Y<47 GOTO610
580 IF X=1 OR X=126 RESET(X,Y)
590 X=RND(90):IF X<38 GOTO590  ELSE GOTO640
600 Y=RND(36):IF Y<12 GOTO600
610 RESET(X,Y)
620 IF X<64 THEN X=X-2 ELSE X=X+2
630 IF Y<24 THEN Y=Y-1 ELSE Y=Y+1
640 SET(X,Y)
650 R(I)=X:R(I+5)=Y
660 NEXT I:NEXT N
670 CLS:GOTO150
680 '*** MOVE WITHIN QUADRANT ***
690 GOSUB850
700 IF A(M)=0 A(L)=0:R(M)=I:L=M:P=P-100:GOTO890
710 PRINT"POSITION OCCUPIED"
720 FOR J=1 TO 500:NEXT J:GOTO230
730 '*** FIRE PHOTON TORPEDOS ***
740 GOSUB850
750 IFA(M)=7 A(M)=0:PRINT"YOU SCORED A HIT!"
760 P=P-150:PRINT
770 FOR J=1TO500:NEXT J:GOTO890
780 '*** MAINTAIN POSITION AND TAKE-ON FUEL ***
790 IF(A(L-1)+(A(L+1))=2GOTO810
800 GOTO890
810 CLS:PRINT"DOCKED WITH STARBASE...";
820 P=P+RND(250)+100
830 FOR I=1TO500:NEXT I:GOTO890
840 '*** SUBROUTINE TO INPUT NEW COORDINATES ***
850 INPUT"INPUT VERTICAL";Y:IFY>7GOTO850
860 INPUT"INPUT HORIZONTAL";X:IFX>7GOTO860
870 Y=Y-1 M=7*Y+X:CLS:RETURN
880 '*** RANDOM HIT BY KLINGON SHIP ***
890 X=RND(32):IF X<25 OR k=0 GOTO1020
900 '*** DAMAGE REPORT IF HIT BY KLINGON ***
910 CLS:P=P-350:PRINT"*** WE'VE BEEN HIT!!! ***
920 PRINT:IF P>=1000 PRINT" NO DAMAGE":GOTO1000
930 PRINT"DAMAGE REPORTS COMING IN..."
940 IF P>900 PRINT" NO INJURIES":GOTO1000
950 IF P<900 PRINT" SEVERAL CREWMEN INJURED
960 IF P<600 PRINT" DAMAGE TO ANTIMATTER POD
970 IF P<500 PRINT" POD JETTISONED
980 IF P<400 PRINT" PHASER BANKS INOPERATIVE
990 IF P<200 PRINT" SHIELDS DAMAGED
1000 FOR I=1TO2000:NEXT I:PRINT
1010 '*** STATUS REPORTS ***
1020 CLS:PRINT"ENGINEERING REPORT":PRINT
1030 IFP>=1000PRINT" ALL SYSTEMS FUNCTIONING"
1040 IFP<1000PRINT" NON-ESSENTIAL SYSTEMS SHUT DOWN"
1050 IFP<800PRINT" SHIELDS HOLDING"
1060 IFP<600PRINT" LIFE SUPPORT SYSTEMS CUT BACK"
1070 IFP<500PRINT" PHASER BANKS INOPERATIVE"
1080 IFP<400PRINT" ONLY IMPULSE POWER AVAILABLE"
1090 IFP<300PRINT" SHIELDS STARTING TO FAIL!"
1100 FORI=1TO2000:NEXTI
1110 '*** TERMINAL STATUS REPORT ***
1120 IF P>=100 GOTO230
1130 CLS:PRINT"  YOUR POWER LEVEL HAS DROPPED TO";P:PRINT
1140 PRINT"ONE OF THE DILITHIUM CRYSTALS HAS CRACKED!
1150 PRINT"ANTIMATTER POD IS CRITICAL, AND WILL EXPLODE IN
1160 PRINT"SECONDS!!  SCOTTY REPORTS THE SHIELDS ARE DOWN,
1170 PRINT"AND SPOCK'S SCANNERS SHOW 3 PHOTON TORPEDOS ON
1180 PRINT"A COLLISION COURSE!!!":PRINT:END
1190 '*** INSTRUCTIONS ***
1200 PRINT"EACH MANEUVER COSTS SOME OF THE AVAILABLE POWER. TO
1210 PRINT"RECHARGE THE SUPPLY, DOCK ALONGSIDE (NOT ABOVE OR
1220 PRINT"BELOW!) A STARBASE.  MAINTAIN THAT POSITION UNTIL
1230 PRINT"YOU HAVE ACCUMULATED AS MUCH ENERGY AS YOU WISH.
1240 PRINT:PRINT"AS LONG AS THE POWER LEVEL IS ABOVE 1000, NO
1250 PRINT"HARM CAN COME TO THE ENTERPRISE.  AS THE LEVEL
1260 PRINT"DROPS, THE AMOUNT OF DAMAGE FROM A LUCKY KLINGON
1270 PRINT"HIT CAN BECOME SEVERE!":PRINT
1280 PRINT"REMEMBER THAT NOT EVERY QUADRANT WILL CONTAIN A
1290 PRINT"STARBASE!":PRINT
1300 PRINTTAB(10);"HIT 'ENTER' TO CONTINUE ";:INPUTA$:RETURN
```

RAGING ROBOTS

Landmine the PET!

BY LARRY HATCH

Raging Robots is surprisingly popular with my friends even though it appears very simple. The game relies primarily on screen memory for much of its action. The program randomly POKEs 80 grey *landmines* about the screen. Twenty *robots* appear, also at random locations. The *player* is represented by a white circular graphic near the center of the screen. Each initial screen setup is unique, with the possible combinations running into the millions.

The player can move vertically, horizontally, or diagonally by pressing the number keys:

```
7  8  9
4  5  6   The numbers five and zero do nothing.
1  2  3
```

The object of the game is to outsmart all 20 robots by luring them into landmines. This feat requires considerable skill since for every move the player makes, *each* robot gets a move. The robots' moves are always toward the player. If one robot jumps on the player's current position, the game ends.

At the end of each game, the user is prompted for a new game. Statistics on the number of games won and lost are POKEd into places in memory locations 850 and 851 (2nd cassette storage area) so that an accidental break in the game doesn't destroy these variables. A running tally of how many robots are left to be trapped appears in the top left corner of the screen.

This version of Raging Robots is simplified from my regular game that contains sound effects. My "musicbox" is of my own design (5 plus octaves), and my POKE commands are not standard. If anyone wants information on these items, write me at 22 Coleman Place # 14, Menlo Park, CA 94025.

Enjoy the Raging Robots!

```
50 POKE 850,0 POKE851,0
100 CLR PRINT"]■ RAGING ROBOTS ■"
110 PRINT PRINT" YOU ARE BEING CHASED BY RAGING ROBOTS."
120 PRINT"■■ ← RAGING ROBOT.  ● ← THIS IS YOU."
130 PRINT PRINT" TO GO UP (↑) PRESS 8. DOWN=2"
140 PRINT"TO GO LEFT (←)PRESS 4, RIGHT=6"
145 PRINT"DIAGONAL MOTION = 1,3,7 OR 9 JUST AS"
150 PRINT"THE POSITION ON THE KEYBOARD POINTS."
155 PRINT PRINT"THE ROBOTS AREN'T VERY BRIGHT AND WILL"
160 PRINT"CHASE YOU ACROSS A MINEFIELD!"
165 PRINT PRINT" ■ ▒ ← MINEFIELD."
170 PRINT PRINT"THEY ALWAYS MOVE HORIZONTALLY FIRST."
175 POKE245,20 PRINT PRINT"      BY LARRY HATCH"
180 SC=32768 TD=500 GOSUB 10100
210 PRINT"■  PRESS A KEY WHEN READY TO RUN"
220 POKE525,0 WAIT525,1 POKE525,0
260 PRINT"]▶▶▶▶▶▶▶▶▶■■■■■■■■ HERE THEY COME!!"
265 R=20 TD=300 GOSUB 10100
270 DIM R(20),P(20)
280 REM* RANDOM FIELD SETUP
290 PRINT"]"
300 SC=32768
310 FORI=1TO80 MF=INT(995*RND(1)) POKESC+MF,102
320 NEXT I
350 FORI=1TO20
360 R(I)=INT(988*RND(2))+12 IF R(I)=500THEN360
370 POKESC+R(I),214 P(I)=1 NEXTI
400 YU=500 SC=32768
420 POKE SC+YU,81 POKE525,0
490 REM*R=NUMBER OF ROBOTS LEFT, YS=PLAYER SCORE
495 REM* PLAYERS MOVE
500 IFR=0THENPRINT"■■YOU WIN■■        " YS=YS+1
503 IFR=0THENPOKE851,YS GOTO9220
505 PRINT"■YOUR TURN  " POKE525,0 WAIT525,1
510 GETD$ D=VAL(D$) SC=32768 YS=PEEK(851)
515 IFD=0ORD=5GOTO500
520 IFD=1THENYN=YU+39
530 IFD=2THENYN=YU+40
540 IFD=3THENYN=YU+41
550 IFD=4THENYN=YU-1
560 IFD=6THENYN=YU+1
570 IFD=7THENYN=YU-41
580 IFD=8THENYN=YU-40
590 IFD=9THENYN=YU-39
600 IFPEEK(SC+YN)<>32THENLC=YU GOTO9200
610 IFYN<0ORYN>999 THEN YN=500
630 POKESC+YN,81 POKESC+YU,32 YU=YN POKE525,0
640 REM* ROBOTS MOVES
650 PRINT"■                    " FORI=1TO20
655 IFP(I)=0 GOTO900
660 X1=R(I)-40*INT(R(I)/40) X2=YU-40*INT(YU/40) X=X2-X1
670 Y1=INT((R(I)-1)/40) Y2=INT((YU-1)/40) Y=Y2-Y1
810 J1=R(I)+SGN(X) D1=PEEK(SC+J1) POKESC+R(I),32
820 POKE SC+J1,214 R(I)=J1 J2=R(I)+40*SGN(Y) D2=PEEK(SC+J2)
825 IFX=0 GOTO845
830 IF D1=81 THENLC=SC+YU GOTO 9200
840 IF D1=102ORD1=214THENLC=SC+J1 P(I)=0 R=R-1 GOSUB9000
845 IF P(I)=0ORY=0 GOTO900
850 POKE SC+J2,214 POKESC+R(I),32 R(I)=J2
860 IFD2=81 THENLC=SC+YU GOTO9200
870 IF D2=102ORD2=214 THEN LC=SC+J2 P(I)=0 R=R-1 GOSUB9000
900 PRINT"■■"R"■ " NEXTI
910 GOTO500
1000 END
8995 REM*SPECIAL VISUAL EFFECTS
9000 FORJ=1TO10 POKELC,90 POKELC,32 NEXTJ
9010 RETURN
9200 FORJ=1TO10 POKELC,90 POKELC,32 NEXTJ
9205 POKE SC+YU,32 POKESC+YN,93 POKESC+YN-40,91
9210 RS=PEEK(850) PRINT"■TRULY A PITY" RS=RS+1 POKE850,RS
9220 POKE245,23 PRINT PRINT"HERO"PEEK(851);"ROBOTS"PEEK(850)
9240 PRINT"■■ANOTHER RUN?" POKE525,0 WAIT525,1 GET G$
9250 IF G$<>"N"THEN CLR GOTO260
10000 PRINT"GOODBYE" END
10100 FORK=1TOTD Z1=1 Z1=0 NEXTK RETURN
READY.
```

Cursor Control Characters

100	Clear Screen; RVS on; RVS off
120	RVS on; RVS off
210	Home Cursor
260	Clear Screen; 9 Right; 8 Down
500	Home Cursor; RVS on; RVS off; 1 Down
650	Home Cursor
900	Home Cursor; 2 Down
9210	Home Cursor
9240	Home Cursor; 1 Down

'FROGS' FOR THE APPLE

A Logic Game

BY EARL KEYSER

This classic game last appeared in RC in the Nov.-Dec. 1978 issue. The object of the game is to reverse two sets of objects from this starting board position,

```
X X X X   0 0 0 0
1 2 3 4 5 6 7 8 9
```

to this ending position,

```
X X X X   0 0 0 0
1 2 3 4 5 6 7 8 9
```

by making a series of slides or "hops" to an empty board space. In the starting position, the only legal moves are: slide 4 to 5, slide 6 to 5, "hop" 3 to 5, and "hop" 7 to 5.

Earl Keyser (who operates the Pilot Exchange) now brings you a version of this game for the Apple. He indicates that he also has a fancy LORES copy of this game for sale. Write to him at 22 Clover Lane, Mason City, Iowa 50401. – *RZ*

```
100 REM  EARL KEYSER'S VERSION OF FROGS FOR THE APPLE II COMPUTER.  COPYRIGH
    T 1979.
110 DIM F(9),J$(3),K$(1),A$(10)
120 FOR J=1 TO 4:F(J)=1:F(J+5)=0: NEXT J:F(5)=2
130 J$='   ':K$=' '
140 CALL -936: VTAB 10
150 INPUT 'DO YOU WANT INSTRUCTIONS?',A$: IF LEN(A$)=0 THEN 140: IF A$(
    1,1)='Y' THEN 470: GOTO 240
160 VTAB 15: FOR Z=1 TO 9: PRINT K$;: POKE 50,127: PRINT J$;: POKE 50,255
    : NEXT Z: PRINT
170 VTAB 17: FOR Z=1 TO 9: PRINT K$;: POKE 50,127: PRINT J$;: POKE 50,255
    : NEXT Z: PRINT
180 VTAB 16: PRINT ' ';
190 FOR Z=1 TO 18: POKE 50,127: IF Z=9 OR Z=10 THEN POKE 50,255: PRINT
    ' ';: POKE 50,127: IF Z<11 OR Z MOD 2=0 THEN POKE 50,255: PRINT ' '
    ;
200 NEXT Z: PRINT
210 VTAB 19: PRINT '  ';: FOR Z=1 TO 9: PRINT Z;'   ';: NEXT Z: PRINT
220 VTAB 15: TAB 18: PRINT '   ': VTAB 17: TAB 18: PRINT '   '
230 RETURN
240 CALL -936: GOSUB 160
245 FOR J=1 TO 4:F(J)=1:F(J+5)=0: NEXT J:F(5)=2
250 VTAB 5: CALL -868
260 VTAB 5: PRINT 'FROM?';
270 K= PEEK (-16384): IF K<176 OR K>185 THEN 270: IF K=176 THEN 460:OLD=
    K-176: PRINT OLD: POKE -16368,0
280 VTAB 5: TAB 15: PRINT 'TO?';
290 K= PEEK (-16384): IF K<176 OR K>185 THEN 290: IF K=176 THEN 460:P=K-
    176: PRINT P: POKE -16368,0
300 PRINT : IF F(OLD)#2 AND F(P)=2 AND OLD#P AND ABS (P-OLD)<3 AND ABS
    (P-OLD)>0 THEN 320
310 PRINT 'OOPS! ': FOR J=1 TO 1000: NEXT J: VTAB 7: CALL -868: GOTO 250
320 FOR I=0 TO 2: VTAB 15+I: TAB 4*OLD-2: PRINT J$: NEXT I
330 IF ABS (P-OLD)=2 THEN 340:Z=P*4-2: GOTO 400
340 FOR Z=OLD*4-2 TO P*4-2 STEP SGN (P-OLD)*4
350 VTAB 11: TAB Z: GOSUB 740: VTAB 12: TAB Z: GOSUB 740+(10*(F(OLD)=1)
    ): VTAB 13: TAB Z: GOSUB 740
360 FOR I=1 TO 200: NEXT I
370 FOR I=2 TO 4: VTAB 15-I: CALL -868: NEXT I
380 NEXT Z
390 Z=Z+(4* SGN (OLD-P))
400 VTAB 15: TAB Z: GOSUB 740: VTAB 16: TAB Z: GOSUB 740+(10*(F(OLD)=1)
    ): VTAB 17: TAB Z: GOSUB 740
410 M=M+1:F(P)=F(OLD):F(OLD)=2
420 FOR J=1 TO 4: IF F(J)<>0 OR F(5)#2 THEN 250: NEXT J
430 VTAB 21: PRINT 'WOW!  YOU DID IT!  IN ONLY ';M;' MOVES!'
440 VTAB 23: INPUT 'WOULD YOU LIKE TO PLAY AGAIN?',A$: IF LEN(A$)<1 THEN
    END
450 IF A$(1,1)='Y' THEN 240
460 PRINT 'THANKS FOR PLAYING MY GAME.': END
470 CALL -936: PRINT : PRINT 'WELCOME TO FROGS - THE AMPHIBIAN GAME!'
480 PRINT
490 PRINT 'THE OBJECT IS TO REVERSE THE TWO SETS'
500 PRINT 'OF SYMBOLS IN THE FEWEST MOVES.'
510 GOSUB 160
520 FOR J=1 TO 2000: NEXT J
530 VTAB 7
540 PRINT 'THE GAME ENDS WHEN THE SYMBOLS '
550 PRINT 'LOOK LIKE THIS.'
560 VTAB 16: PRINT ' ';
570 FOR Z=1 TO 18: POKE 50,127: IF Z=9 OR Z=10 THEN POKE 50,255: PRINT
    ' ';: POKE 50,127
580 IF Z>8 OR Z MOD 2=0 THEN POKE 50,255: PRINT ' ';
590 NEXT Z: POKE 50,255: PRINT : VTAB 10: FOR J=1 TO 1500: NEXT J
600 INPUT 'PRESS 'RETURN' TO CONTINUE.',A$
610 FOR J=1 TO 14: VTAB J: CALL -868: NEXT J: VTAB 2
620 PRINT 'A MOVE IS A SLIDE OF ONE SYMBOL '
630 PRINT 'TO A VACANT SQUARE OR A HOP OF ONE'
640 PRINT 'SYMBOL OVER ANOTHER TO A VACANT SQUARE.'
650 PRINT : PRINT 'NO OTHER MOVES ARE ALLOWED.': PRINT
660 PRINT 'THE NUMBERS OF THE SQUARES TO BE MOVED'
670 PRINT 'TO AND MOVED FROM ARE TYPED IN ANSWER'
680 PRINT 'TO THE QUESTIONS 'TO' AND 'FROM'.'
690 PRINT : PRINT 'HITTING THE 0 WILL END THE GAME.'
700 VTAB 21: CALL -868: INPUT 'ARE YOU READY TO PLAY?',A$
710 IF LEN(A$)=0 THEN 700
720 IF A$(1,1)='Y' THEN 240: GOTO 700
730 GOTO 240
740 POKE 50,127: PRINT J$: POKE 50,255: RETURN
750 POKE 50,127: PRINT K$;: POKE 50,255: PRINT K$;: POKE 50,127: PRINT
    K$: POKE 50,255: RETURN
```

PLAYING SIMON™ ON THE PET

Can You Repeat 99 Notes?

BY STEVE GULL

Steve is a 16-year old student from Loara High School, Anaheim, CA. He has experience on the PET, the Apple, the TRS-80, and the ADAM 31.

Steve recently got his own PET. He read the article "Making Music on the PET" by Fred Bruey in the Nov-Dec 1979 issue of RC, and sat down and created this PET version of the popular Milton-Bradley electronic game. – *RZ*

The object of this game is to repeat the musical pattern the computer makes. Each time you get a sequence correct, the program replays the sequence and adds one more note. If you miss a note in the sequence, the PET will splutter and repeat the pattern, using a different ending note. You tell the computer how many notes (1 to 99) you want to try. You also input how fast you want the PET to play the notes. If you try three times and don't get the sequence correct, a short version of "Taps" will be sounded. If you successfully get all the notes correct up to the maximum you selected, a "Charge" tune is played.

The game only uses four distinct notes. The keys 4, 5, 6, and the asterisk (*) are used in inputs. Are you ready? Listen closely!

SIMON™ is a registered trademark of the Milton-Bradley Company.

```
1 REM * SIMON *
2 REM * BY STEVE GULL - C 1979 *
5 DIM S(100),A(100),X$(100)
7 B=1
10 PRINT"'clr'QQQQQQQQQQ";TAB(14);"** SIMON **"
15 FOR W=1 to 2500:NEXT W
20 POKE 59467,16:POKE 59466,15:POKE 59464,0
30 PRINT"'clr'SIMON IS A 1 PLAYER GAME."
40 PRINT"YOU MUST FOLLOW THE PATTERN I MAKE."
50 PRINT"AND EACH TIME YOU'RE RIGHT, I'LL ADD 1"
60 PRINT"MORE NOTE TO THE SEQUENCE TILL YOU"
70 PRINT"REACH YOU'RE GOAL."
80 PRINT"BUT IF YOU MISS 3 TIMES, YOU LOSE.":PRINT
90 PRINT TAB(11);"*** GOOD LUCK ***":PRINT:PRINT
160 PRINT"WHAT IS YOU'RE GOAL (MAX. AMOUNT"
165 PRINT"OF NOTES)";
170 INPUT G:PRINT"'clr'":IF G>=1 THEN 200
180 GOTO 160
200 PRINT"WHAT'S THE DURATION OF YOUR NOTES";
210 INPUT D:PRINT"'clr'":IF D>=1 THEN 240
220 GOTO 200
240 REM * START GAME *
260 PRINT"'clr'MISSES:";M
270 PRINT" GOAL:";G
280 PRINT"NOTES MADE:";K:PRINT:PRINT:PRINT
290 PRINT"  ┌────┐ ┌────┐ ┌────┐ ┌────┐"
300 PRINT"  │    │ │    │ │    │ │    │"
310 PRINT"  │    │ │    │ │    │ │    │"
320 PRINT"  │ 4  │ │ 5  │ │ 6  │ │ *  │"
330 PRINT"  │    │ │    │ │    │ │    │"
340 PRINT"  │    │ │    │ │    │ │    │"
350 PRINT"  └────┘ └────┘ └────┘ └────┘"
355 IF B=1 THEN 360
356 RETURN
360 FOR W=1TO 2000:NEXT W:PRINT:PRINT
370 FOR X=0 TO K
375 IF X<>K THEN 430
380 IF X=K THEN R=RND(1)*20:R=INT(R/4)
390 ON R GOTO 400,405,410,415
395 GOTO 380
400 A(X)=236.8:S(X)=33134:X$(X)="4":GOTO 430
405 A(X)=117.4:S(X)=33143:X$(X)="5":GOTO 430
410 A(X)=57.7:S(X)=33152:X$(X)="6":GOTO 430
415 A(X)=27.7:S(X)=33161:X$(X)="*"
430 POKE 59464,A(X):L=S(X):A$=CHR$(160):GOSUB 450
440 FOR W=1 TO D:NEXT W:A$=CHR$(32):GOSUB 450:POKE59464,0:FOR W=1TO75:NEXTW,X
445 GOTO 480
450 POKE L-39,ASC(A$):POKE L-1,ASC(A$):POKE L-41,ASC(A$)
455 POKE L-40,ASC(A$):POKE L+40,ASC(A$)
460 POKE L+1,ASC(A$):POKE L+39,ASC(A$):POKE L+41,ASC(A$)
470 RETURN
480 PRINT"YOUR TURN...":PRINT
485 FOR W=1 TO 10:GET Z$:Z$="":NEXT W
490 FOR X=0 TO K
500 GET Z$:IF Z$="" THEN 500
510 IF Z$="4" THEN 515
511 IF Z$="5" THEN 520
512 IF Z$="6" THEN 525
513 IF Z$="*" THEN 530
514 GOTO 500
515 L=33134:J=236.8:GOTO 540
520 L=33143:J=117.4:GOTO 540
525 L=33152:J=57.7:GOTO 540
530 L=33161:J=29.7
540 POKE 59464,J
550 IF Z$=X$(X) THEN 570
560 M=M+1:PRINT"WRONG!":GOTO 620
570 FOR Y=1 TO 2
580 IF Y=1 THEN A$=CHR$(160)
585 IF Y=2 THEN A$=CHR$(32)
590 GOSUB 450
600 FOR W=1 TO 5:NEXT W,Y:POKE 59464,0:NEXT X
610 K=K+1:IF K=G THEN 680
615 B=1:GOTO 240
620 FOR W=1 TO 75:POKE 59464,255:POKE 59464,0:NEXT W
625 FOR W=1 TO 250:NEXT W
626 B=1
630 IF M<=2 THEN 240
631 B=0:GOSUB 240
635 A=0:B=150:FORY=1TO5:FORW=ATOBSTEP3:POKE 59464,W:NEXT W:A=A+25:B=B+25:NEXTY
636 POKE 59464,0:FOR W=1 TO 250:NEXT W
640 POKE 59464,236.8:FOR W=1 TO 530:NEXT W:POKE 59464,0
650 POKE 59464,236.8:FOR W=1 TO 200:NEXT W:POKE 59464,176.9
660 FOR W=1 TO 800:NEXT W:POKE 59464,0
670 PRINT:PRINT"SORRY, BUT YOU LOSE.":PRINT:GOTO 760
680 B=0:GOSUB 240
685 FOR F=1 TO 3
690 FOR W=255 TO 0 STEP-7.3:POKE 59464,W:NEXT W,F
700 FOR W=1TO5:POKE 59464,57.8:GOSUB 710:POKE 59464,0:GOSUB 710:NEXT W:GOTO715
710 FOR Y=1 TO 20:NEXT Y:RETURN
715 FOR I=1 TO 6
720 READ F:POKE 59464,F
730 FOR W=1 TO 200:NEXT W,I:GOTO 750
740 DATA 236.8,187.6,157.4,117.4,157.4,117.4
750 POKE 59464,0:PRINT:PRINT:PRINT"GREAT!!! YOU REACHED YOUR GOAL!!! ":PRINT
760 POKE 59467,0
800 END
```

Q's are Cursor Down key

Shift @ gives horizontal lines;
Shift gives vertical lines;
Shift U, J, K, and I give the rounded corners.

Continued from pg. 31

28. ..... RPxR
29. QxP R-K1
30. B-B2 .....

This is the only move that prevents Black's mate in 2. Otherwise: R-K7, QxP mate.

30. ..... R-K7
31. QxP RxP

Black is gobbling up pawns while waiting for White to make a move.

32. BxP??? .....

This move is disastrous since it allows QxP mate. Notice, however, that should Black's queen leave the back rank, White has mate in one. BxP gained a pawn, however, the computer did not foresee QxP as making it a trade. This problem has since been rectified. However, at this point, the computer is utterly lost anyway. See Figure 4.

32. ..... QxP mate

THE FINAL ANALYSIS

Many strengths and weaknesses may be seen in analyzing the computer's general strategy and basic approach toward offense and defense in this game. Among some of its best points are its rapid piece development and its concentrated attack. Its pieces were mobilized quickly and efficiently and it did not overextend itself while attacking. Weaknesses were evident when it failed to press the attack. It did not aim for the king, but rather for other pieces. Near the end, it seemed more interested in capturing pawns than preparing a mating sequence. It also made some poorly timed waiting moves, yielding the offensive advantage to Black.

Many implications of moves are seen several moves in advance by the human. Black's checkmate could be envisioned as many as 10 moves in advance. Unfortunately, the computer does not have sufficient computing power to make such advanced predictions. When a capture sequence involves pinned pieces or vital protectors, the computer has difficulty in proper analysis. Perhaps a tree search algorithm would improve this situation. In general, however, the computer played a respectable game of chess; especially impressive was its formidable initial attack.

Our program is one among many to address itself to the challenge of computer chess programming in BASIC. We have found our program to operate efficiently under the restrictions of the system available. It still is in need of some fine tuning in its analysis values as well as some additional factors for consideration that would make a mating sequence perceivable to the computer. Many more general factors need to be considered in end-strategy. In this game, the computer began using end-strategy on the 29th move. From there it tends to revert to the basic strategy of advancing pawns and capturing enemy ones.

Our program represents an attempt to diverge from the customary Shannon type I and II algorithms with tree searches. We wanted to demonstrate that a respectable chess program could be designed which had no need for tree search routines. We believe that our program shows that greater computer speed and huge storage are not necessarily commensurate with playing ability. A few refinements to our program would make it competitive with many of the current computer chess programs that rely on a tree search algorithm, and in this we feel that we have accomplished our original goal. However, trends in computer chess indicate that a different opinion is prevalent.

Current chess programs have the tendency to operate on brute force using exhaustive tree searches and extensive computing power to decide upon a move. This tendency is becoming more pronounced as machines get faster and faster. Nevertheless, it is our opinion that a computer chess program cannot be competitive on the grand-master level unless it has a sophisticated goal-oriented strategy. It is our hope that future chess programs will rely less on exhaustive continuations and more on strategic goals.

As far as chess programs for micros are concerned, the level of play will remain lower than that of the larger computers, but the programs will still rely heavily on computing speed using tree searches, a Shannon type I algorithm.

Since this is our concluding article, we would like to thank everyone for patiently bearing with the unfolding of our tale and also everyone at our old high school for putting up with us. We would especially like to thank Ms. Helen McGuigan who has supported us and our work for almost 5 years, even after our graduation from high shcool.

REFERENCES

Bach, R., "Man bites computer," *Readers Digest,* 107:151-2, Nov. 1975.

"CDC 6400 R-B4 ch; computerized chess tournament in Boston," *Newsweek,* 80:79, Aug. 28, 1972.

"Cheops plays Chess," *Scientific American,* 235:66, July 1976.

Hanver, Milton L., *Chess Made Simple.* NY: Double Day and Company Inc., 1957.

Hickey, Lisa Lane, "I felt a genuine sense of confrontation," *Think,* 37:32-5, June 1971.

Horowitz, I.A., *All About Chess,* NY: MacMillan, 1971.

"N x P!" *Scientific American,* 231:51-4, November 1974.

Newborn, Monroe, *Computer Chess.* Academic Press. 1975.

Scurrah, M.J., and D.A. Wagner, "Cognitive modell of problem solving in chess," *Science,* 169:209-11, July 10, 1970.

Steen, L.A., "Computer Chess-Mind vs. machine," *Science News,* 108:345,350, Nov. 29, 1975.

Worth, Thomas, *Basic for Everyone.* NY: Prentice Hall Inc., 1976.

Zobrist, A.L., and F.R. Carlson Jr., "Advice Taking Chess Computer," *Scientific American,* 228:92-105, June 1973.

ANNOUNCEMENTS

Educational Software

UNLIST8, BASIC Protection for BASIC programs. Automatically converts any named BASIC program (in ASCII form) saved on disk into a version that is protected; it cannot be listed or printed. Modified program can still be run, saved, and loaded as usual. $19.95, Data Associates, Box 882, Framingham, MA 01701.

Automatic Printing of Disk Programs in paged format. PAGER8, for the TRS-80, will provide program listings in a paged format ideal for manuals, publications, and for convenient editing.

PAGER8 automatically reads any named BASIC program saved in ASCII form on disk and prints it on the line printer. Instructional manual included. $19.95. Data Associates, Box 882, Framingham, MA 01701.

Course of Study for computer literacy in the classroom. Covers the essentials of programming for TRS-80, PET, Apple and others through five tapes, from Getting Started With the Computer to Programming Techniques and Flow Charts. Includes filmstrips, tapes, and handbooks, $84. Educational Activities, Inc. P.O. Box 392, Freeport, NY 11520.

Chemistry CAI Programs are 15 educational software programs for the Apple II in Applesoft BASIC. Designed to supplement the high school or junior college chemistry course. Topics from gas relationships to organic chemistry. J & S Software, 140 Reid Ave., Port Washington, NY 11050.

Apple Grade Book is a grade recording system for any teacher that records, lists, sorts, and averages student grades for the semester. It can make corrections and back-up files. On Diskette for $29.50 from J & S Software, 140 Reid Ave., Port Washington, NY 11050.

Understanding Data Communications Networks, a new educational package to instruct professionals in design and implementation of data communicating systems. 16 audio cassettes and a 650-page workbook, in a four-module format. Complete $995. Info III, 21250 Califa St., Suite 107, Woodland Hills, CA 91367.

Information Storage and Retrieval for TRS-DOS. ISAR is a data base mangement system designed to accommodate personal applications for TRSDOS random file structures. Package includes complete source listing, documentation, and suggested personal applications with sample implementation. $13.95, diskette is $16.95. The Alternate Source, 1806 Ada Street, Lansing, MI, 48910.

Advanced Inventory Control System. INV-V is for 32K TRS-80 disk systems, designed to provide control functions to maintain an efficient inventory system. It is on-line, interactive, menu-driven and human engineered. Package is $99. Micro Architect, 96 Dothan St., Arlington, MA 02174.

Hardware

Super Isolator is designed to curb those severe electrical problems that often cause memory loss, crashes, etc. It features 3 individually dual-Pi filtered 3-prong AC sockets, and can accommodate an 1875 watt load. $79.95. Electronic Specialists, Inc., 171 South Main Street, Natick, MA 01760.

TRS-80 Keyboard Modification Kit includes everything you need—wire, solder, control key, 2102 memory chip, slide switch and mounting hardware.

The documentation is detailed and complete, with many pictures and diagrams. Emmanuel B. Garcia, Jr. & Associates, 3950 N. Lake Shore Dr., Rm. 2310, Chicago, IL 60613. $19.95.

RomWriter™ is an EPROM programmer designed to permit the Apple Computer owner to program 2K 2716 (5V) EPROMs. Diskette based software included with RomWriter permits virtually foolproof programming. $159. Mountain Hardware, Inc., 300 Harvey West Blvd., Santa Cruz, CA 95060.

Tape Punch for PET, as an accessory with typesetting system or a peripheral for mini and micro systems for numerical control, typesetting, and other applications. Punch operates at up to 50 characters a second. Type-Share, 8315 Firestone Blvd., Downey, CA 90241.

Skyles MacroTeA is a relocatable, conditional macro, text editor, monitor that allows you to program right into the 6502 heart of the PET. No disc system; it needs only 8K of RAM. $395. Skyles Electric Works, 10301 Stonydale Dr., Cupertino, CA 95014.

VersaWriter, a digitizer and software drawing package for the Apple II Computer that provides high resolution, mass color graphics comparable to the quality of the Apple. $179.95. Rainbow Computing, Inc., 9719 Reseda Blvd., Northridge, CA 91324.

User Groups

Eastern Iowa Computer Club covers a variety of interests, uses, computers, and topics in monthly meetings. Contact EICC, Box 164, Hiawatha, IA 52233.

Superboard Club for OSI, CIP and Superboards. Updated information on your computer in monthly newsletter, programs, ideas, technical data and more. Superboard Club, Box 55, Agincourt, Ontario, Canada M1S 3B4.

South Carolina Apple, a new apple computer users group, meets on the second Tuesday of each month at 7:30 pm at the Byte Shop, 1920 Blossom Street in Columbia, SC. For information contact South Carolina Apple, Felix Clayton, President, 1610 Longview Road, Mount Pleasant, SC 29464.

Other

Call for Papers for the Fourth Western Educational Computing Conference being held in San Diego on November 20 & 21, 1980, by the California Educational Computing Consortium. Ask for specs.

Send original abstracts with two copies by March 30, 1980 to Professor Virginia S. Lashley, Glendale College, 1500 N. Verdugo Road, Glendale, CA 91208.

MICRO/EXPO 80, the leading European show on microcomputers, will be held in Paris on May 6-8, 1980. Participants include leaders from industry, education and the press. Attendance is expected to exceed 10,000. Micro/Expo 80, 2020 Milvia St., Berkeley, CA 94704.

First World Conference on Transborder Data Flow Policies to be held in Rome Italy, 23-27 June, 1980. The growth of data networks operated by transnational data users, time sharing services, carriers, etc. to be covered. Intergovernmental Bureau for Informatics, P.O. Box 10253, 23, viale Civilta del Laboro 00144 Roma, Italy.

New Educational Computing Consulting Service to help teachers and administrators use low-cost microcomputer technology. Determine instructional computer needs; locate, design and evaluate courseware and facilities; conduct training programs; prepare applications for grants, and purchase equipment. Logics One, Box 41, Utica, MI 48087.

Computer Camp, a unique recreational and educational experience for youngsters, ages 10-17, directed by Dr. Michael Zabinski, Professor at Fairfield University. Action packed week from June 29 to July 4, with small group instruction on mini and micro computers. Computer Camp, Grand View Lodge, Box 22, Moodus, CT, 06469.

Apple Educators' Newsletter on events, developments and software for the Apple computer in education. Minnesota Educ. Computing Consortium, Conduit in Iowa, bibliographies, advertisement, software distribution, letters and articles. 12 issues $12. AEN, 9525 Lucerne, Ventura, CA 93003.

Personal Computer NETwork (PCNET), started out of People's Computer Company, offers personal computer-based electronic mail software. A unique grass-roots hardware/software project. For more information contact PCNET, 1263 El Camino Real, Box E, Menlo Park, CA 94025.

SOFTWARE

PROGRAM INSTRUCTION BUILDER

Saves Time and Typing

BY MARC B. STANIS

Here is a simple utility routine created by a senior at Bremen H.S. in Midlothian, Illinois. This program helps you put the set of instructions on a game program, or any program, in a hurry. It takes some of the drudgery out of the task, and leaves you more time for creating other games and diversions. —*RZ*

I am presently enrolled in Computer Concepts III, computer science course where I am learning assembly language programming. The computer is a DEC PDP/8A with CRT, two printers, two disks, and a mark sense card reader.

I had just completed one of the game programs I had been working on, and needed to write the instructions for it. (The program simulates a race. It allows up to nine racers.) I decided to write a program that would make the job of writing a set of instructions a lot easier. The program would have to generate the line numbers, put a set of quotes around the inputted line, and put a print statement in front of it. All the user would have to do is sit in front of the CRT and type in the instructions.

The first thing the program requests is the device and file name into which the instructions are to be stored. The program then asks for a command. There are two choices:

1. Start a new set of instructions. To get this option, simply type '1' to the input prompt.
2. Continue an already started (old) set of instructions. To get this option, simply type '2' to the input prompt. The program will now delete the 'close' and 'end' statements which are at the end of the old set of instructions.

Instruction Builder Listing

```
100 REM
110 REM
120 REM                 -----   INSTRUCTION BUILDER   -----
130 REM                 -----    MARC  B.  STANIS     -----
140 REM                 -----    JANUARY  10, 1979.   -----
150 REM
160 REM
170 REM                 BREMEN   H.S.            MIDLOTHIAN
180 REM
190 REM
200 DIM A$(72),F$(10)
210 PRINT 'PLEASE INPUT THE FILE NAME';\INPUT F$\REM STORES THE FILE NAME
220 PRINT 'RXA0:   OR   RXA1:';\INPUT D$\REM STORES THE DISK DRIVE
230 PRINT ' 1= START NEW INSTRUCTIONS   2= CON. ON OLD INSTR.';
240 INPUT O\IF O=1 THEN 310\REM START NEW INSTRUCTIONS?
250 IF O<>2 THEN 230\REM PROPER INPUT
260 FILE#2: D$ & F$\FILEV#1: D$ & F$\REM OPEN PROPER FILES
270 INPUT#2: A$\IF POS(A$,'CLOSE#N',1)>0 THEN 300\PRINT#1: A$
280 PRINT A$\REM OUTPUT TO THE SCREEN WHAT PART OF THE INSTR. HAVE BEEN
290 X=VAL(SEG$(A$,1,5))+10\GOTO 270\REM KEEP TRACK OF LINE NO.
300 CLOSE#2\PRINT 'TYPE <CR> TO CONTINUE.'\INPUT Z$\GOTO 350
310 FILEV#1: D$ & F$\REM OPENS THE FILE
320 PRINT 'AT WHICH LINE NUMBER DO YOU WISH TO BEGIN AT';\INPUT X
330 X=X-10\REM SET UP TO START CYCLE
340 X=ABS(INT(X))\REM BE SURE LINE NO. IS A POS. INTEGER
350 PRINT PNT(27);'H';PNT(27);'J';\REM CLEAR SCREEN AND HOME CURSOR
360 FOR Z=1 TO 12\PRINT TAB(57);'* '*STOP''\NEXT Z\REM PRINTS THE BOR.
370 PRINT PNT(27);'H'\REM HOMES THE CURSOR (UPPER RIGHT HAND CORNER)
380 INPUT A$\REM INPUTS LINE TO BE CONVERTED
390 IF LEN(A$)<56 THEN 420\REM CHECK TO SEE IF LINE IS TOO LONG
400 PRINT 'THAT LINE WAS TO LONG, PLEASE RE-INPUT'
410 GOTO 490
420 IF A$='*STOP' THEN 510\REM TIME TO STOP?
430 IF A$='' THEN 450\REM SKIP PUTTING QUOTES AROUND IT
440 A$=CHR$(34) & A$ & CHR$(34)\REM PUT A SET OF QUOTES AROUND THE LINE
450 X=X+10\REM LINE NO. OF THE NEXT LINE OF THE INSTRUCTIONS PROGRAM
460 REM   NOW PUT THE LINE NO. AND ' PRINT#N: ' IN FRONT OF THE INPUTTED
470 REM     LINE, AND THEN STORE IT
480 PRINT#1: STR$(X) & ' PRINT#N: ' & A$
490 PRINT TAB(57);'*';PNT(13);\REM PRINTS THE BORDER, CAR. RET.
500 GOTO 380
510 PRINT#1: STR$(X+10) & ' CLOSE#N'
520 PRINT#1: STR$(X+20) & ' END'
530 CLOSE#1\REM CLOSE THE FILE WHICH STORES NEW PROGRAM
540 PRINT PNT(7)\REM SOUNDS THE TERMINAL BUZZER
550 PRINT 'DON'T FORGET TO FINSIH UP THE INSTRUCTION PROGRAM'
560 PRINT 'REMEMBER . . . . . . . . . .';D$;F$
570 PRINT 'DO YOU WISH TO RUN THIS PROGRAM AGAIN? (N/Y)';\INPUT Z$
580 IF Z$='Y' THEN 210\REM RUN THE PROGRAM AGAIN?
590 REM
600 REM
610 REM
620 REM
630 REM                 *****   VARIABLE CHART   *****
640 REM
650 REM
660 REM
670 REM        O              OPTION SWITCH
680 REM                   1 = START NEW INSTRUCTIONS
690 REM                   2 = CONTINUE OLD INSTRUCTIONS
700 REM        X              USED TO STORE WHAT LINE NO. WILL BE IN
710 REM                   FRONT OF THE NEW LINE OF THE NEW PROGRAM
720 REM        A$             STORES YOUR INPUTTED LINE, UNTIL IT IS
730 REM                   CHANGED AND STORED ON THE NEW PROGRAM
740 REM        D$             STORES WHICH DISK DRIVE THE NEW PROGRAM WILL
750 REM                   BE STORED ON
760 REM        F$             STORES WHICH FILE NAME THE NEW PROGRAM WILL
770 REM                   BE STORED UNDER
780 REM        Z$             DUMMY VARIABLE -- USED FOR Y N QUESTIONS
790 REM
800 REM
810 REM
820 REM        PNT(13);    IS A CURSUR FUNCTION, WHICH CAUSES CARRIDGE
830 REM                   RETURN, WITHOUT LINE FEED
840 REM
850 REM
860 END
```

NOTE: If neither '1' or '2' is typed to the input prompt, the question will be asked again.

The screen will now clear, and a border will be printed on the right hand side. The distance the border is to the right is determined by the maximum length a line of code is allowed to be. (In this case, that number is 72 characters, with up to five digits for a line number.) The user now types in the instructions. When the user wishes to stop typing, '*STOP' must be entered to the input prompt. The program will now close out all files used and ask if you wish to run this program again. If the answer is yes, type 'Y' to the input prompt. If anything else is typed, the run will be terminated.

Lines 320 to 480 contain the heart of the program. The rest of the code deals mostly with the file handling on the PDP. For those of you wanting to put the Program Instruction Builder on your machine, concentrate on lines 320-480, and set up files as you would for your computer.

Example Output

```
1 REM
2 REM
3 REM          -----   RACER9  INSTRUCTIONS   -----
4 REM          -----     MARC  B.  STANIS     -----
5 REM          -----     JAN.  17,   1978     -----
6 REM
7 REM
8 REM           BREMEN   H.S.             MIDLOTHIAN
9 REM
10 PRINT 'HOW DO YOU WISH YOUR INSTRUCTIONS:'
20 PRINT '(1=PRINTER      2=SCREEN)'
30 INPUT N
40 IF N=1 THEN 60
50 IF N<>2 THEN10
55 FILEV#2:'TTY:'\GOTO 110
60 FILEV#1:'LPT:'
110 PRINT#N: '                   *** RACER 9 ***'
120 PRINT#N:
130 PRINT#N:
140 PRINT#N: '    THE OBJECT OF THIS GAME IS TO RACE YOUR CAR AROUND'
150 PRINT#N: 'THE RACE TRACK (COUNTER CLOCKWISE) UNTIL YOU CROSS'
160 PRINT#N: 'THE FINISH LINE. YOU TRY TO DO THIS IN THE QUICKEST NO.'
170 PRINT#N: 'OF TURNS POSSIBLE.'
180 PRINT#N:
190 PRINT#N:
200 PRINT#N: '    WHEN THE GAME STARTS, YOUR CAR WILL BE THE SAME NO.'
210 PRINT#N: 'RACER THAT YOU ARE. EX. IF YOU ARE RACER NO. 4, THEN'
220 PRINT#N: 'YOUR CAR WILL BE NO. 4. YOU WILL NOTICE THAT IN THE BE-'
230 PRINT#N: 'GINING OF THE GAME THE ONLY CAR THAT IS VISIBLE, IS CAR'
240 PRINT#N: 'NO. 1, THAT IS BECAUSE, THE OTHER CARS ARE UNDER IT. AS'
250 PRINT#N: 'SOON AS THE FIRST CAR MOVES, THE SECOND CAR WILL APPAER'
260 PRINT#N: 'AND SO ON UNTIL ALL OF THE CARS ARE OUT. ONCE'
270 PRINT#N: 'ALL OF THE CARS ARE OUT, IT IS IMPORTANT NOT TO HIT'
280 PRINT#N: 'THEM. IF YOU DO YOU WILL BE PENALIZED.'
290 PRINT#N:
300 PRINT#N: '**** DIRECTIONS *****'
310 PRINT#N: '    THERE ARE 4 DIRECTIONS IN WHICH YOU MAY MOVE YOUR'
320 PRINT#N: 'CAR AROUND THE TRACK. THESE FOUR DIRECTIONS WILL BE'
330 PRINT#N: 'REPRESENTED BY THE NO. 1-2-3-4.'
340 PRINT#N: 'DIRECTION  NO. 1  IS TOWARD THE  TOP'
350 PRINT#N: 'DIRECTION  NO. 2  IS TOWARD THE  RIGHT'
360 PRINT#N: 'DIRECTION  NO. 3  IS TOWARD THE  BOTTOM'
370 PRINT#N: 'DIRECTION  NO. 4  IS TOWARD THE  LEFT'
380 PRINT#N:
390 PRINT#N: '***** SPEEDS *****'
400 PRINT#N: 'THERE ARE THREE BASIC SPEEDS  1-2-3'
410 PRINT#N: 'NO. 1   IS HALF OF YOUR PRESENT SPEED (IF YOUR SPEED'
420 PRINT#N: 'IS ONE, YOU WILL GO TO ZERO, ISTEAD OF ONE/HALF.'
430 PRINT#N: 'NO. 2   IS THE SAME SPEED'
440 PRINT#N: 'NO. 3   IS TWICE YOUR PRESENT SPEED  (IF YOUR SPEED IS'
450 PRINT#N: 'ZERO, IT WILL BRING YOU UP TO THE SPEED OF ONE.'
460 PRINT#N:
470 PRINT#N: '    ON THE SCREEN YOU WILL NOTICE THE TRACK, WITH THE'
480 PRINT#N: 'CARS IN THIER PRESENT POS., AND AT THE BOTTOM WILL '
490 PRINT#N: 'BE THE NAME OF WHOSEVER TURN IT IS. NEXT TO THE NAME,'
500 PRINT#N: 'WILL BE HIS PRESENT SPEED, AND THEN IT WILL WAIT FOR'
510 PRINT#N: 'YOU TO INPUT YOU DIRECTION AND SPEED.'
520 PRINT#N: '    WHEN YOU INPUT YOUR TURN IT MUST BE IN THE FOLLOW-'
530 PRINT#N: 'ING FORMAT: FIRST THE DIRECTION, THEN A COMMA, AND THEN'
540 PRINT#N: 'THE SPEED YOU WISH TO GO. IF YOUR INPUT IS INCOR-'
560 PRINT#N: 'RECT, THE COMPUTER WILL ASK YOU AGAIN FOR YOUR MOVE.'
570 PRINT#N:
580 PRINT#N: '***** ACCIDENT *****'
590 PRINT#N: '    IN THE EVENT THAT YOU RUN INTO ANOTHER RACER, OR THE'
600 PRINT#N: 'WALL OF THE RACETRACK, YOU WILL BE PENALIZED. THE FAST-'
610 PRINT#N: 'ER YOU ARE GOING WHEN YOU CRASH, THE LONGER THE WAIT'
620 PRINT#N: 'UNTIL YOUR CAR IS REPAIRED. IF YOU CRASH, THE INTO ANOTHER
630 PRINT#N: 'RACER, BOTH CARS WILL BE PENALIZED, HOWEVER THE CAR'
640 PRINT#N: 'WHO WAS AT FAULT WILL HAVE TO WAIT LONGER FOR HIS CAR'
645 PRINT#N: 'TO BE FULLY REPAIRED.'
650 PRINT#N:
670 PRINT#N: '***** END OF GAME *****'
680 PRINT#N: '    THE END OF THE GAME IS REACHED WHEN ALL OF THE'
690 PRINT#N: 'RACERS HAVE CROSSED THE FINISH LINE. ONCE THIS IS DONE'
700 PRINT#N: 'THE COMPUTER WILL DISPLAY THE ORDER IN WHICH THE RACERS'
710 PRINT#N: 'CAME IN, AND ON THE PRINTER WILL COME A PRINT-OUT OF'
720 PRINT#N: 'HOW YOU DID DURING THE RACE.'
730 PRINT#N:
740 PRINT#N: '***** EXTRA OPTIONS *****'
750 PRINT#N: '    THERE ARE THREE EXTRA OPTIONS:'
760 PRINT#N: 'NO. 10  IF YOU WISH TO RESIGN DURING THE GAME'
770 PRINT#N: 'NO. 13  IF YOU WISH TO SAVE A GAME (IF YOUR TIME IS AL-'
780 PRINT#N: '      MOST UP, AND YOU WISH THE CONTINUE THIS GAME ON'
790 PRINT#N: '      SOME OTHER DAY, USE THIS OPTION)'
800 PRINT#N: 'NO. 15 IF YOU WISH TO RECALL A SAVED GAME'
810 PRINT#N: 'NOTE: IF YOU WISH TO CALL UP ANY OF THESE OPTIONS YOU'
820 PRINT#N: '      MUST INPUT THE OPTION NO. INRLACE OF YOUR SPEED'
830 PRINT#N: '      BE SURE THAT YOU HAVE A LEGAL DIRECTION.'
840 PRINT#N:
850 PRINT#N: 'GOOD LUCK...'
860 PRINT#N:
870 PRINT#N:
880 CLOSE#N
890 END
```

SOFTWARE

DATA RETRIEVAL: An Introduction

A Simple TRS-80 Search Module

BY RAMON ZAMORA

This article shows you how to create a beginning program that will adequately search and retrieve data from small files. The material is excerpted from a forthcoming book: TRS-80: A Self Teaching Guide, *by Albrecht, Inman, and Zamora, © 1980, John Wiley & Sons, Inc.*

The program that is developed works on data records containing string information. The particular application discussed, a directory search, could be generalized to a number of other areas: inventories, business accounts, or computer club records. The program is written in Level II BASIC for the TRS-80. —*RZ*

THE DATA FILE FIRST

The program that follows will search a file of data containing the names, addresses, and telephone numbers of individuals, and extract the records that match a search key that the user specifies.

The file, in this case, is contained in DATA statements within the program. The routine could be extended for use with a larger data base that is perhaps stored on tape or disk, but the demonstration program only uses DATA statements.

In fact, to keep things really simple, only two data records are used. The records will contain the names, addresses, and telephone numbers for two people:

Jack Jones 1511 Westport Dr. Menlo Park, CA 94025 415/555-0707
Jane Jonarts 1521 Westbury St. Menlo Pk., Calif. 94025 415/555-1975

We have deliberately chosen two data records that are similar in many respects so that we can demonstrate the search capabilities of the program.

The program searches each data record by: first name, last name, street address, street name, city, state, zip code, or area code. The search key can be from one to several letters long. If an exact match is needed, more letters can be used. If fewer letters are used, all records that match the search key are displayed. Sounds like a lot of program, doesn't it? Actually, the program is quite simple.

THE PROGRAM

Here is the program for a computerized directory search:

```
10    REM** COMPUTERIZED DIRECTORY SEARCH **
20    DIM N1$(10),N2$(10),A1$(10),A2$(10),C$(910) S$(10),Z$(10),T$(10)
30    CLS: INPUT "NUMBER OF DATA ITEMS: ";D: GOSUB 1000
40    CLS: PRINT "COMPUTERIZED DIRECTORY"
50    PRINT STRING$(22,"-");
60    PRINT @256, "WHAT DO YOU WANT TO DO? SEARCH BY:"
70    PRINT @320, " 1-FIRST NAME!"
80    PRINT @384, " 2-LAST NAME!"
90    PRINT @448, " 3-STREET ADDRESS"
100   PRINT @512, " 4-STREET NAME"
110   PRINT @576, " 5-CITY"
120   PRINT @640, " 6-STATE"
130   PRINT @704, " 7-STATE"
140   PRINT @768, " 8-AREA CODE"
150   PRINT @832, " 9-STOP THE PROGRAM"
160   K$=@ INKEY$: IF K$="THEN 160 ELSE K = VAL(K$)
170   IF K = 9 THEN END
180   IF K = THEN 160
190   CLS: INPUT "ENTER THE SEARCH KEY: ";K$: P = 0
200   FOR N = 1 TO D: ON K GOTO 210, 220, 230, 240, 250, 260, 270, 280
210   IF LEFT$(N1$(N),LEN(K$)) = K$ THEN 290 ELSE 320
220   IF LEFT$(N2$(N),LEN(K$)) = K$ THEN 290 ELSE 320
230   IF LEFT$(A1$(N),LEN(K$)) = K$ THEN 290 ELSE 320
240   IF LEFT$(A2$(N),LEN(K$)) = THEN 290 ELSE 320
250   IF LEFT$(C$(N),LEN(K$)) = K$ THEN 290 ELSE 320
260   IF LEFT$(S$(N),LEN(K$)) = THEN 290 ELSE 320
270   IF LEFT$(Z$(N),LEN(K$)) = K$ THEN 290 ELSE 320
280   IF LEFT$(S$(N),LEN(K$)) = THEN 290 ELSE 320
290   PRINT @ 256+P*64,N1$(N)+" "+N2$(N)+" "+N2$(N);TAB(25);A1$(N)+" "+A2$(N)
300   PRINT @ 256+(P + 1)*64,C$(N)+" "+S$(N)+ " "+Z$(N);TAB(40);T$(N)
310   P = P + 2
320   NEXT N
330   PRINT @780, "HIT ANY KEY TO CONTINUE:"
340   K$ = INKEY$: IF K$ = "" THEN 340 ELSE 40

1000  REM** DATA INPUT ROUTINE **
1010  FOR N = 1 TO D
1020  READ N1$(N),N2$(N),A1$(N),A2$(N),C$(N),S$(N),Z$(N),T$(N)
1030  NEXT N: RETURN
1040  DATA "JACK", "JONES", "1511", "WESTPORT DR.", "MENLO PARK", "CA", "94025", "415/555-0707"
1050  DATA "JANE", "JONARTS", "1521", "WESTBURY ST.", "MENLO PK.", "CALIF", "415/555-1975"
```

Before the program is entered and RUN, let's examine the key parts of the routine.

Line 20: Explicitly dimensions the data record elements. The order of the variables is the order that the elements appear in the data records. The arrays will contain: N1$, first names; N2$, last names; A1$, street numbers; A2$, street names; C$, cities; S$, states; Z$, zip codes; T$, telephone numbers.

Line 30: Requests number of data items (D), and calls the data input routine (Line 1000).

Lines 40-150: Presents "menu" of selections.

Lines 160-180: Accepts "menu" selection, and validates entry.

Line 190: Gets the search "key." The variable P will be used later to position the output on the screen.

Line 200: Begins the loop through the data file. A branch is made to the appropriate test statement (Lines 210-280) where the search "key" is checked against the data item.

Lines 210-280: Tests for match of data with search "key."

Lines 290-300: The entire record is displayed.

Line 310: Increments the screen position index.

Line 320: End of the search loop.

Lines 330-340: Pause to view output.

Lines 1000-1030: Subroutine to READ the data.

Lines 1040-1050: The two example data records.

If the program is entered and RUN, the first thing to show on the screen will be a request for the number of data records. Entering a two (2) to this request, will cause the two data records to be READ into the appropriate arrays. The screen will then clear, and the "menu" will appear.

RUNNING THE PROGRAM

The "menu" will look like this:

```
WHAT DO YOU WANT TO DO? SEARCH BY:

1- FIRST NAME
2- LAST NAME
3- STREET ADDRESS
4- STREET NAME
5- CITY
6- STATE
7- ZIP CODE
8- AREA CODE
9- STOP THE PROGRAM
```

The program then waits for an input of a number from 1 to 9. When a number is entered, the program then requests a search key. The key can be from one to as many characters as are needed.

Let's say we select menu item 1, a search by first name, and enter the letter J for the search key. The screen will show:

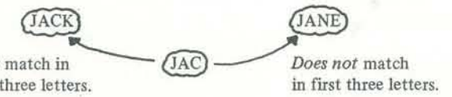
```
ENTER THE SEARCH KEY: J

JACK JONES                1511 WESTPORT DR.
MENLO PARK CA 94025             415/555-0707
JANE JONARTS              1521 WESTBURY ST.
MENLO PK. CALIF 94025           415/555-1975

                HIT ANY KEY TO CONTINUE:
```


Both records are displayed since both first names begin with the letter J. Pressing any key will cause the "menu" to reappear. What will appear on the screen if a search by first name is made with the key JAC?

If menu selection one (1), a search by first name, and the key JAC is entered the screen shows:

```
ENTER THE SEARCH KEY: JAC

JACK JONES                1511 WESTPORT DR.
MENLO PARK CA 94025             415/555-0707

                HIT ANY KEY TO CONTINUE:
```

Why does this occur? Well, the search key is three letters long, and only matches exactly with the first three letters of Mr. Jones' first name. The key does not match Ms. Jonarts' first name in the first three positions. So only the record for Mr. Jones is displayed.



If the search key had been JA, both records would have been put on the screen. Let's examine line 210 of the program where this matching operation is taking place.

```
210 IF LEFT$(N1$(N),LEN(K$))=K$ THEN 290 ELSE 320
```

The search key is in K$. N1$(N) contains the first name of the Nth record being examined. The function LEFT$ extracts the leftmost characters from N1$(N) based on the length, LEN, of K$. That is, the number of characters looked at in N1$(N) is determined by the length of the search key. All the other searches are carried out in the same way in lines 220-270.

THE SEARCH IS OVER

When you give this program a try, experiment with various menu selections, and use different search keys. Notice how quickly the TRS-80 is able to scan the records and display the results of the search.

Can you think of a way to *really* simplify this program? Hint: Change the arrays (N1$, N2$, and so on . . .) that are being used to one array!! If one data array, say D$, is used with dimensions D$(8, 10), lines 210-270 reduce to a single line in the program:

```
IF LEFT$(D$(K,N),LEN(K$))=K$ THEN 290 ELSE 320
```

Amazing!! The entire search and display section reduces to just a few lines of BASIC! What else could you do with this program? The techniques shown here would work on any data file. Instead of names and addresses, you might have a parts inventory, a list of business accounts, or your computer club dues records. Let us hear from you on what your computer searches turn up.

PROGRAMMING PROBLEMS: SOLUTIONS

PROBLEM # 1

Here are two solutions to Problem # 1 (Positive, Negative, or Zero). They were written on a "bare-bones" TRS-80, Level I, 4K.

A. Works for all real numbers in the Basic's range:

```
10 INPUT "YOUR NUMBER"; N
20 ON 1*(N>0) + 2*(N=0) + 3*(N<0) GOTO 30, 40, 50
30 PRINT "NEGATIVE": GOTO 10
40 PRINT "ZERO": GOTO 10
50 PRINT "POSITIVE": GOTO 10
```

B. Works for integers with absolute values that are within the limits imposed upon the subject of an ON . . . GOTO . . . statement.

```
10 INPUT "YOUR (INTEGER) NUMBER"; N
20 ON ABS(N) + 1 GOTO 70
30 ON ABS(N)/N + 2 GOTO 50, 40, 60
40 PRINT "ERROR": STOP
50 PRINT "NEGATIVE": GOTO 10
60 PRINT "POSITIVE": GOTO 10
70 PRINT "ZERO": GOTO 10
```

Charles M. Somerville
5476 Broomall Street
Dayton, OH 45424

Note that the range allowed is $-32768 < N < 32768$ with the exception of the two specific numbers, -32000 and $+32000$. This latter is due to a bug in level I which treats "ON N GOTO . . . " as "ON N - 32000 GOTO . . . " for all $N > 32000$.

Here are two PET solutions to the Problem # 1. The second solution is *two* lines of code.

A. I assume this is the solution for which a hint was given.

```
10 REM***PROBLEM#1 POSITIVE NEGATIVE OR ZERO
20 REM***RECREATIONAL COMPUTING 1/80
30 REM***SUBMITTED BY LEN LINDSAY
40 REM
200 INPUT"NUMBER, PLEASE";N : REM ASK FOR NUMBER
300 PRINT"IT IS ";: REM PRINT START OF REPLY
400 ON SGN(N)+2 GOSUB 500, 600, 700 : REM GOTO
    CORRECT REPLY
410 GOTO 200 : REM GET NEXT NUMBER
500 PRINT "NEGATIVE" : RETURN
600 PRINT "ZERO" : RETURN
700 PRINT "POSITIVE" : RETURN
```

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

B. Two line solution.

```
10 REM***PROBLEM#1 POSITIVE NEGATIVE OR ZERO
20 REM***RECREATIONAL COMPUTING 1/80
30 REM***SUBMITTED BY LEN LINDSAY
40 REM
100 M$(1) = "NEGATIVE" : M$(2) = "ZERO" : M$(3) =
    "POSITIVE" : REM ASSIGN 3 REPLIES
200 INPUT "NUMBER, PLEASE" ; N : PRINT "IT IS " ;
    M$(SGN(N) + 2) : GOTO 200
```

LETTERS

ABOUT ADVENTURE

Over the past few months I've read in *Recreational Computing* and elsewhere about a game referred to as "Adventure." I assume that these references are to the now-famous main-frame game written, I believe, by Wil Crowther. If so, the game originally ran on a full-house IBM 360 using about 1,100 FORTRAN statements plus text plus data. About a year ago it was made available to Heath H8 owners in a disk version running in 24K. This version was distributed to members of the Heath Users' Group (HUG) for $10.00, and ran exactly the same as the large version, so far as I recall. Truly remarkable for an 8080-based system without IBM's megabytes of memory. Not much later, a tape version running first in 32K, but later available for 24K systems, was made available by Eggert Engineering, 95 Adams Drive, Stow, MA 01775 for $16.50. I understand a paper tape version is available for those not blessed with a Heath H8/H89 system. The point I wanted to make is that this is not a watered-down version or a pale copy as some of the more recent issues appear to be.

For anyone interested, there is a medium-sized computer club in the Blacksburg, VA, area known as the New River Computer Club. Drop me a line if you're interested. The organization operates TRS-80s, Heath equipment and home-brew, with bags of software and hardware expertise.

Thanks for listening.

D.C. Shoemaker
2000 A Foxridge
Blacksburg, VA 24060

Adventure International, (Box 3435, Longwood, FL 32750), has several versions of the fantasy game for 16K TRS-80 machines. Each adventure has a vocabulary of 150-200 words. We have observed kids playing these games for hours! These versions are remarkable, given the size of machine for which they are configured. – *Eds.*

AN APPEAL FROM YOUTH

I never want to stop getting your fine magazine. I especially like the articles that speculate on new uses for computers. Please enter me as a *retaining subscriber* and add the appropriate number of issues to my subscription.

I am a high school junior, and I have a Model I Level II TRS-80 with a disk, line printer and voice synthesizer. I want to share my enjoyment with fellow students. There are about 2500 students at present and the town is growing rapidly, so I think it would be a good idea if we got something started in the computer field. The math and science teachers at my school are gathering information to present to the school board to explain why we should get one or two small computers. I would appreciate any suggestions you and your readers could offer to help our cause.

THANK YOU!

William M. Richman II
Hill Top Heights
North Platte, NB 69101

TOLKIEN AGAIN

Once again I write regarding the so-called "Tolkien Debate" that has been running for two issues since I last made a few remarks concerning your attempts to get into the film critiquing business. *That*, after all, was what I was writing about in the first place! But it seems that *RC*'s (Michael Madaj's) continued correspondence in this matter disregards the fact that I make no reference to having read the Tolkien books. I merely went to a movie, found it wanting, and upbraided the author of the article for having no discernment between good films and the other kind.

But your readership, as well as several heated letters I have received personally (all of which run to *undue* length), completely ignore the content of the original correspondence. The writers of the so-called "Tolkien Debate" (which *is* no debate but a diatribe) are saying, "Lon, you are a cad for not liking sweet old Mr. Tolkien's stories and we pity you for not being able to appreciate them. How could you say such things when *we*, the *cognoscenti*, can quote you chapter and verse about this or that or the other thing."

Well, a careful reading of my original letter will reveal that *Mr. Bakshi's* film, not dear old Mr. Tolkien's books, took the worst of it! I am in complete agreement with C.A. Cozart (*Letters, RC* Nov.-Dec.) that movies and literature have no place in a computing magazine.

The point of the letter was that I liked some things about *RC* regarding the concept of the game *Universe*, and I did not like the praise of a film which could only be enjoyed by the most knuckle-headed of individuals.

I mentioned elsewhere that I thought the *Universe* article by Les LaZar (*RC* March-April) was the best that I had seen. Well, I also get a number of other computing magazines, and I thought the readership of *RC* should be aware of an article in *kilobaud MICROCOMPUTING* (July 1979). The article describes a vocational school in Carolina which has a fully integrated microcomputing instruction course using the whole set of Heath Continuing Education courses as text for preparation for careers in the field. I think that an in-depth article for *RC* on this school would be most appreciated by the educators among your readers.

Lon Ponschock
203 S. Douglas St.
Appleton, WI 54911

TOWER OF HANOI CORRECTION

The Tower of Hanoi program by Herbert Dersham is fun to use and to watch (*RC*, Nov.-Dec. 1979, p. 35). In keying it in from the listing, I found several minor corrections were necessary in order for the program to run properly. These are:

```
120 ON R GOTO 300, 130, 1000
(THE TEXT READ "100")

210 FOR I=1 to 100 PRINT@896,
    "(APPLAUSE!! . . .
(THE @ SIGN WAS MISSING)

220 FOR J=1 TO 100 NEXT J:PRINT
    @896, CHR$(30);
(CHANGE THE S TO $)

440 THIS LINE SHOULD BEGIN: IF
(MISSING)

490 CHANGE 10000 TO 1000
(SAVES FRUSTRATION)

6090 DELETE THE PERIOD FROM
     THE END OF THE LINE
(PRINTS AN UNNECESSARY ZERO
ON THE SCREEN)
```

There is one other small (?) bug still left in the program which I have been unable to find. However, it does not affect the running of the program. (The number of discs called for at the start of the program on Tower 1 is one less than it ought to be. The proper number of discs are in play.)

Perhaps some sharp-eyed reader can find this still hidden (to me) bug!

I have enjoyed the programs in your recent issues and find them both recreational and practical.

Philip M. Reidy
332 Main Street
Worcester, MA 01608

AN IMPROVEMENT ON CHOOSE-A-TITLE

I read the Nov.-Dec. 1979 issue of *RC* with interest and tried my hand with the Choose-a-Title program, using insurance terms and jargon instead of music. All very amusing and it ran fine on my TRS-80 except for the Rnd(-1) statements which were easily corrected. However, when I started adding more words, it was not just a simple matter of changing the Rnd statements. I started getting "out of data" errors and soon realized that the formulas had to be changed as well. If line 220 is changed, then line 320 must also be changed. Here is what seems to me to be a simpler solution which has the advantage of keeping the lines that must be changed right next to each other. For example:

```
220 R1=Rnd(13)
230 For Z=1 to 13
240 Read Z$
250 If Z=R1 then A$=Z$
260 Next Z
      etc.
```

Thus, if you add 2 words to A$ data, simply change the 13s in 220 and 230 to 15s. Or, how about the possibility of defining A, B, C, D, and E at the beginning of the program and using these variables instead of the numbers? Then only 1 number must be changed.

Thus:

```
10 A=13:B=15 etc.
220 R1=Rnd(A)
230 For Z=1 to A
      etc.
```

No big thing, but this one is even easier to change.

Keep up the good work.

Jerry Rutledge
Box 123
Waseca, MN 56093

GOOD INPUT

As a PET owner, I was pleased with your first issue of 1980. Especially enjoyable were the PET reading program and "Capture"—both of which I intend to put on tape as soon as my machine returns from its trip to the vet.

Charles C. Burgess
560 Bay St.
Port Richey, FL 33568

MORE ABOUT ZORK

The article you republished (July-August, 1979 *RC*) was hardly enough to do more than frustrate. I wish you would say a bit more about it. For example, I have never written or seen a complete program which parses English. It may be a straightforward program to P. David Lebling & Co., but that's because they've worked with that kind of thing before.

Also, the article only described about half (or less) of the structure of Zork. That's not enough (for me) to modify, translate, rewrite, or do anything better than Adventure. How can anyone write Tiny Zork without knowing what Zork is?

Please publish more about Zork. Release more of the source, publish articles about natural languages, or *something*, anything! How about more pieces of a dungeon, a listing of the 'substrate' or parts of it, a list of comments, an explanation of what the parser uses for a data base, an explanation of some of the 'internals' and 'primitives,' etc.

Raul Miller
Rt. 1 Box 283B
Molalla OR 97038

Yes, we want to publish the answers to all of your requests, but ZORK, like any fantasy adventure game, is a creation. You can take what was presented and make up your own version of ZORK. TINY ZORK may or may not look like the original, based on what you or others do to create it. We suggest that you get together with a group of friends, and put together your ZORK. –*Eds*

Continued from pg. 3

your friends, relatives and associates to subscribe. You could become a retaining subscriber, a sustaining subscriber, or a lifetime subscriber. The premium you pay over normal subscription rates helps us reach new readers and improve the quality of the magazine. Or simply send us a contribution: PCC is a non-profit tax-exempt corporation chartered to promote the educational and personal use of computers. To fulfill that charter takes both imagination and money! – *Dennis Allison*

Submitting Items For Publication

We welcome your articles, drawings, photographs, programs or cartoons. Submissions should be addressed to Editor, *Recreational Computing*, PO Box E, Menlo Park, CA 94025. Please include a self-addressed stamped envelope. If you plan to send a listing please query first for special instructions; listings are very difficult to reproduce legibly.

Letters and short communications help make *Recreational Computing* interesting to read. We assume correspondence to the editor may be published unless you request otherwise. All letters must be signed; we reserve the right to edit for clarity and brevity.

REVIEWS

TWO GREAT SIMULATIONS
1000 Miles & Galactic Empires
Level II, 16K TRS-80
Cybernautics
P. O. Box 40132
San Francisco, CA 94140

From the makers of the outstanding simulations Taipan and Trek 78 (see reviews of these two games in the May-June 1979 issue of *RC*) come two more completely consuming diversions. The latest pair, 1000 Miles and Galactic Empire, provide two programs that could easily entice you to spend several months at your TRS-80. PET and Apple users will just have to find a friend with a TRS-80 until Cybernautics can get around to converting their stable of offerings for other machines.

1000 Miles is a computer simulation of the card game dealing with the events and strategies of being in a 1000 mile road race. You are pitted against the computer, a clever and formidable opponent. Each of you "holds" a hand of cards that describes the actions and events that you can "play" into the game. There are cards that work for you, such as a card that allows you to get your racer started. There are cards that you can play against your opponent, such as an "out-of-gas" card. There are "accident" cards, "repair" cards, and countless ways in which you and the computer can play them.

The simulation is written by Clyde Farrell, a master at developing highly interactive and graphical software products. The action of the game is fast paced, but the complex strategies that can be applied make each game last a l-o-n-g time. The element of chance is set up by the kind of cards you and the computer are dealt and can draw. Beating the computer is either a frustration or a challenge, based on what cards you get. The $11.98 price of the simulation will provide you with hours of enjoyment.

Galactic Empires is for those of you who are becoming jaded with Taipan. Written by Douglas G. Carlston, Galactic Empires establishes you as the commander of the planet Galactica's imperial forces. Your mission: conquer the 20 worlds of the central galactic system. You have 1000 stardates to complete your mission. *You will need that many!*

This simulation is not a "run-in-and-shoot-'em-up" version of some simple wargame translated into space. You must begin in the early stages to establish long-range plans for procuring weapons and supplying your fleet as you move out into the galaxy. Initial attempts to conquer planets must be successful, or you may have to spend several stardates regrouping your resources.

Judicious use of cyrogenics and the high speed capabilities of your fleet can be applied to your ultimate advantage, if you plan carefully. The visual displays of this package are first rate. You begin to quickly feel that you really are in the control room of the fleet's command ship. Your computer can tell you just about anything you need to know. From the bridge of the command ship, you can send scouts off to distant planets, set up navigation orders, tax, conscript, and hire the local planet's resources, and put your entire fleet into cold storage for a few stardates. The price of this package is $14.98, and well worth the money.

Reviewed by Ramon Zamora

ADVENTURES GALORE
The Adventure Tapes
Adventure International
P. O. Box 3435
Longwood, FL 32750

Scott Adams, Chief Adventurer, has parlayed his initial efforts to produce a version of the game Adventure for the microcomputer into a series of truly absorbing simulations. These games (they are more like complex logic puzzles) require you to be inquisitive, innovative, a thinker, a risk taker, a logician, a warrior, and a lover of real challenges—in short, an *adventurer.*

In each adventure, you can move about a series of rooms or caves or islands (based on the particular Adventure you are playing. Using two word commands, you can pick up objects, throw them, eat them, burn them, and otherwise manipulate and examine whatever is in your vicinity. You must sometimes do things to the objects in order to proceed with your Adventure. There is a HELP command when you get hopelessly lost. You can SAVE a game and return to it at a later time. There is treasure to be discovered, and your score in the Adventure involves both finding the treasure and using the objects that you encounter.

Each adventure contains surprises and great comic relief at various points in every game. The simulations are written in machine language, and are based on one basic game framework that uses different data sets. Don't worry, though, the framework does *not* become transparent as you try each adventure. In fact, the latest adventures (there are now eight of them) seem to be getting more and more complex.

I can't possibly talk about each tape individually. The discussion would consume all the pages of the magazine. Perhaps a brief story about one use of these tapes will give you some idea of how good these games are.

In a recent San Diego School District workshop for 60 children, grades 7-9, the Adventure tapes were made available for use as part of the two-day program. Several teacher/observers of the workshop were amazed at how three or four students working together at a TRS-80 would stick with an Adventure for periods exceeding three hours. The students were completely absorbed. They didn't want to take either a rest or a meal break. Suggestions that they might want to stop for a while, and take a break were often met with the response, "Do we have to?"

Of course, these Adventures had the same effect on the adults, once the children could be removed from the machines.

The eight Adventures now available are labeled Adventure 1 through Adventure 8, and have these titles: Adventure Land, Pirate's Adventure, Mission Impossible, Voodoo Castle, The Count, Strange Odyssey, Mystery Fun House, and Pyramid of Doom. Write to Scott Adams, and put some Adventures into your life!

Reviewed by Ramon Zamora

WHAT THE DOCTOR ORDERED
Dr. Daley's Software Library
425 Grove Avenue
Berrien Springs, MI 49103
$69.95 Cassettes
$79.95 Disks

Dr. Daley's Software Library contains 50 (yes, *fifty!*) programs for the PET. If you are concerned with statistics, that works out to be $1.40 to $1.60 per program. If you are more concerned with an innovative way to package and market software, Dr. Daley provides a good example.

The cassette version comes in cassette binders, with accompanying documentation. The documentation is 45 pages long; nearly one page for each program module. The material briefly describes each program, outlines the program's objectives, and provides hints on using each software item. The writing style is concise and informative.

The disk version of the package is also housed in a binder, and includes the same documentation as the cassettes. A brief introduction in the beginning of the documentation tells everything that is needed to run either the cassette or disk series of programs.

The collection of 50 programs is subdivided into eight categories: Social Science and Sports, Pre-School Children's Programs, Elementary School Programs, Arcade-Type Games, Science Fiction Games, Board Games, More Serious Programs, and The Rest.

Most of the programs in the package are taken from the stable of games, simulations, and applications that have become "classics" in the microcomputer software industry. Hamurabi, Hangman, Trek, Othello, Calendar, and Eliza are all to be found here.

For the beginning PET user, the classroom teacher who would like a "packaged" software item, and those of you who would like to have good *user oriented* copies of your favorite programs, Dr. Daley more than fills the bill. The software has been well tested, and appears to have few, if any, errors in *any* of the programs. No attempt has been made to "protect" the software, and the documentation summarizes the company's attitude on this issue, in this way, "It is our conclusion that most of our software customers are honest people, and that they can benefit by being able to tinker with the programs."

As microcomputers proliferate into thousands, and perhaps, millions of homes, neatly packaged products such as Dr. Daley's Software Library will be the norm and not the exception. Before long, I expect to walk into local bookstores and find shelves of software being marketed in Dr. Daley's format. As new storage media (ROM packages, Videodisk, and so on . . .) become available, I expect to subscribe to Software-of-the-Month Clubs, and even have my selections transmitted to me over phone or TV circuits. What else is possible? Perhaps Dr. Daley already has some plans. Write to them and find out.

Reviewed by Ramon Zamora

BALLY ARCADE
Bally Consumer Products Div
10750 West Grand Ave.
Franklin Park, IL 60131
$300

The Bally ARCADE (or HOME LIBRARY COMPUTER when marketed by JS & A) has been around since January of 1978 but until recently was capable only of TV game playing. Now, the additional investment of $49.95 puts one into the home computer field. That sum purchases a Tiny BASIC cartridge that allows access to the Z-80 chip and resident RAM through the existing keypad.

The base unit includes two hand control units (Two more can be attached, for an additional $30), a wall plug transformer, a built-in RF modulator, and the TV/Game switch for the TV set (preferably a color unit). The base unit has a 24-key keypad, a slot for cassette-sized cartridges (called VIDEOCADEs), and storage area for 16 of the cartridges. A tape cassette unit is also available. Locating a machine & cartridges is still difficult; Bally's distribution net is not complete.

Internally there is a Z-80 CPU, and memory consisting of 8K ROM and 4K RAM. This RAM is normally devoted to TV game operation, controlling 8 colors simultaneously and providing a graphic configuration of 160 × 102, but also stores data when in the 5-function calculator mode. Insertion of an additional game cartridge adds that cartridge's ROM to the memory, but for use with the game only.

The games are generally derived from Bally's full-sized arcade games. Football and Baseball are probably the best of the sports games, with Escape, Clowns, and Space Invaders topping the action games. When the tiny BASIC cartridge is installed, the 24-key keypad is converted into a 91-function device, and an overlay card is used to show the new configuration. A key-shift arrangement is used for a total of five levels. In the unshifted mode, numerals and the four arithmetic functions are operational. Then, when the GOLD key is pressed, the system shifts to allow BASIC *commands* to be entered by a single keystroke. Similarly, *letters* and *punctuation marks* require pressing of the BLUE, RED, or GREEN key first. As each shift key is pressed, its corresponding color floods the screen, and as each input key is pressed, a distinctive tone is heard. Constant attention to the screen is therefore not needed. This scheme allows use of the inexpensive keypad as a stopgap affair until a full keyboard is developed. Operation of the keypad is somewhat laborious, but programs can be entered at a fair clip as one gets used to it. Once inadvertently struck, the program is lost.

The language itself is a slight modification of Palo Alto Tiny BASIC III, documented by Dr. Wang in PCC's *Reference Book of Personal and Home Computing.* The Bally manual is less complete than the explanation in the original document.

Because of the video card heritage, color and sound are enhanced enabling programs and self-generated games to have unique features:

- GRAPHICS—The unusual commands of LINE and BOX provide the user with the graphics capability to construct graphs, charts, and shapes on the screen.
- COLOR—A palette of 256 colors is available, with only two available at a time.
- SOUND—The machine provides a single voice in Tiny BASIC that is 3 octaves in span including sharps and flats. These are played by PRINTing numbers 1 thru 7, which correspond to notes C thru B. Prefix operators are used to shift between octaves and half notes.
- HAND CONTROLLERS—Since these contain input devices (switches and a potentiometer) that are addressable program data can be generated from the controller. It is also possible to input the results of totally external operations through these ports.

Examination of the sample tape listings of the cassette interface revealed that there were operations and commands that were not mentioned in the manual. The most powerful of these are $\%(n)=m$ which provides the PEEK and POKE operators; and $\&(a)=t$ which yields varying results having to do with color and sound as various values of a are set, and t is varied from 0 to 255.

From other experimentation have come machine language inputs, memory dump programs in hex and/or binary, and four function calculator operations with decimals. Other things being worked on include data storage on tape, peripheral and memory additions, as well as other work-arounds to go beyond the boundaries of Tiny BASIC's capability. A group of experimenters subscribe and contribute to the *ARCADIAN*, a journal containing material unique to the Bally product.

Owners are awaiting the release of the Keyboard Programmer option. Prototype units include a storage increase to a total of 20K RAM and 32K ROM on board with access to two cassettes and to other peripherals via an RS-232 serial and a 50-pin parallel connector. A new language, ZGRASS, will be used that utilizes labels for subroutines and can perform multiple operations at once through a layering technique. The option includes a super video system called GRAFIX, whose capabilities were shown at the CES last January.

Bally is now saying that they will delay production of the Keyboard until the question of reduced interference requirements now before the FCC is resolved. The basic unit was first advertised by JS & A in October of 1977, with a four week delivery. That date was not met, and things never really got better. The advertising indicated that the Keyboard would be introduced in 1978. At the January CES, Bally was talking about June/July 1979. In a field of extremely rapid movement, these delays, coupled with the lack of advertising of the basic unit and its capabilities, are certainly hurting the concept.

Reviewed by Robert Fabris
San Jose, CA 95127