$14.95

GAMES TIs PLAY

By Scott L. Singer and Tony E. Bartels

Based on Games Apples Play by
Mark James Capella and Michael D. Weinstock

Learn programming the fun, enjoyable way . . . by gaming!
Included is a vast selection of classic games for your TI-99/4A
written in TI BASIC. Why make programming hard work?

GAMES
TIs
PLAY

GAMES TIs PLAY

by

Scott L. Singer

and

Tony E. Bartels

Based on Games Apples Play

by

Mark James Capella and Michael D. Weinstock

Cover Art and Illustrations by

Art Huff

DATAMOST INC™
8943 Fullbright Avenue, Chatsworth, CA 91311-2750
(818) 709-1202

DATAMOST™
ISBN 0-88190-120-2

Copyright © 1983 by DATAMOST Inc.
All Rights Reserved

This manual is published and copyrighted by DATAMOST Inc. Copying, duplicating, selling or otherwise distributing this product is hereby expressly forbidden except by prior written consent of DATAMOST, Inc.

The word TI-99/4A and the TI logo are registered trademarks of Texas Instruments Inc.

Texas Instruments Inc. was not in any way involved in the writing or other preparation of this manual, nor were the facts presented here reviewed for accuracy by that company. Use of the term TI-99/4A should not be construed to represent any endorsement, official or otherwise, by Texas Instruments Incorporated.

ACKNOWLEDGMENTS

To my wife, Genie, for all those times she turned her head to see me sneaking up the stairs to the computer room.

Mark James Capella

To Diana and Aaron. Diana for her patience, understanding and the cold dinners she ate while waiting for me to finish just a little bit more. Aaron for the thrill of understanding what it means having a four year old around when things just aren't working correctly.

To Brian Davis for his creative artistic talents that greatly assisted our effort.

To Dave Gordon because he is Dave Gordon, a trusted friend.

Michael D. Weinstock

To Mark James Capella and Michael D. Weinstock for preparing an excellent collection of games and allowing them to be adapted to the TI-99/4A.

To Tony E. Bartels for opening his bag of programming tricks and incorporating some of his best POKEs and PEEKs in the programs.

Special thanks to Gary Heberling, Quality Control Manager, for debugging and standardizing the program listings.

To all of the DATAMOST staff who added their energy in getting this book completed.

Scott L. Singer

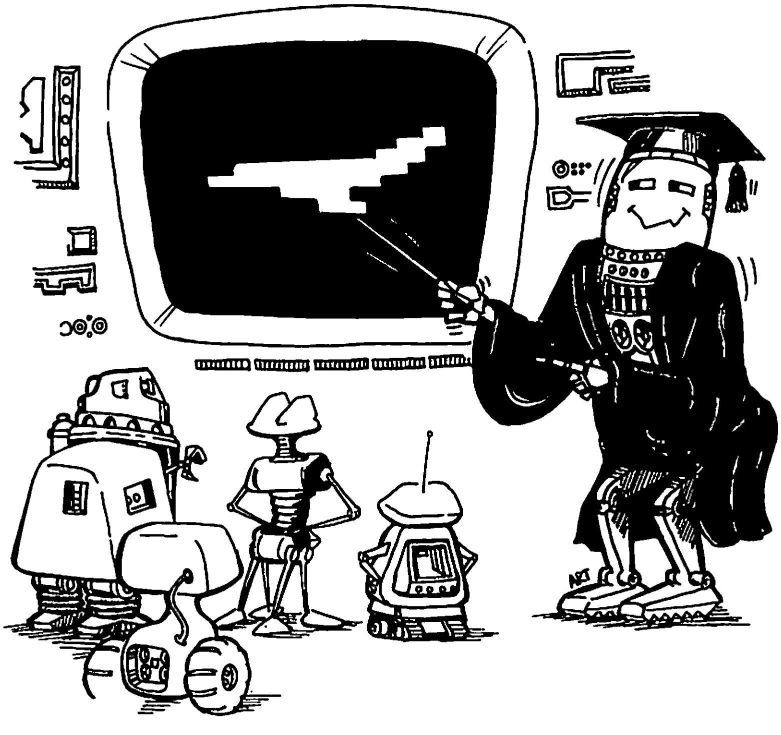
Table of Contents

Programming Word and Number Games

Programming Graphic Action Games

INTRODUCTION
Using games to learn BASIC

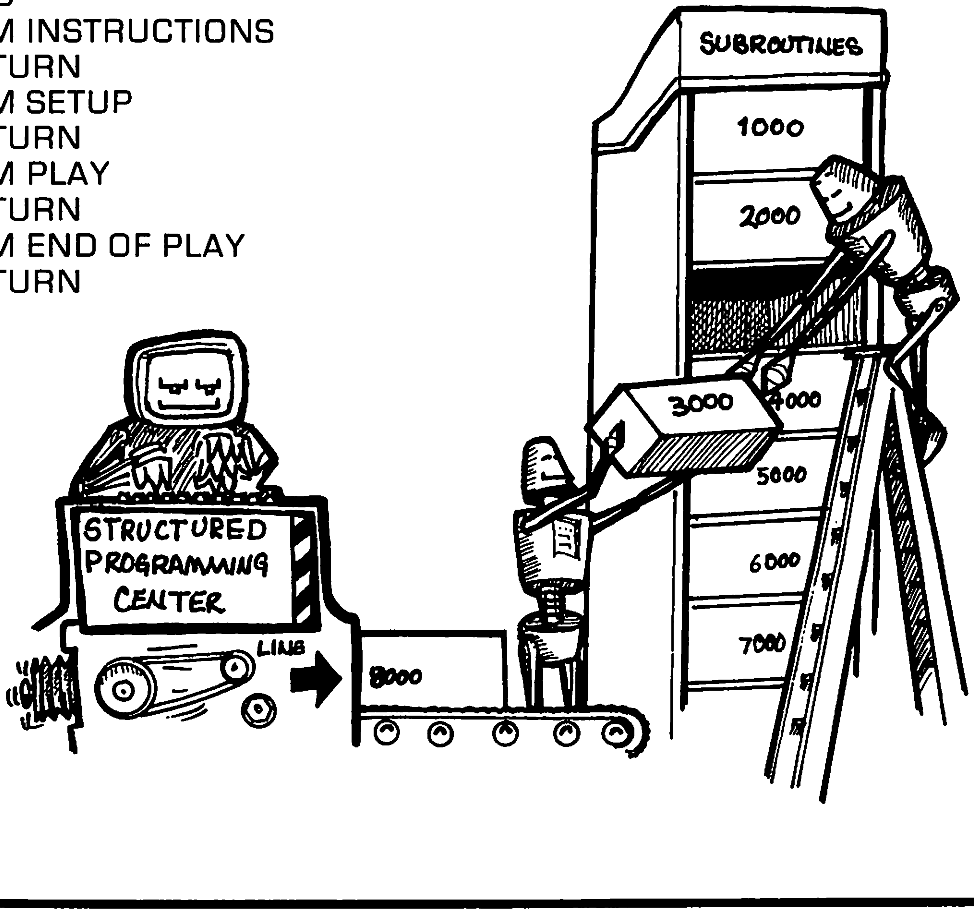
To date, Texas Instruments does not have an extensive market for games. Even if it did, today's games are written in machine language with elaborate protection schemes. These new techniques of increasing program speed and thwarting "pirates" have greatly limited the beginner's ability to look at and learn from game programs. This is a fairly recent development. Back in the sixties and early seventies no one was writing such sophisticated programs. Back then, people were hesitant and even afraid of the mechanical monstrosities we call computers. Now, those futuristic contraptions with 16K and virtually no available software are practically obsolete. Programs were of little value, and when a particularly outstanding program was written, it was published — line by line — in a magazine. By comparison, machine speed was very slow. Early graphics could be termed monotonous. Everything seemed to move in slow motion. In the middle of a game you could run to the refrigerator and back and not miss much. Today, fantastic graphics move so fast that any distraction can spell disaster. Author Michael Weinstock has compiled a selection of classic, easy to understand games. The programs are written in straightforward, easily understood, structured BASIC. With most of the games, speed is not a factor. Adventure games such as GRUE STEW, word games such as HANGMAN, and board games such as CONNECT FIVE all require the players to think — and you can't rush genius.



The games are written in TI BASIC using a structured format. Structuring renders the programs more understandable. You will be able to edit, change, and reuse different sections of a program. The games are explained in such a way that you will be able to customize and modify them to your heart's content. Two of the games, MUBBLE CHASE and SCI-FI, are examined line by line. SCI-FI introduces text formatting, input routines, READ statements, and data manipulation. These are the same concepts you'll find at the heart of any word game. MUBBLE CHASE will help you to understand simulated movement (animation), basic grid construction, and figure detection and identification.

All of the programs in this book reflect the same fundamental structure. Four GOSUBs control the program. Each target line contains a blank or empty REMark, and each major subroutine ends with a RETURN statement at the end of the section. Using this structured technique, you can write the basic structure before you write the inner workings of the program. Below, you will find a complete program outline.

```
 100 REM **STRUCTURED PROGRAM**
 140 GOSUB 1000
 150 GOSUB 2000
 160 GOSUB 3000
 170 GOSUB 4000
 190 END
1000 REM INSTRUCTIONS
1990 RETURN
2000 REM SETUP
2990 RETURN
3000 REM PLAY
3990 RETURN
4000 REM END OF PLAY
4990 RETURN
```



You can use this outline for your own programs. It helps in general organization, and it helps the programmer know where certain steps belong. Copy and save the program (program outline) for when you feel creative.

You will be given various methods and suggestions on how to dissect BASIC programs and see what makes them tick. You will learn a lot about programming by typing in, debugging, and running the programs in this book. You may order the games diskette from DATAMOST or you can copy them from the book. Either way, you'll agree that this is a truly enjoyable way to learn BASIC!

Don't be afraid of your computer. Remember, unlike a person, a computer has only a very small vocabulary. Not only that, the computer can't figure out misspelled words or omissions. The bottom line is — a computer can't think. It can do what it is told to do and no more. TI BASIC has a carefully explained set of error messages. If your program gets interrupted, you'll be able to find out why very quickly. There is nothing you can type that will damage the computer, which is probably a good thing since everyone gets frustrated occasionally. The most important thing is to experiment; after all, you can't learn without making mistakes.

Whatever is typed into memory or loaded into it from an external device exists in a temporary state. If you turn off the computer or the dog chews through the power cord, the program will cease to exist (in a temporary state). But anything that was SAVEd will exist exactly as it was saved on a permanent storage device. The relevance of this is that you can alter and experiment with a program and, as long as you don't SAVE the changed version under the same name, recall the original. Try the following techniques for investigating the inner workings of your progams:

Change Lines

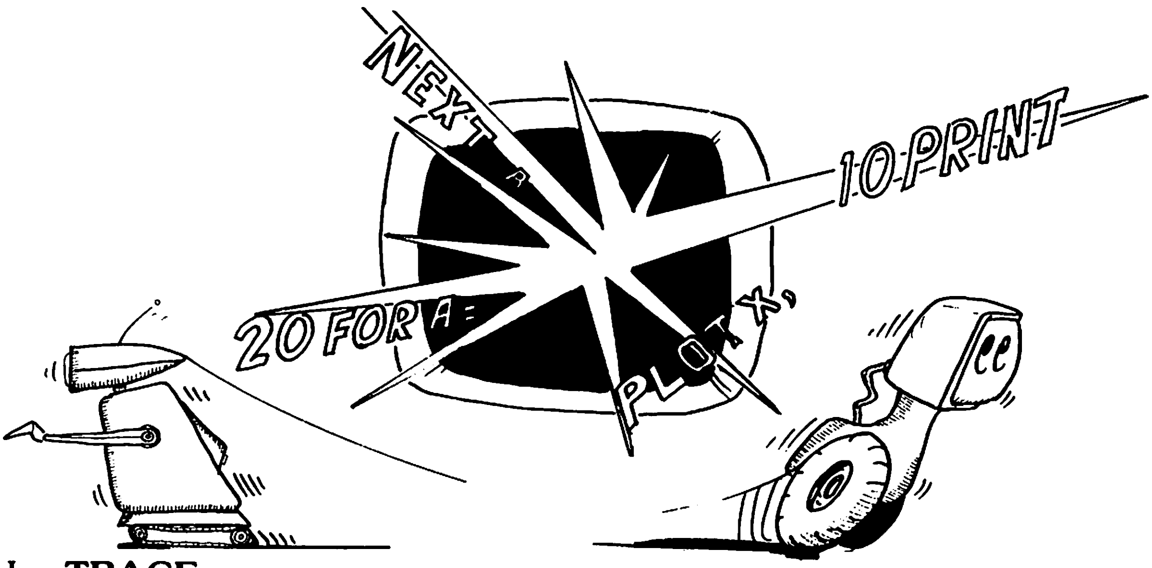
As an example, if you see a line that reads CALL COLOR(2,X,12), change it to CALL COLOR(2,7,12) and note which figure is incorrectly colored. Using this technique you can determine the function of almost any line. Oftentimes, figures are drawn at one location, blacked out or made invisible, and redrawn one position away. This technique simulates motion and can be called "animation." The following is a brief explanation of blacking out a drawing. If a blue figure is drawn on a white background, it will stand out. If a blue figure is drawn on a black background it will stand out — but not as well. If a blue figure is drawn on an aqua background it will be distinguishable, but not clear. If a blue figure is drawn on a blue background it will be invisible — blacked out. The point is, when a figure is blacked out, the implication is that the figure is covered in black, but this is seldom the case. The normal policy is to redraw the figure so that the background and the foreground colors match. If you change a line where the background and the foreground colors match — CALL COLOR(X,7,7) to a line where the colors don't match — CALL COLOR(X,4,7) you'll undermine the process of blacking out. By making changes, you will discover which instruction colors which game piece. Also, change variables (A,B,X,Y,ME etc.) to numbers. Does the program still run? Correctly? Do you get an error message? Variables are often hard to understand. The following technique should help remedy this common difficulty. Assume that line 3450 is the problem.

```
3450 A=INT(RND*X+1)
```

If you don't understand this line, then add line 3451.

```
3451 PRINT X,A
```

Perhaps the values which correspond to X and A will help you to understand what purpose the variables serve.



Use TRACE

If you type this magic word (TRACE) before you run a program, then the computer will display the line number of each function it executes. This is extremely helpful when debugging a program. If a loop exists, then the line numbers will be repeated the appropriate number of times. Look at the example below.

```
10 FOR A = 1 TO 4
20 PRINT "THE PROGRAM WILL LOOP"
30 PRINT 4-A;
40 PRINT "MORE TIMES."
50 PRINT
60 NEXT A
```

Type TRACE and then run. The output will be:

(Line numbers appear between angle brackets.)

```
<10><20>
THE PROGRAM WILL LOOP
<30> 3 <40>MORE TIMES.
<50>
<60><20>
THE PROGRAM WILL LOOP
<30> 2 <40>MORE TIMES.
<50>
<60><20>
THE PROGRAM WILL LOOP
<30> 1 <40>MORE TIMES.
<50>
<60><20>
THE PROGRAM WILL LOOP
<30> 0 <40>MORE TIMES.
<50>
<60>
** DONE **
```

By following the line numbers (in parentheses), you can follow the flow of the program. To stop TRACing, type NOTRACE.

Use STOP

Put in a STOP command immediately following a line that you don't understand. By seeing what has and what has not been done, you may be able to understand the line's function. If not, precede the line in question with a STOP. Perhaps the absence of a certain attribute will clarify the function of the mystery line.

Detour

If you would like to know what effect skipping a certain line has, immediately precede it with a GOTO statement. For example, if line 40 is in question, type:

```
10 -
20 -
30 -
39 GOTO 50
40 REM SKIP THIS LINE
50 -
```

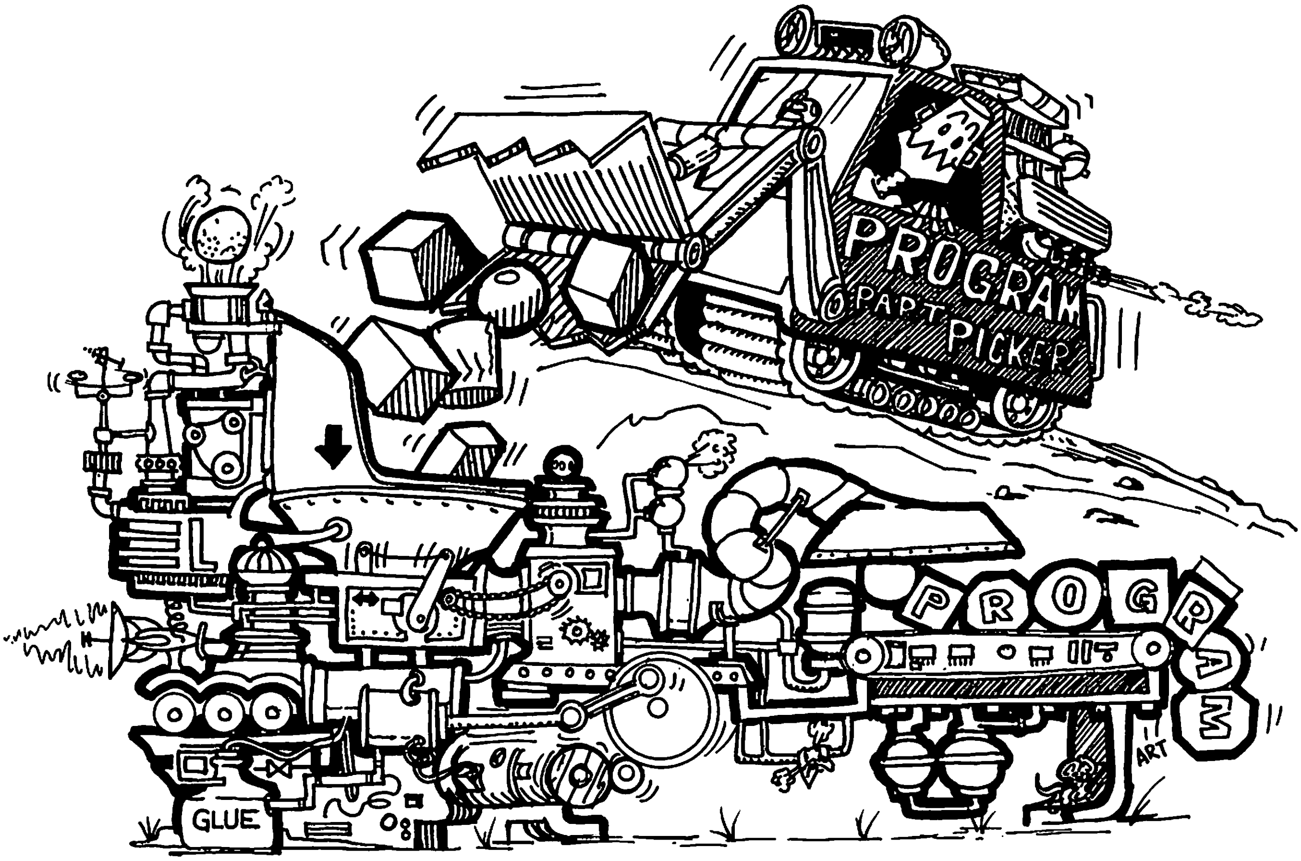
Using this approach, line 40 will be skipped.

Delete Lines

Rather than using the detour method, you can simply delete a line. This is done by typing the line number and pressing ENTER. After deleting a line, run the program to see what omissions or errors occur. To recapture the original, unedited program, reload the file that you initially loaded.

Isolate Parts of the Program

If you want to run a certain segment of a program there are three ways to do it. One, you can delete all but the lines in question. This may prove tedious. Two, you can add a line at the beginning of the program which directs flow to a specific line, i.e., 10 GOTO 2230 (or whichever line you choose). An example will be used to demostrate the third method. Assume that in the program below you want to isolate lines 40-60:



```
 10 CALL CLEAR
 20 REM EXEMPLARY PROGRAM
 30 INPUT "WHAT IS YOUR NAME? ":N$
 40 PRINT "YOUR NAME IS ";N$
 50 INPUT "WHAT IS YOUR FAVORITE COLOR? ":C$
 60 PRINT "GREEN IS MY FAVORITE."
 70 CALL CLEAR
 80 FOR I=1 TO 16
 90 CALL COLOR(I,I,I)
100 CALL VCHAR(6,8+I,I*8+24,I)
110 NEXT I
120 GOTO 120
```

Copy and run the above program. Now for isolation method three: Add line 61 by typing: 61 STOP. Now type: RUN 40. The computer will start at line 40 and STOP at line 61.

Save Your Changes

When you make changes to the program, and you want to save the changed version, you can do one of two things. If you save the changed program under the same name as the original, then the updated version replaces the original. The alternative is to save the changed program under a new name. This preserves the original for future comparison.

A Color Program

Before you start looking at the individual programs, be sure to copy and run the program in the "Isolate Parts of the Program" section.

When this program is run, you will see a display of 15 colors. Color 1 is omitted because it is transparent (or invisible). The background color is 3 (medium green) so the color strip for 3 is indistinguishable. Line 110 keeps the colors on the screen. When you have seen enough, press FCTN-4 (CLEAR). The following color chart should help you to identify colors.

CODE #	COLOR
1	TRANSPARENT
2	BLACK
3	MEDIUM GREEN
4	LIGHT GREEN
5	DARK BLUE
6	LIGHT BLUE
7	DARK RED
8	CYAN
9	MEDIUM RED
10	LIGHT RED
11	DARK YELLOW
12	LIGHT YELLOW
13	DARK GREEN
14	MAGENTA
15	GREY
16	WHITE

This program is designed to write clever little science fiction stories. Contained within this program are both constant data and user inputted data. You have the opportunity to make the stories as personally relevant as you wish, so have fun.

SCI-FI is one of the two programs in this book that contain an in-depth explanation.

10 REM stands for REMark. Any comments, numbers, symbols, expletives, or anything else may follow a REM. In this case, the remark is used to highlight the program name. In effect, the computer ignores the material which follows the REM statement. In this case, the REMark ******************** is made.

20-50 These REMarks complete the program name.

60 The entire program is controlled by lines 60-90. GOSUB 110 instructs the computer to branch to line 110, and to continue from there until the RETURN command is encountered.

70 When the subroutine begun by line 60 is completed by the RETURN command, the program drops down to the next sequential instruction, which in this case is line 70.

80-90 The function of these lines is identical to the function of line 60.

100 END takes control from the program and returns it to the computer.

110 Although it may or may not contain any instructions, line 110 must exist because it was specified in line 60. The sole purpose of the subroutine consisting of lines 110-410 is to give INSTRUCTIONS.

140 CALL CLEAR clears the screen of all text.

150-220 These lines draw the introductory announcement. Anything written between the quotation marks ("") will be PRINTed verbatim.

230 This line initiates a loop with eight passes (FOR I=1 TO 8). The loop is performed when I=1,2,3,4,5,6,7 and 8. After the eighth pass of the loop, I is incremented, making I equal to 9. Because this violates the parameters (1 TO 8), the program will drop to the first sequential instruction following the loop.

250 NEXT completes each pass of the loop.

260-270 This is referred to as a delay loop. The delay is used to give the user time to read the introductory statement.

280 This line calls a subroutine at line 380 that prints the title at the top of the screen.

290-320 These lines display the game description.

330-350 In this loop, 12 blank lines are PRINTed.

360 An INPUT statement pauses indefinitely, waiting for the user to validate the input by pressing ENTER. Copy and run the following program:

```
10 CALL CLEAR
20 INPUT "WHAT IS YOUR NAME? ":ANS$
30 INPUT "WHAT IS YOUR AGE? ":AGE
40 CALL CLEAR
50 PRINT ANS$;" IS";AGE;"."
```

When you run this program, the screen will clear and then the question, "WHAT IS YOUR NAME?" will be displayed. At this point, the computer will pause indefinitely, until you press ENTER. The computer does not care whether or not you have entered a name. Whatever you have typed, up to the time you press ENTER, is stored in ANS$. The same basic procedure is repeated for line 30 — with a slight twist. The dollar sign ($) in ANS$ allows you to input any and all types of keyboard characters into the variable location, i.e., KATHY,K.R.Y.,I'M #1,SCOTT,R2D2,***ME, or whatever. The absence of a dollar sign restricts legitimate input to numbers (including periods, minus signs, and plus signs, but not commas), for example, 73222811, -3.1415, +7355.2, etc. The reason is as follows: only numeric fields (variables without dollar signs) may be used in mathematical equations. Even though a string field (variable with a dollar sign) may contain a number, it cannot be used in an equation. To demonstrate this point, add this line:

```
60 B=A+ANS$
```

When you run the program, be sure to enter a number for your name. You should get the following error message: * STRING-NUMBER MISMATCH IN 60.

370 This line returns control to the next sequential instruction after the GOSUB which sent it there. In this case control will be returned to line 70.

380-400 Subroutine to print title.

410 Control was sent to this subroutine from line 280, so control will revert back to line 290 when this RETURN is encountered.

420 Line 420 is essential, due to line 70 (see the explanation for line 110).

460 The computer normally reserves one storage location per variable. This means that if you wanted to store the seven days of the week, you would need seven different variable names, right? Not necessarily. Instead of having seven names (MON$,TUE$,WED$, etc.), the seven names can be stored, independently, in one file. The first step is to DIMension memory: DIM DAYS$(7). Then, because DAYS$ has reserved seven memory locations, DAYS$(1) can equai Monday, DAYS$(2) can equal Tuesday, DAYS$(3) can equal Wednesday . . . and DAYS$(7) can equal Sunday.

470-490, 500-520, 530-550, 560-580 These four loops allow the user to input his own data.

590 In order to avoid the subroutines in 600-870, this line instructs the computer to skip to line 880.

600-660, 670-730, 740-800, 810-870 Each of these loops performs a subroutine five times. You will recall that line 70 initiated the current subroutine, and these lines create loops which also contain GOSUBs. A GOSUB within a GOSUB (referred to as a nested GOSUB) is a legitimate programming technique.

880 A DATA statement contains material which is read into memory via a READ statement.

890 DEF is short for DEFine. In this particular instance, the instruction allows the computer to identify R(any number) as RND * (any number) + 1. In other words, because of the DEF instruction, R(8) = INT (RND * 8) + 1.

900, 940, 970, 1000 These lines begin seven-pass READ loops.

910, 950, 980, 1010 A READ statement will locate and read the first piece of available (unused) data.

1030-1050 These lines contain data. Unless otherwise stipulated, data may be read only once. A single DATA statement may contain many pieces of data. Each item is separated by a comma.

1060 RETURN ends the subroutine initiated by line 70. Program flow is returned to 70, and drops to 80.

1100 There is an interesting feature which accompanies RaNDom number functions. Although the generated numbers are indeed random, unless instructed otherwise, the computer will generate the same random numbers each time the program is run. To illustrate this point, copy and run the following program:

```
10 FOR X=1 TO 5
20 PRINT INT[RND * 9] + 1
30 NEXT X
```

When this program is run, a list of five INTegers between 1 and 9 will be displayed. If you run this program ten more times, the same list will be displayed. Now add:

```
15 RANDOMIZE
```

Because of this addition, a different set of five random integers will be displayed each time the program is run.

1120 An ON statement merits careful study. Remember from line 890 that R(5) equals a random number between 1 and 5. To paraphrase 1120: Depending on the value of R(5), GOTO line 1130, 1150, 1170, 1190, or 1210. In other words, if R(5)=1 then GOTO 1130, if R(5)=2 then GOTO 1150, and so on.

1240 This line RaNDomly chooses (R(12)) one of the 12 sites to be attacked.

1250 The subroutine beginning at line 2110 is performed in order to add WRD$ (CA$(R(12))) to the story.

1390 The word " FROM " and preceding and trailing spaces are moved into WRD$.

1440 The value of J is set to 1 so that the first time through, line 1470 will not cause the program to branch to 1870.

1450-2110 These lines serve to build the story.

2130 The value of X, which serves to count the characters in WRD$, is set to 1 at the beginning of the subroutine.

2140 A comparison between the value of X and the number of characters in WRD$ (its LENgth) is made to see if the subroutine should be exited.

2150 The number of characters which have been printed on a line (LL=Line Length) is monitored so that the story won't breech the right margin.

2160 This line PRINTs a SEGment of the story, one character at a time.

2190 If the Line Length is greater than 18, and the current character (the Xth character of WRD$) is a space, then PRINTing skips to the beginning of the next line.

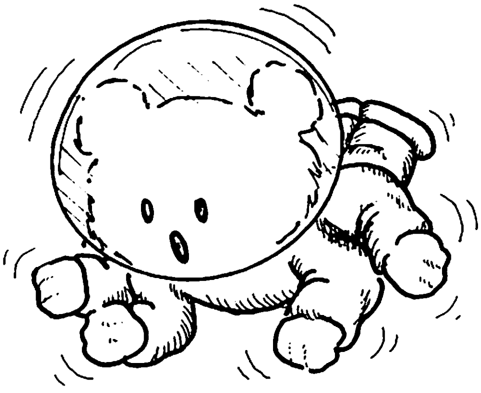
2210 LL is set back to 1 because PRINTing is to begin at the 1st position of a new line.

2280-2300 This PRINT loop serves to center the story on the screen.

2310-2320 This loop delays the program while the computer counts to 200.

2380-2390 These two lines check to see if the SEGment of the string ANS$, starting at position 1, and reading only 1 character (SEG$(ANS$,1,1)), is either an upper or a lower case "N." By employing this technique, the program will end if the response to line 2330 is "no", "NO WAY", "Negative", or "not today".

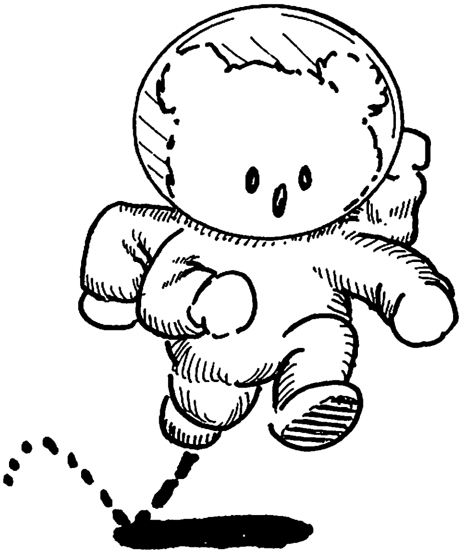
```
10 REM ********************
20 REM ***              ***
30 REM ***    SCI-FI    ***
40 REM ***              ***
50 REM ********************
60 GOSUB 110
70 GOSUB 420
80 GOSUB 1070
90 GOSUB 2250
100 END
110 REM
120 REM ***INSTRUCTIONS***
130 REM
140 CALL CLEAR
150 PRINT "  ._______________
__________."
160 PRINT "  :
```



```
         :"
170 PRINT "  : SCIENCE FICTI
ON TALES :"
180 PRINT "  :
         :"
190 PRINT "  :         FROM
         :"
200 PRINT "  :
         :"
210 PRINT "  :  THE ELECTRON
IC MIND  :"
220 PRINT "  :______________
_________:"
230 FOR I=1 TO 8
240 PRINT
250 NEXT I
260 FOR I=1 TO 1000
270 NEXT I
280 GOSUB 380
290 PRINT "This program will
 produce   lots of funny lit
tle sciencefiction stories."
300 PRINT
310 PRINT "You are given the
 chance to enter your own in
formation"
320 PRINT "in hopes of makin
g the      stories more pers
onal."
330 FOR I=1 TO 12
340 PRINT
350 NEXT I
360 INPUT "Press ENTER when
ready to   continue: ":ANS$
370 RETURN
380 CALL CLEAR
390 PRINT TAB(9);"*** SCI-FI
 ***"
400 PRINT
410 RETURN
420 REM
430 REM ***SETUP***
440 REM
450 CALL CLEAR
460 DIM AA$(13),BA$(13),CA$(
13),DA$(13)
470 FOR A=1 TO 5
480 GOSUB 600
490 NEXT A
500 FOR A=1 TO 5
510 GOSUB 670
```

```
520 NEXT A
530 FOR A=1 TO 5
540 GOSUB 740
550 NEXT A
560 FOR A=1 TO 5
570 GOSUB 810
580 NEXT A
590 GOTO 880
600 PRINT "Input the names o
f five     characters who mi
ght save    the day."
610 FOR I=1 TO 14
620 PRINT
630 NEXT I
640 INPUT "Press ENTER after
 each        selection : ":AA$
(A)
650 CALL CLEAR
660 RETURN
670 PRINT "Enter the names o
f five     grizzly monsters
who will    attack."
680 FOR I=1 TO 14
690 PRINT
700 NEXT I
710 INPUT "Press ENTER after
 each        selection : ":BA$
(A)
720 CALL CLEAR
730 RETURN
740 PRINT "Enter the names o
f five     places that will
be attacked"
750 FOR I=1 TO 15
760 PRINT
770 NEXT I
780 INPUT "Press ENTER after
 each        selection : ":CA$
(A)
790 CALL CLEAR
800 RETURN
810 PRINT "Enter the names o
f five     places where the
monsters    come from"
820 FOR I=1 TO 14
830 PRINT
840 NEXT I
850 INPUT "Press ENTER after
 each        selection : ":DA$
(A)
860 CALL CLEAR
```

```
870 RETURN
880 DATA BIG BIRD,THE FONZ,T
HE DUKE BOYS,GEORGE OF THE J
UNGLE,TOM SELLECK,MAGIC JOHN
SON,SUPERMAN
890 DEF R(X)=INT(RND*X)+1
900 FOR I=6 TO 12
910 READ A$
920 AA$(I)=A$
930 NEXT I
940 FOR I=6 TO 12
950 READ CA$(I)
960 NEXT I
970 FOR I=6 TO 12
980 READ BA$(I)
990 NEXT I
1000 FOR I=6 TO 12
1010 READ DA$(I)
1020 NEXT I
1030 DATA DISNEYLAND,DETROIT
,CUCAMONGA,TIMBUKTU,THE UNIT
ED STATES,OKEEFENOKEE SWAMP,
MT. ST. HELENS
1040 DATA ANDROIDS,ALIENS,HA
RI KRISHNAS,ZOMBIES,PUNK ROC
KERS,FLYING GOOKIES,A SWARM
OF BRAZILIAN KILLER BEES
1050 DATA MARS,A TIME WARP,T
HE MILKY WAY,HALEYS COMET,SE
ARS ROEBUCK,TIJUANA,A BLACK
HOLE
1060 RETURN
1070 REM
1080 REM ***PLAY***
1090 REM
1100 RANDOMIZE
1110 CALL CLEAR
1120 ON R(5)GOTO 1130,1150,1
170,1190,1210
1130 PRINT "   *** FLASH! FL
ASH! ***"
1140 GOTO 1220
1150 PRINT "     *** BULLETI
N ***"
1160 GOTO 1220
1170 PRINT "     *** ALERT!!
! ***"
1180 GOTO 1220
1190 PRINT " *** SPECIAL BUL
LETIN ***"
1200 GOTO 1220
```

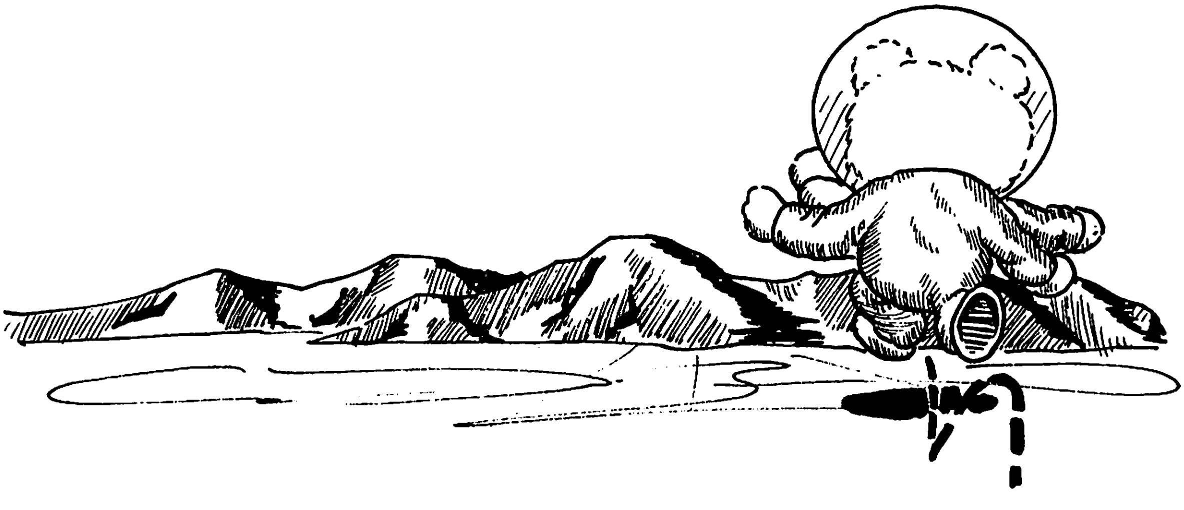


```
1210 PRINT " *** TO ALL CITI
ZENS!!! ***"
1220 PRINT
1230 PRINT
1240 WRD$=CA$(R(12))
1250 GOSUB 2110
1260 ON R(5)GOTO 1270,1290,1
310,1330,1350
1270 WRD$=" WAS ATTACKED BY
"
1280 GOTO 1360
1290 WRD$=" WAS TICKLED BY "
1300 GOTO 1360
1310 WRD$=" WAS OVER-RUN BY
"
1320 GOTO 1360
1330 WRD$=" IS BEING INVADED
 BY "
1340 GOTO 1360
1350 WRD$=" IS UNDER THE SPE
LL OF "
1360 GOSUB 2110
1370 WRD$=BA$(R(12))
1380 GOSUB 2110
1390 WRD$=" FROM "
1400 GOSUB 2110
1410 WRD$=DA$(R(12))
1420 GOSUB 2110
1430 PRINT ". ";
1440 J=1
1450 WRD$=AA$(R(12))
1460 GOSUB 2110
1470 IF J=88 THEN 1870
1480 WRD$=" TRIED TO "
1490 GOSUB 2110
1500 ON R(5)GOTO 1510,1530,1
550,1570,1590
1510 WRD$="FEED THEM CHOCOLA
TE COOKIES, "
1520 GOTO 1600
1530 WRD$="FIGHT THEM, "
1540 GOTO 1600
1550 WRD$="TURN ON HOWARD CO
SELL, "
1560 GOTO 1600
1570 WRD$="MAKE THEM EAT VEG
ETABLES, "
1580 GOTO 1600
1590 WRD$="ATTACK AT DAWN, "
1600 GOSUB 2110
1610 WRD$="BUT THEY "
```

```
1620 GOSUB 2110
1630 ON R(5)GOTO 1640,1660,1
680,1700,1720
1640 WRD$="SCREAMED AND LAUG
HED. "
1650 GOTO 1730
1660 WRD$="SENT OUT FOR PIZZ
A. "
1670 GOTO 1730
1680 WRD$="SMILED HAPPILY. "
1690 GOTO 1730
1700 WRD$="WERE TOO TOUGH. "
1710 GOTO 1730
1720 WRD$="SCREAMED FOR MORE. "
1730 GOSUB 2110
1740 ON R(5)GOTO 1750,1770,1
790,1810,1830
1750 WRD$="FINALLY, "
1760 GOTO 1840
1770 WRD$="THEN, "
1780 GOTO 1840
1790 WRD$="BUT THEN, "
1800 GOTO 1840
1810 WRD$="AFTER, "
1820 GOTO 1840
1830 WRD$="LATER, "
1840 GOSUB 2110
1850 J=88
1860 GOTO 1450
1870 ON R(5)GOTO 1880,1900,1
920,1940,1960
1880 WRD$=" SHOWED THEM RERU
NS OF I LOVE LUCY, "
1890 GOTO 1970
1900 WRD$=" YELLED AT THEM, "
1910 GOTO 1970
1920 WRD$=" EXPOSED THEM TO
MEASLES, "
1930 GOTO 1970
1940 WRD$=" DROPPED CHICKEN
SOUP ON THEM, "
1950 GOTO 1970
1960 WRD$=" NUKED THEM, "
1970 GOSUB 2110
1980 WRD$="SO THEY "
1990 GOSUB 2110
2000 ON R(5)GOTO 2010,2030,2
050,2070,2090
2010 WRD$="KICKED THE BUCKET."
2020 GOTO 2110
2030 WRD$="VOTED FOR JERRY B
```

```
ROWN FOR PRESIDENT."
2040 GOTO 2110
2050 WRD$="CRIED LIKE BABIES
."
2060 GOTO 2110
2070 WRD$="TURNED INTO GREEN
 SLIME."
2080 GOTO 2110
2090 WRD$="LEFT FOR HOME."
2100 REM
2110 REM  WRD$ WRITE SUBROUT
INE
2120 REM
2130 X=1
2140 IF X>LEN(WRD$)THEN 2240
2150 LL=Y+X
2160 PRINT SEG$(WRD$,X,1);
2170 Y=Y+1
2180 X=X+1
2190 IF (LL>18)*(SEG$(WRD$,X
,1)=" ")THEN 2210
2200 GOTO 2140
2210 Y=1
2220 PRINT
2230 GOTO 2140
2240 RETURN
2250 REM
2260 REM ***END***
2270 REM
2280 FOR A=1 TO 6
2290 PRINT
2300 NEXT A
2310 FOR DELAY=1 TO 200
2320 NEXT DELAY
2330 PRINT "WOULD YOU LIKE T
O SEE ONE   MORE STORY? (Y/N
)"
2340 FOR B=1 TO 3
2350 PRINT
2360 NEXT B
2370 INPUT ANS$
2380 IF SEG$(ANS$,1,1)="N" T
HEN 2420
2390 IF SEG$(ANS$,1,1)="n" T
HEN 2420
2400 LL=1
2410 GOTO 80
2420 CALL CLEAR
2430 PRINT "HOPE YOU ENJOYED
 THE CUTE   SCIENCE FICTION
STORIES!"
```

```
2440 PRINT
2450 PRINT TAB(10);"BYE BYE!
"
2460 FOR Z=1 TO 14
2470 PRINT
2480 NEXT Z
2490 RETURN
```



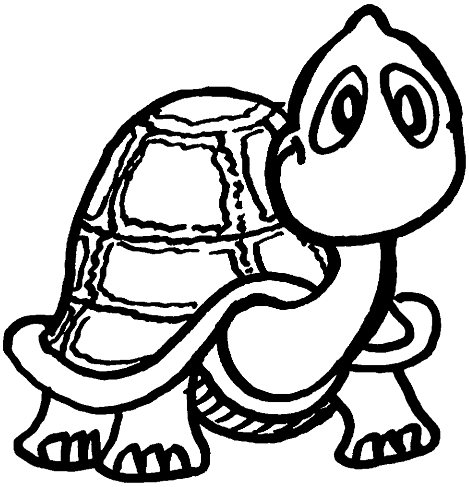
ANIMAL LEARNER

This game is designed to compel the user to define any given animal. The computer stores this information, and thereby "learns." The computer, according to program design, can store up to fifty separate animal types and fifty variations of the fifty animal types. Although each run starts with the same information, your additions make it harder and harder to stump the computer.

To see the SETUP, type LIST 300-410. The only two objects which the computer "knows" are a lion and a grizzly bear. Line 380 sets RA$(1) to LION and line 390 sets WA$(1) to GRIZZLY BEAR. Line 340 DIMensions memory to allow for fifty memory locations for certain variables. Memory is normally structured to accommodate one value per variable. The DIM statement specifically tells the computer to arrange memory so that certain variables can contain many different values.

For instance, QU$ has fifty locations; enough for the fifty states. QU$(1) could equal Alaska, QU$(2) could equal Alabama, QU$(3) could equal Arizona . . . QU$(49) could equal Wisconsin, and QU$(50) could equal Wyoming. This program shows the user how to use arrays. It is essential to understand this concept.

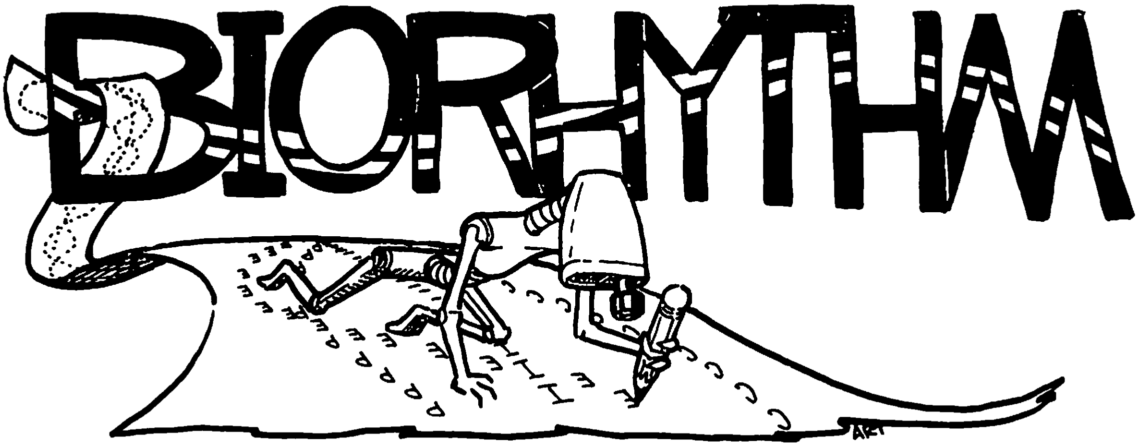
```
10 REM *******************
20 REM ***             ***
30 REM ***ANIMAL LEARNER***
40 REM ***             ***
50 REM *******************
60 GOSUB 100
70 GOSUB 300
80 GOSUB 420
90 END
100 REM
110 REM ***INSTRUCTIONS***
120 REM
130 GOSUB 260
140 PRINT "THIS IS A GAME TH
AT HAS THE ABILITY TO LEARN.
  IT WILL  ATTEMPT TO GUESS
THE NAME"
150 PRINT "OF AN ANIMAL THAT
 YOU PICK  AT RANDOM."
160 PRINT
170 PRINT "WHENEVER YOU STUM
P THE      COMPUTER, YOU ARE
 ASKED     ABOUT THE ANIMAL
YOU CHOSE."
180 PRINT "BY COMPILING THIS
 DATA, THE COMPUTER 'LEARNS'
."
190 PRINT
200 PRINT "ENTER 'STOP' WHEN
 YOU ARE   DONE."
210 FOR I=1 TO 6
220 PRINT
230 NEXT I
240 INPUT "PRESS ENTER WHEN
READY TO   CONTINUE: ":ANS$
250 RETURN
260 CALL CLEAR
270 PRINT TAB(4);"*** ANIMAL
 LEARNER ***"
280 PRINT
290 RETURN
300 REM
310 REM ***SETUP***
320 REM
330 CALL CLEAR
340 DIM QU$(50),RI(50),WR(50
),RA$(50),WA$(50)
350 QU$(1)="DOES IT ROAR"
360 RI(1)=0
370 WR(1)=0
380 RA$(1)="LION"
```



```
390 WA$(1)="GRIZZLY BEAR"
400 FR=2
410 RETURN
420 REM
430 REM ***PLAY***
440 REM
450 LI=1
460 GOSUB 260
470 FOR I=1 TO 10
480 PRINT
490 NEXT I
500 PRINT
510 PRINT "I KNOW";FR;"ANIMA
LS...."
520 PRINT
530 PRINT QU$(LI);
540 INPUT "?":ANS$
550 ANS$=SEG$(ANS$,1,1)
560 IF ANS$="Y" THEN 690
570 IF ANS$="N" THEN 740
580 IF ANS$="S" THEN 1340
590 A$="PLEASE ENTER 'YES' O
R 'NO'."
600 FOR I=1 TO LEN(A$)
610 CALL HCHAR(12,I+2,ASC(SE
G$(A$,I,1)))
620 NEXT I
630 FOR I=1 TO 300
640 NEXT I
650 FOR I=1 TO LEN(A$)
660 CALL HCHAR(16,I+2,32)
670 NEXT I
680 GOTO 540
690 IF RI(LI)<>0 THEN 720
700 GU$=RA$(LI)
710 GOTO 790
720 LI=RI(LI)
730 GOTO 520
740 IF WR(LI)<>0 THEN 770
750 GU$=WA$(LI)
760 GOTO 790
770 LI=WR(LI)
780 GOTO 520
790 PRINT
800 PRINT "IS IT A ";GU$;
810 INPUT "?":TA$
820 TA$=SEG$(TA$,1,1)
830 IF TA$="Y" THEN 1040
840 IF TA$="S" THEN 1340
850 PRINT
860 INPUT "WHAT WAS THE ANIM
AL? ":NA$
```

```
870 IF FR=51 THEN 1140
880 PRINT
890 PRINT "WHAT IS A QUESTIO
N I COULD  ASK TO TELL THE D
IFFERENCE  BETWEEN A ";GU$
900 PRINT "AND A ";NA$;
910 INPUT "? ":QU1$
920 PRINT
930 PRINT "FOR ";NA$;", THE"
940 PRINT "ANSWER IS WHAT";
950 INPUT "? ":YN$
960 YN$=SEG$(YN$,1,1)
970 IF (YN$<>"Y")*(YN$<>"N")
THEN 920
980 IF ANS$="Y" THEN 1230 EL
SE 1270
990 QU$(LI)=QU1$
1000 IF YN$="Y" THEN 1310
1010 RA$(LI)=GU$
1020 WA$(LI)=NA$
1030 GOTO 450
1040 FOR I=1 TO 200
1050 NEXT I
1060 CALL CLEAR
1070 A$="WOW!  I GOT IT!"
1080 FOR I=1 TO LEN(A$)
1090 CALL HCHAR(12,I+8,ASC(S
EG$(A$,I,1)))
1100 NEXT I
1110 FOR I=1 TO 300
1120 NEXT I
1130 GOTO 450
1140 CALL CLEAR
1150 PRINT "I CAN'T REMEMBER
 THAT ONE,"
1160 PRINT "MY MEMORY IS FUL
L."
1170 FOR I=1 TO 12
1180 PRINT
1190 NEXT I
1200 FOR I=1 TO 400
1210 NEXT I
1220 GOTO 450
1230 RI(LI)=FR
1240 LI=FR
1250 FR=FR+1
1260 GOTO 990
1270 WR(LI)=FR
1280 LI=FR
1290 FR=FR+1
1300 GOTO 990
```

```
1310 RA$(LI)=NA$
1320 WA$(LI)=GU$
1330 GOTO 450
1340 RETURN
```



This program is designed to interpret input, and from it, graph the user's biorhythms. You can accept the output to be as valid as you like, but don't expect the results to be testimony. Biorhythms, though fascinating, are still considered pseudo-science. There are several features of the program worth noting:

Lines 340-570 set up a simple calendar - printing routine. Lines 490-560 and 1430-1530 position the P, E and I characters into beautiful sine curves. Even if you have forgotten all your high school math you can try out the trig functions by plotting points to give a graphic representation of the function.

Did you recognize 6.28318531 as being 2*Pi?

Lines 660-670 ask for your birthdate and tell you exactly how to format the response. Lines 680-790 'error check' the input. Error checking is a vital part of so-called user - friendly programs.

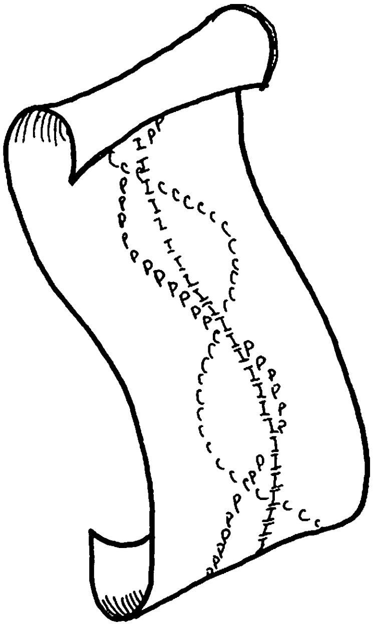
Lines 1650-1680 let you press any key to pause the printing and read the chart. A second keypress lets the printing continue.

Running Biorhythm in the TRACE mode will dramatically demonstate the FOR/NEXT loops in lines 1380-1720 that create the patterns.

The program uses complicated string functions and nested loops, so don't feel bad if it all looks like Relativity Theory. You can write a lot of programs which never get this complex.

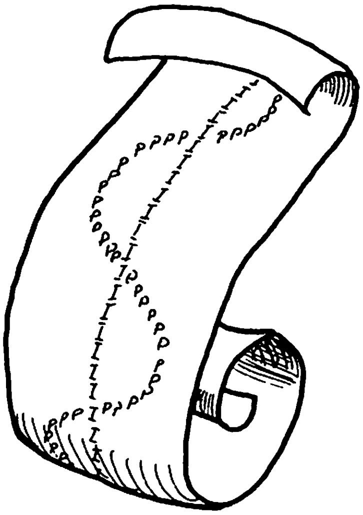
```
10 REM ********************
20 REM ***              ***
30 REM ***   BIORHYTHM  ***
40 REM ***              ***
50 REM ********************
60 GOSUB 100
70 GOSUB 340
80 GOSUB 580
90 END
100 REM
110 REM ***INSTRUCTIONS***
120 REM
130 GOSUB 300
140 PRINT "THIS PROGRAM WILL
 GRAPH OUT YOUR UNIQUE BIORH
YTHMIC     CYCLES."
150 PRINT
160 PRINT "A BIORHYTHMIC CAL
ENDAR IS   COMPOSED OF THE L
ETTERS I, EAND P.  EACH OF T
HESE REPRE-"
170 PRINT "SENTS ONE OF YOUR
 MEASURABLEBIORHYTHMIC CYCLE
S."
180 PRINT
190 PRINT
200 PRINT "   I = INTELLECTU
AL STATE"
210 PRINT
220 PRINT "   E = EMOTIONAL
STATE"
230 PRINT
240 PRINT "   P = PHYSICAL S
TATE"
250 FOR I=1 TO 4
260 PRINT
270 NEXT I
280 INPUT "PRESS ENTER WHEN
READY TO   CONTINUE: ":ANS$
290 RETURN
300 CALL CLEAR
310 PRINT TAB(7);"*** BIORHY
THM ***"
320 PRINT
330 RETURN
340 REM
350 REM ***SETUP***
360 REM
370 CALL CLEAR
380 SW=21
390 PI2=6.28318531
```

```
400 DIM A$(21),PL(23),EL(28)
,IL(33)
410 C$="JANFEBMARAPRMAYJUNJU
LAUGSEPOCTNOVDEC"
420 W$="SUNMONTUEWEDTHUFRISA
T"
430 FOR I=1 TO SW
440 A$(I)=" "
450 NEXT I
460 B=INT(SW/2)
470 A=B+1.5
480 PRINT "PLEASE WAIT";
490 FOR I=1 TO 33
500 IL(I)=INT(A+B*SIN(PI2*(I
-1)/33))
510 IF I>28 THEN 560
520 EL(I)=INT(A+B*SIN(PI2*(I
-1)/28))
530 PRINT ".";
540 IF I>23 THEN 560
550 PL(I)=INT(A+B*SIN(PI2*(I
-1)/23))
560 NEXT I
570 RETURN
580 REM
590 REM ***PLAY***
600 REM
610 GOSUB 300
620 PRINT "WHAT IS YOUR NAME
?"
630 INPUT " ";N$
640 ER=0
650 PRINT
660 PRINT "WHAT IS YOUR BIRT
HDATE?"
670 INPUT "(MM,DD,YY) ";M,D,
Y
680 IF (M>=1)*(M<=12)*(M=INT
(M))THEN 710
690 PRINT "INCORRECT MONTH"
700 ER=1
710 IF (D>=1)*(D<=31)*(D=INT
(D))THEN 740
720 PRINT "INCORRECT DAY"
730 ER=1
740 IF (Y>=0)*(Y=INT(Y))THEN
 770
750 PRINT "INCORRECT YEAR"
760 ER=1
770 IF ER=0 THEN 800
780 ER=0
```



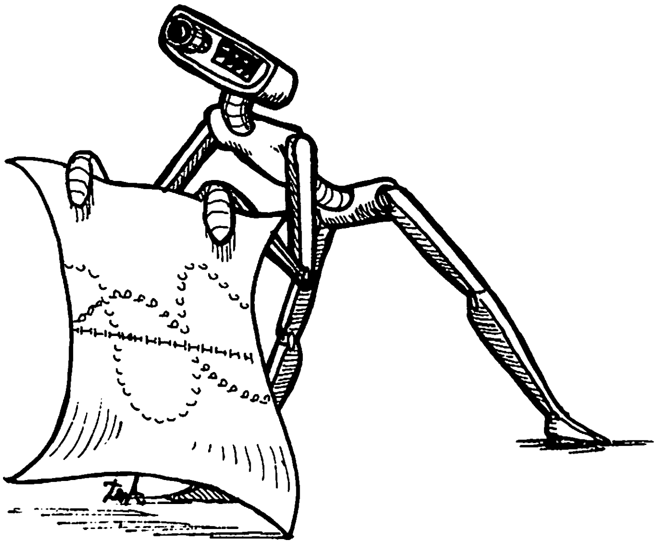
```
790 GOTO 650
800 IF Y>99 THEN 820
810 Y=Y+1900
820 GOSUB 1840
830 S=J
840 PRINT
850 PRINT "WHAT IS THE START
 DATE?"
860 INPUT "(MM,DD,YY) ":M1,D
1,Y1
870 IF (M1>=1)*(M1<=12)*(M1=
INT(M1))THEN 900
880 PRINT "INCORRECT MONTH"
890 ER=1
900 IF (D1>=1)*(D1<=31)*(D1=
INT(D1))THEN 930
910 PRINT "INCORRECT DAY"
920 ER=1
930 IF (Y1>=0)*(Y1=INT(Y1))T
HEN 960
940 PRINT "INCORRECT YEAR"
950 ER=1
960 IF ER=0 THEN 990
970 ER=0
980 GOTO 840
990 IF Y1>99 THEN 1010
1000 Y1=Y1+1900
1010 IF Y1>Y THEN 1060
1020 IF (Y1=Y)*(M1>M)THEN 10
60
1030 IF (Y1=Y)*(M1=M)*(D1>=D
)THEN 1060
1040 PRINT "START DATE BEFOR
E BIRTHDATE"
1050 GOTO 840
1060 PRINT
1070 INPUT "HOW MANY DAYS? "
:Z
1080 IF (Z<1)+(Z<>INT(Z))THE
N 1060
1090 CALL CLEAR
1100 PRINT "    BIORHYTHMIC
CALENDAR"
1110 PRINT "        --- FOR
---"
1120 PRINT TAB(INT((30-LEN(N
$))/2));N$
1130 PRINT "  BORN ON ";SEG$
(W$,3*N-2,3);D;SEG$(C$,3*M-2
,3);Y
1140 PRINT
```

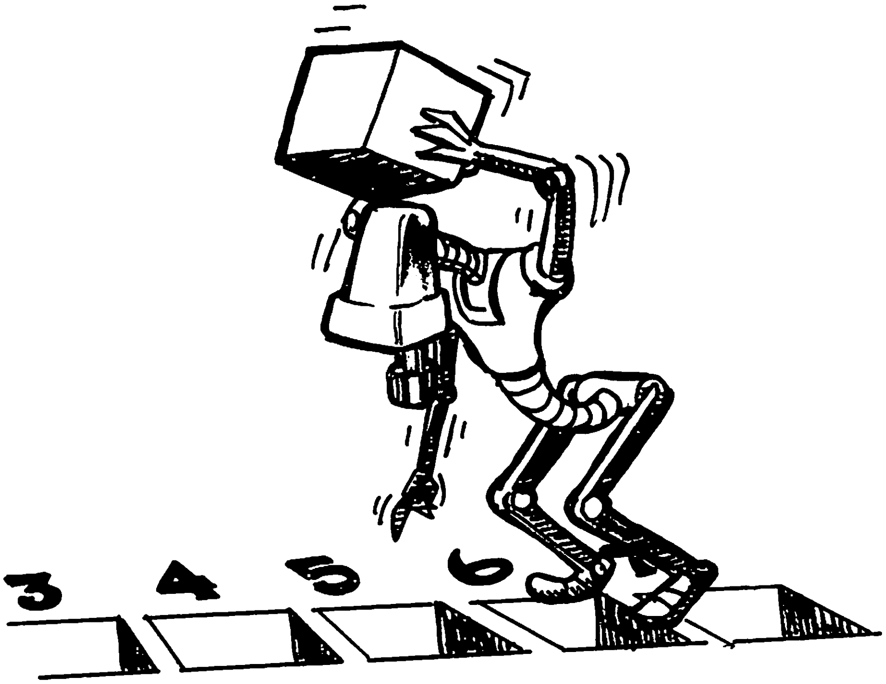
```
1150 PRINT
1160 PRINT "P=PHYSICAL
23 DAY CYCLE"
1170 PRINT "E=EMOTIONAL
28 DAY CYCLE"
1180 PRINT "I=INTELLECTUAL
33 DAY CYCLE"
1190 PRINT
1200 PRINT
1210 PRINT
1220 M=M1
1230 D=D1
1240 Y=Y1
1250 GOSUB 1840
1260 GOSUB 1980
1270 J2=J1
1280 M=M+1
1290 D=1
1300 GOSUB 1980
1310 LE=J1-J2
1320 IF LE<=Z THEN 1340
1330 LE=Z
1340 Z=Z-LE
1350 D1=D1-1
1360 F=J-S
1370 E=F+LE-1
1380 FOR K=F TO E
1390 D1=D1+1
1400 PK=K-INT(K/23)*23+1
1410 EK=K-INT(K/28)*28+1
1420 IK=K-INT(K/33)*33+1
1430 A$(PL(PK))="P"
1440 IF A$(EL(EK))<>" " THEN
 1470
1450 A$(EL(EK))="E"
1460 GOTO 1480
1470 A$(EL(EK))="*"
1480 IF A$(IL(IK))<>" " THEN
 1510
1490 A$(IL(IK))="I"
1500 GOTO 1520
1510 A$(IL(IK))="*"
1520 IF A$(B+1)<>" " THEN 15
40
1530 A$(B+1)="."
1540 IF (D1<>1)*(K<>F)THEN 1
560
1550 PRINT SEG$(C$,3*M1-2,3)
;Y;"-.......0.......+"
1560 PRINT SEG$(W$,3*N-2,3);
1570 IF D1>9 THEN 1590
```



```
1580 PRINT " ";
1590 PRINT D1;
1600 FOR I=1 TO SW
1610 PRINT A$(I);
1620 A$(I)=" "
1630 NEXT I
1640 PRINT
1650 CALL KEY(0,KEY,STAT)
1660 IF STAT=0 THEN 1690
1670 CALL KEY(0,KEY,STAT)
1680 IF STAT<>1 THEN 1670
1690 N=N+1
1700 IF N<8 THEN 1720
1710 N=1
1720 NEXT K
1730 D1=1
1740 M1=M1+1
1750 IF M1<13 THEN 1780
1760 M1=1
1770 Y1=Y1+1
1780 IF Z THEN 1220
1790 PRINT TAB(10);"-.......
0.......+"
1800 PRINT "DO YOU WANT ANOT
HER CHART?"
1810 INPUT ANS$
1820 IF SEG$(ANS$,1,1)="Y" T
HEN 580
1830 RETURN
1840 REM  FIND DAY OF WEEK
1850 IF M<3 THEN 1890
1860 M0=M-2
1870 Y0=Y
1880 GOTO 1910
1890 M0=M+10
1900 Y0=Y-1
1910 J=INT(Y0/100)
1920 K=Y0-(J*100)
1930 N=INT((13*M0-1)/5)+D+K+
INT(K/4)+INT(J/4)-2*J+77
1940 N=N-INT(N/7)*7+1
1950 M0=M0-1
1960 J=INT((146097*J)/4)+D+I
NT((1461*K)/4)+INT((153*M0+2
)/5)
1970 RETURN
1980 REM  FIND JULIAN DATE
1990 L=2
2000 IF M>=3 THEN 2030
2010 L=0
2020 GOTO 2070
```

```
2030 IF Y<>INT(Y/4)*4 THEN 2
070
2040 IF Y<>INT(Y/100)*100 TH
EN 2060
2050 IF Y<>INT(Y/400)*400 TH
EN 2070
2060 L=1
2070 J1=INT((3055*(M+2))/100
)-91+D-L
2080 RETURN
```





Connect Five

This classic game requires you to connect five squares, either vertically or horizontally. Though not much of a challenge, this game is good for your ego.

The graphics are fairly basic, so let's take a look. List 310-450. Lines 360 through 380 output the numbers 1-7 (which correspond to ASCII values 49-55). Lines 390-440 draw 176 asterisks (character 42). By running the program you will see that the gameboard is comprised of asterisks.

List 520-570. Lines 520-560 take a keyboard character (which you input), store its ASCII value in N, subtract an offset of 48 from N to change the ASCII value to a number, and check to see if N is between 1 and 7. Each time that a column is chosen, line 570 increments the Column Count (CC). Because there are seven columns, there are seven CC values (CC(1), CC(2), . . . CC(7)).

```
10 REM ********************
20 REM ***              ***
30 REM *** CONNECT FIVE ***
40 REM ***              ***
50 REM ********************
60 GOSUB 100
70 GOSUB 280
80 GOSUB 460
90 END
100 REM
110 REM ***INSTRUCTIONS***
120 REM
130 GOSUB 240
140 PRINT "THE OBJECT OF THE
 GAME IS TOGET FIVE PIECES I
N A ROW,   EITHER VERTICALLY
 OR        HORIZONTALLY."
150 PRINT
160 PRINT "WHEN IT IS YOUR T
URN TO     MOVE, ENTER THE N
UMBER OF   THE COLUMN YOU WO
ULD LIKE TOMOVE INTO."
170 PRINT
180 PRINT "AFTER YOU MOVE, T
HE COMPUTERWILL TAKE A TURN.
  THE FIRSTPLAYER TO CONNECT
 FIVE IS   THE WINNER."
190 FOR I=1 TO 6
200 PRINT
210 NEXT I
220 INPUT "PRESS ENTER WHEN
READY TO   CONTINUE :":ANS$
230 RETURN
240 CALL CLEAR
250 PRINT TAB(5);"*** CONNEC
T FIVE ***"
260 PRINT
270 RETURN
280 REM
290 REM ***SETUP***
300 REM
310 CALL CLEAR
320 RANDOMIZE
330 DIM GB(8,8),CC(7)
340 CALL COLOR(9,5,5)
350 CALL COLOR(10,7,7)
360 FOR I=1 TO 7
370 CALL HCHAR(3,9+2*I,48+I)
380 NEXT I
390 FOR Y=4 TO 18 STEP 2
400 CALL HCHAR(Y,10,42,15)
```

```
410 NEXT Y
420 FOR X=10 TO 24 STEP 2
430 CALL VCHAR(4,X,42,15)
440 NEXT X
450 RETURN
460 REM
470 REM ***PLAY***
480 REM
490 FOR I=1 TO 15
500 CALL HCHAR(21,9+I,ASC(SE
G$("YOUR MOVE (1-7)",I,1)))
510 NEXT I
520 CALL KEY(0,N,T)
530 IF T=0 THEN 520
540 N=N-48
550 IF N<1 THEN 1110
560 IF N>7 THEN 1110
570 CC(N)=CC(N)+1
580 IF CC(N)>7 THEN 1200
590 Z=CC(N)
600 GB(N,Z)=99
610 CALL HCHAR(19-2*Z,9+2*N,
99)
620 FOR I=1 TO 15
630 CALL HCHAR(21,9+I,32)
640 NEXT I
650 CG=INT(RND*7)+1
660 RG=INT(RND*4)+1
670 ON RG GOSUB 940,970,1020
,1070
680 FOR I=1 TO 7
690 IF CC(I)<7 THEN 710
700 FC=FC+1
710 NEXT I
720 IF FC=7 THEN 1440
730 FC=0
740 IF Z>7 THEN 650
750 GB(CG,Z)=1
760 CALL HCHAR(19-2*Z,9+2*CG
,111)
770 FOR X=1 TO 7
780 T1=0
790 T2=0
800 T3=0
810 T4=0
820 FOR Y=1 TO 7
830 IF GB(X,Y)=99 THEN 1300
840 IF GB(X,Y)=1 THEN 1330
850 T1=0
860 T2=0
870 IF GB(Y,X)=99 THEN 1360
```

```
880 IF GB(Y,X)=1 THEN 1390
890 T3=0
900 T4=0
910 NEXT Y
920 NEXT X
930 GOTO 490
940 CC(CG)=CC(CG)+1
950 Z=CC(CG)
960 RETURN
970 IF N=1 THEN 1030
980 CC(N-1)=CC(N-1)+1
990 CG=N-1
1000 Z=CC(N-1)
1010 RETURN
1020 IF N=7 THEN 980
1030 CC(N+1)=CC(N+1)+1
1040 CG=N+1
1050 Z=CC(N+1)
1060 RETURN
1070 CC(N)=CC(N)+1
1080 Z=CC(N)
1090 CG=N
1100 RETURN
1110 FOR I=1 TO 22
1120 CALL HCHAR(22,4+I,ASC(S
EG$("ENTER 1,2,3,4,5,6 OR 7"
,I,1)))
1130 NEXT I
1140 FOR I=1 TO 200
1150 NEXT I
1160 FOR I=1 TO 22
1170 CALL HCHAR(22,4+I,32)
1180 NEXT I
1190 GOTO 520
1200 CALL SOUND(250,320,2,18
0,2)
1210 FOR I=1 TO 22
1220 CALL HCHAR(22,4+I,ASC(S
EG$("THAT COLUMN IS FULL..."
,I,1)))
1230 NEXT I
1240 FOR I=1 TO 300
1250 NEXT I
1260 FOR I=1 TO 22
1270 CALL HCHAR(22,4+I,32)
1280 NEXT I
1290 GOTO 520
1300 T1=T1+99
1310 T2=0
1320 IF T1=495 THEN 1460 ELS
E 870
```

```
1330 T2=T2+1
1340 T1=0
1350 IF T2=5 THEN 1420 ELSE
870
1360 T3=T3+99
1370 T4=0
1380 IF T3=495 THEN 1460 ELS
E 910
1390 T4=T4+1
1400 T3=0
1410 IF T4=5 THEN 1420 ELSE
910
1420 PRINT "THE GAME, UNFORT
UNATELY, HASBEEN WON BY THE
COMPUTER.   BETTER LUCK NEXT
 TIME!"
1430 GOTO 1470
1440 PRINT "THAT'S CALLED A
CAT'S GAME. MAYBE NEXT TIME
YOU'LL WIN!"
1450 GOTO 1470
1460 PRINT "THE GAME IS OVER
, AND YOU   ARE THE WINNER!!
!"
1470 PRINT
1480 FOR I=1 TO 2000
1490 NEXT I
1500 RETURN
```

This is definitely a thinking man's game. The object is to identify a three-digit number (or code). You are given clues each time you make a guess.

Load the program. Lines 620-680 randomly create the code which you must find. Lines 630 and 640 give the first two digits in the code random integer values between 0 and 9. The rules specify that no two digits within the code will be equal. Therefore, 650 checks to see if N1 and N2 are equal. 660 gives the final digit in the code (N3) a value between 0 and 9. Lines 670 and 680 check to see if N3 is different from the other digits in the code.

You are allowed up to seven guesses. As with the code, each guess is comprised of three distinct digits. Lines 960, 990 and 1020 read the keyboard and check to see if a key has been pressed. The corresponding ASCII value of the guess is stored in G1, G2 or G3. Lines 1100-1120 convert the ASCII values of the three digits into their proper numeric values. This program is very straightforward. By scrutinizing and experimenting, you can ascertain the function of almost any line.

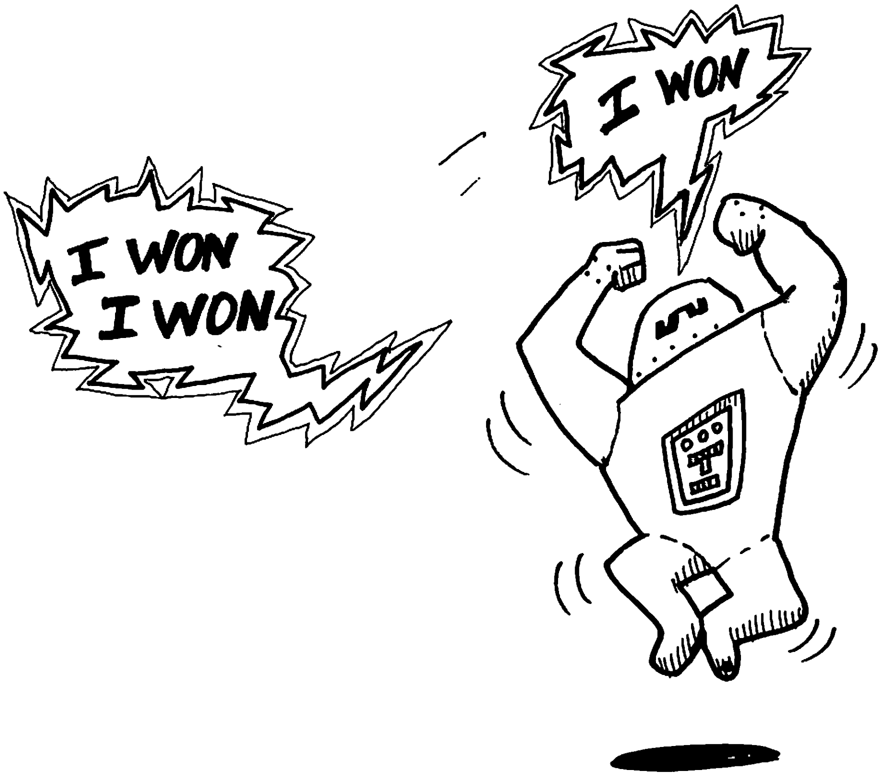
```
10 REM ********************
20 REM ***              ***
30 REM ***    DIGITS    ***
40 REM ***              ***
50 REM ********************
60 GOSUB 110
70 GOSUB 560
80 GOSUB 800
90 GOSUB 1620
100 END
110 REM
120 REM ***INSTRUCTIONS***
130 REM
140 GOSUB 520
150 PRINT "I WILL THINK OF A
 NUMBER     BETWEEN 012 AND 9
87.  EACH  DIGIT IN THE NUMB
ER WILL BE"
160 PRINT "DIFFERENT FROM TH
E OTHERS."
170 PRINT
180 PRINT "THE OBJECT OF THE
 GAME IS TOFIND THE MYSTERY
NUMBER IN  AS FEW GUESSES AS
 POSSIBLE."
190 FOR X=1 TO 12
200 PRINT
210 NEXT X
220 GOSUB 500
230 GOSUB 520
240 PRINT "AFTER EACH GUESS,
 YOU WILL  BE GIVEN A HINT L
INE."
250 PRINT
260 PRINT "FOR EACH DIGIT CO
RRECT AND  IN THE CORRECT PO
SITION, YOUWILL BE GIVEN AN
'X'."
270 PRINT
280 PRINT "FOR EACH DIGIT CO
RRECT BUT  NOT IN THE CORREC
T POSITION,YOU WILL BE GIVEN
 AN 'O'."
290 PRINT
300 PRINT "FOR EACH TOTALLY
INCORRECT  DIGIT YOU'LL GET
A HYPHEN."
310 FOR I=1 TO 7
320 PRINT
330 NEXT I
340 GOSUB 500
```

```
350 GOSUB 520
360 PRINT "PLAY WILL CONTINU
E UNTIL YOUGUESS THE ENTIRE
NUMBER OR  MAKE SEVEN WRONG
GUESSES."
370 PRINT
380 PRINT "THE TABLE BELOW S
HOULD HELP YOU UNDERSTAND TH
E HINTS."
390 PRINT
400 PRINT "  ANSWER  GUESS
HINT LINE"
410 PRINT "  ------  -----
---- ----"
420 PRINT "    065    703
   O--"
430 PRINT "    168    463
   X--"
440 PRINT "    987    536
   ---"
450 PRINT "    732    239
   XO-"
460 PRINT "    609    960
   OOO"
470 FOR I=1 TO 6
480 PRINT
490 NEXT I
500 INPUT "PRESS ENTER WHEN
READY TO   CONTINUE:=":ANS$
510 RETURN
520 CALL CLEAR
530 PRINT TAB(8);"*** DIGITS
 ***"
540 PRINT
550 RETURN
560 REM
570 REM ***SETUP***
580 REM
590 CALL CLEAR
600 RANDOMIZE
610 DIM G(3),M1$(27),M2$(19)
620 DEF R(X)=INT(RND*X)
630 N1=R(10)
640 N2=R(10)
650 IF N2=N1 THEN 640
660 N3=R(10)
670 IF N3=N2 THEN 660
680 IF N3=N1 THEN 660
690 NG=0
700 MM2$="WHAT IS YOUR GUESS
?"
```

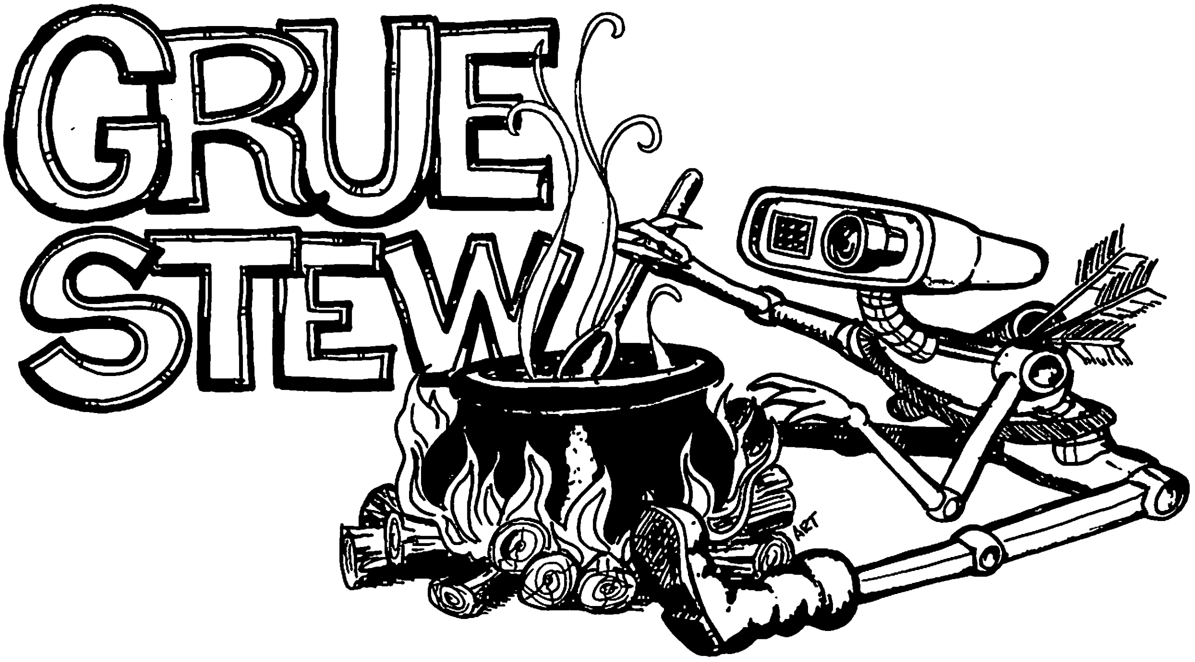
```
710 FOR I=1 TO 19
720 M2$(I)=SEG$(MM2$,I,1)
730 NEXT I
740 MM1$="TYPE THREE DIFFERE
NT DIGITS"
750 FOR I=1 TO 27
760 M1$(I)=SEG$(MM1$,I,1)
770 NEXT I
780 J=0
790 RETURN
800 REM
810 REM ***PLAY***
820 REM
830 GOSUB 520
840 PRINT
850 PRINT
860 PRINT "    I'VE GOT A NU
MBER..."
870 PRINT
880 PRINT " YOUR GUESS     H
INT LINE"
890 PRINT " ---- -----     -
--- ----"
900 FOR I=1 TO 14
910 PRINT
920 NEXT I
930 FOR I=1 TO 19
940 CALL HCHAR(23,3+I,ASC(M2
$(I)))
950 NEXT I
960 CALL KEY(0,G1,T)
970 IF (T=0)+(G1<48)+(G1>57)
THEN 960
980 CALL HCHAR(23,24,G1)
990 CALL KEY(0,G2,T)
1000 IF (T=0)+(T=-1)+(G2<48)
+(G2>57)THEN 990
1010 CALL HCHAR(23,25,G2)
1020 CALL KEY(0,G3,T)
1030 IF (T=0)+(T=-1)+(G3<48)
+(G3>57)THEN 1020
1040 CALL HCHAR(23,26,G3)
1050 IF (G1=G2)+(G1=G3)+(G2=
G3)THEN 1510
1060 CALL HCHAR(23,4,32,25)
1070 G(1)=G1
1080 G(2)=G2
1090 G(3)=G3
1100 G1=G1-48
1110 G2=G2-48
1120 G3=G3-48
```

```
1130 X=1
1140 IF G1<>N1 THEN 1160
1150 GOSUB 1430
1160 IF G2<>N2 THEN 1180
1170 GOSUB 1430
1180 IF G3<>N3 THEN 1200
1190 GOSUB 1430
1200 IF G1<>N2 THEN 1220
1210 GOSUB 1480
1220 IF G1<>N3 THEN 1240
1230 GOSUB 1480
1240 IF G2<>N1 THEN 1260
1250 GOSUB 1480
1260 IF G2<>N3 THEN 1280
1270 GOSUB 1480
1280 IF G3<>N1 THEN 1300
1290 GOSUB 1480
1300 IF G3<>N2 THEN 1320
1310 GOSUB 1480
1320 FOR I=X TO 3
1330 CALL HCHAR(12+NG,21+I,4
5)
1340 NEXT I
1350 FOR I=1 TO 3
1360 CALL HCHAR(12+NG,6+I,G(
I))
1370 CALL HCHAR(22,23+I,32)
1380 NEXT I
1390 NG=NG+1
1400 IF J=99 THEN 1590
1410 IF NG=7 THEN 1610
1420 GOTO 930
1430 CALL HCHAR(12+NG,21+X,8
8)
1440 IF X<3 THEN 1460
1450 J=99
1460 X=X+1
1470 RETURN
1480 CALL HCHAR(12+NG,21+X,7
9)
1490 X=X+1
1500 RETURN
1510 FOR I=1 TO 27
1520 CALL HCHAR(21,3+I,ASC(M
1$(I)))
1530 NEXT I
1540 FOR I=1 TO 200
1550 NEXT I
1560 CALL HCHAR(21,4,32,27)
1570 CALL HCHAR(23,24,32,3)
1580 GOTO 960
```

```
1590 FOR I=1 TO 200
1600 NEXT I
1610 RETURN
1620 REM
1630 REM ***END***
1640 REM
1650 IF J=99 THEN 1730
1660 N1=100*N1+10*N2+N3
1670 PRINT "THE CORRECT ANSW
ER WAS";N1
1680 FOR I=1 TO 14
1690 PRINT
1700 NEXT I
1710 PRINT "BETTER LUCK NEXT
 TIME!"
1720 GOTO 1790
1730 PRINT "     CONGRATULAT
IONS!!!"
1740 FOR I=1 TO 14
1750 PRINT
1760 NEXT I
1770 PRINT "YOU GUESSED IT I
N ONLY";NG
1780 PRINT "TRIES... EXCELLE
NT!"
1790 RETURN
```



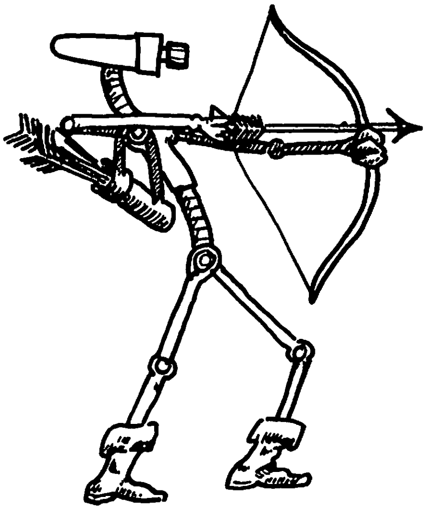
GRUE STEW



To play this game, you should have some paper and a pencil. You travel through an unseen maze of caverns searching for the Grue. As any good spelunker will tell you, drawing a map can prevent fruitless retracing. In other words, draw a map as you go along.

There are no color graphics used in this program. Still, this program incorporates a number of interesting features. You may ask, "Why is each run different?" Here's why. Line 1170 assigns a random value between one and twenty to each room (RO). The number corresponds to one of twenty room types. There are twenty DATA statements which can be read into any one of the twenty RO variables. According to line 1540, you have a 1:15 (one in fifteen) chance of experiencing an earthquake on any turn. There is nothing too sophisticated in this program, but if you should find an instruction whose meaning is unclear, change the line or omit it altogether. When the program is then executed, the absence of some attribute may clear up the uncertainty.

```
10 REM ********************
20 REM ***              ***
30 REM ***   GRUE STEW  ***
40 REM ***              ***
50 REM ********************
60 GOSUB 110
70 GOSUB 670
80 GOSUB 1240
90 GOSUB 2160
100 END
110 REM
120 REM ***INSTRUCTIONS***
130 REM
140 GOSUB 630
150 PRINT "IN THIS GAME. YOU
 ARE A      BRAVE HUNTER. YOU
 ARE ALSO   VERY HUNGRY. SO Y
OU'VE"
160 PRINT "DECIDED TO GO GRU
E HUNTING."
170 PRINT
180 PRINT "A GRUE, AS EVERYO
NE KNOWS,  IS THE KEY INGRED
IENT IN    GRUE STEW. YOU AR
E GOING TO"
190 PRINT "ENTER A SERIES OF
 UNDER-    GROUND CAVES IN S
EARCH OF   THE STEW BASE, TH
E GRUE."
200 PRINT
210 PRINT "IF YOU CAN BAG TH
E GRUE, ANDGET OUT OF THE CA
VES, THEN  YOU WILL GET YOUR
 STEW AND  WIN THE GAME!!"
220 FOR I=1 TO 4
230 PRINT
240 NEXT I
250 GOSUB 610
260 GOSUB 630
270 PRINT "ONCE IN THE MAZE,
 YOU CAN   EITHER MOVE TO A
DIFFERENT  CAVERN OR SHOOT A
N ARROW"
280 PRINT "INTO AN ADJOINING
 CAVE IN   HOPES OF HITTING
A FEROCIOUSGRUE."
290 PRINT
300 PRINT "I WILL ASK: MOVE
OR SHOOT?, AND YOU MUST REPL
Y 'M' FOR  MOVE OR 'S' FOR S
HOOT."
```



```
310 PRINT
320 PRINT "IF YOU DECIDE TO
MOVE, YOU  CAN DO SO IN ANY
OF THE FOURCOMPASS DIRECTION
S. WHEN"
330 PRINT "ASKED WHICH WAY,
ENTER 'N'  FOR NORTH, 'S' FO
R SOUTH,   'E' FOR EAST, OR
'W' FOR    WEST."
340 FOR I=1 TO 2
350 PRINT
360 NEXT I
370 GOSUB 610
380 GOSUB 630
390 PRINT "IF YOU DECIDE TO
SHOOT, YOU WILL BE ASKED WHI
CH WAY,    AND YOU MUST ENTE
R 'N' FOR"
400 PRINT "NORTH, 'S' FOR SO
UTH, 'E'   FOR EAST, OR 'W'
FOR WEST."
410 PRINT
420 PRINT "IF YOU HIT THE GR
UE, YOU    WILL BE TOLD AND
YOU MUST   TRY TO EXIT THE C
AVES."
430 PRINT
440 PRINT "BUT...THERE ARE O
THER THINGSIN THE CAVES. THE
RE ARE     GIANT BATS THAT W
ILL PICK"
450 PRINT "YOU UP AND DROP Y
OU ELSE    WHERE."
460 FOR I=1 TO 5
470 PRINT
480 NEXT I
490 GOSUB 610
500 GOSUB 630
510 PRINT "THERE ARE BOTTOML
ESS PITS.  IF YOU FALL INTO
ONE OF     THESE, YOU'LL NEV
ER GET OUT!"
520 PRINT
530 PRINT "OF COURSE THERE I
S THE GRUE HIMSELF. THOUGH N
OT AN      AGGRESSIVE CREATU
RE, HE WILL"
540 PRINT "EAT YOU IF YOU CO
ME TOO     CLOSE."
550 PRINT
560 PRINT "THERE ARE ALSO EA
```

```
RTHQUAKES  THAT MOVE THINGS
AROUND IN  THE CAVES (BATS,
PITS, THE"
570 PRINT "GRUE, AND THE EXI
T)."
580 FOR I=1 TO 6
590 PRINT
600 NEXT I
610 INPUT "PRESS ENTER WHEN
READY TO   CONTINUE: ":ANS$
620 RETURN
630 CALL CLEAR
640 PRINT TAB(7);"***GRUE ST
EW***"
650 PRINT
660 RETURN
670 REM
680 REM ***SETUP***
690 REM
700 CALL CLEAR
710 RANDOMIZE
720 DIM RO$(20),TR(20,4)
730 PRINT "PLEASE WAIT...";
740 FOR I=1 TO 20
750 READ RO$(I)
760 NEXT I
770 FOR I=11 TO 15
780 RO$(I)=RO$(10)
790 NEXT I
800 DATA YOU ARE IN A SMALL
ROOM WITHROCKS AND DEBRIS SC
ATTERED  EVERYWHERE.
810 DATA DUCK YOUR HEAD IN H
ERE; AS  LARGE ROCK STALACTI
TES HANG FROM THE CEILING.
820 DATA THE ROOM HERE SLOPE
S DOWN-   WARD.
830 DATA THE ROOM IS VERY SM
ALL; BUT I THINK WE CAN MAKE
 IT       THROUGH OK.
840 DATA THIS IS A VERY LARG
E ROOM   WITH A LARGE BOULDE
R IN THE CENTER OF IT.
850 DATA THIS IS THE CENTER
OF A     NARROW PASSAGE THAT
 CONNECTSOTHER ROOMS.
860 DATA THIS PASSAGE IS VER
Y LOW;   BUT IF WE CRAWL; WE
 CAN MAKEIT.
870 DATA THIS IS A VERY DIRT
Y ROOM;  IT HAS BEEN PARTIAL
```

```
LY FILLEDIN BY THE LAST EART
HQUAKE   THAT HIT.
880 DATA THIS ROOM IS ABOUT
AVERAGE  SIZE; BUT IS FILLED
 WITH A  PUNGENT AROMA THAT
IS VERY  NAUSEATING.
890 DATA YOU ARE IN A SMALL
PASSAGE- WAY.,,,,,
900 DATA A SMALL HOLE IN THE
 CEILING LETS LIGHT FROM OUT
SIDE     THROUGH,BUT YOU WOU
LD NEVERFIT THROUGH IT.
910 DATA SOMEONE HAS LEFT A
LIGHTED  TORCH ON THE WALL A
ND IT    ILLUMINATES YOUR PA
SSAGE.
920 DATA A RIVULET OF WATER
SLOWLY   TRICKLES FROM A HOL
E IN THE WALL.
930 DATA A SMALL HOLE TO YOU
R LEFT   ATTRACTS YOUR ATTEN
TION; BUTIT IS TOO SMALL TO
BE OF ANYCONCERN.
940 DATA YOU ARE IN A LOW DE
PRESSION IN THE CENTER OF A
MEDIUM-  SIZED ROOM.
950 DEF FNR(X)=INT(RND*X)+1
960 FOR I=1 TO 20
970 PRINT ".";
980 F=0
990 FOR J=1 TO 4
1000 GOSUB 1150
1010 F=F+TR(I,J)
1020 NEXT J
1030 IF F=0 THEN 990
1040 NEXT I
1050 YO=FNR(20)
1060 GU=FNR(20)
1070 IF GF=0 THEN 1090
1080 GU=-1
1090 EX=FNR(20)
1100 B1=FNR(20)
1110 B2=FNR(20)
1120 P1=FNR(20)
1130 P2=FNR(20)
1140 RETURN
1150 IF (FNR(3)=2)+(TR(I,J))
THEN 1160 ELSE 1170
1160 RETURN
1170 RO=FNR(20)
1180 IF RO=I THEN 1150
```

```
1190 DI=FNR(4)
1200 IF TR(RO,DI)THEN 1150
1210 TR(I,J)=RO
1220 TR(RO,DI)=I
1230 RETURN
1240 REM
1250 REM ***PLAY***
1260 REM
1270 CALL SCREEN(8)
1280 CALL CLEAR
1290 GOSUB 630
1300 FOR I=1 TO 15
1310 PRINT
1320 NEXT I
1330 PRINT RO$(YO)
1340 FOR I=1 TO 5
1350 CALL SOUND(100,262,5)
1360 FOR J=1 TO 5
1370 NEXT J
1380 NEXT I
1390 FOR I=1 TO 4
1400 CO=TR(YO,I)
1410 IF CO<>EX THEN 1440
1420 GOSUB 1610
1430 PRINT "EXIT NEARBY..."
1440 IF CO<>GU THEN 1470
1450 GOSUB 1610
1460 PRINT "I SMELL THE GRUE
!!!"
1470 IF (CO=B1)+(CO=B2)THEN
1480 ELSE 1500
1480 GOSUB 1610
1490 PRINT "FLAP...FLAP...FL
AP..."
1500 IF (CO=P1)+(CO=P2)THEN
1510 ELSE 1530
1510 GOSUB 1610
1520 PRINT "I FEEL A DRAFT!!
!"
1530 NEXT I
1540 IF FNR(15)<>4 THEN 1630
1550 GOSUB 1610
1560 PRINT
1570 PRINT "<<<<EARTHQUAKE>>
>>"
1580 PRINT
1590 GOSUB 1060
1600 GOTO 1330
1610 CALL SOUND(100,1760,2)
1620 RETURN
1630 PRINT
```

```
1640 INPUT "MOVE OR SHOOT:":
MS$
1650 IF (MS$="M")+(MS$="S")T
HEN 1680
1660 PRINT "TYPE IN 'M' OR '
S'..."
1670 GOTO 1640
1680 INPUT "WHICH WAY:":ANS$
1690 PRINT
1700 FOR I=1 TO 4
1710 IF ANS$=SEG$("NSEW",I,1
)THEN 1750
1720 NEXT I
1730 PRINT "ENTER 'N','S','E
','W'"
1740 GOTO 1640
1750 IF MS$="S" THEN 2000
1760 IF TR(YO,I)THEN 1810
1770 GOSUB 1610
1780 PRINT "YOU CAN'T GO THA
T WAY..."
1790 GOTO 1330
1800 PRINT
1810 PRINT "OK..."
1820 PRINT
1830 YO=TR(YO,I)
1840 IF YO<>EX THEN 1870
1850 WL=0
1860 RETURN
1870 IF YO<>GU THEN 1900
1880 WL=1
1890 RETURN
1900 IF (YO<>P1)+(YO<>P2)THE
N 1930
1910 WL=2
1920 RETURN
1930 IF (YO<>B1)+(YO<>B2)THE
N 1330
1940 PRINT "BATS HAVE YOU!!!
"

1950 PRINT "THEY'RE LIFTING
YOU UP!!!"
1960 PRINT "OOOOOH, WHERE AR
E WE NOW???"
1970 PRINT
1980 YO=FNR(20)
1990 GOTO 1330
2000 IF TR(YO,I)THEN 2050
2010 GOSUB 1610
2020 PRINT "CLUNK!!!"
2030 PRINT "THE ARROW BOUNCE
```

```
D OFF THE   WALL"
2040 GOTO 1330
2050 IF TR(YO,I)<>GU THEN 21
30
2060 GOSUB 1610
2070 PRINT "OUCH!!!!"
2080 PRINT "YOU BAGGED A GRU
E!!!!"
2090 PRINT "NOW TO FIND THE
WAY OUT!!!!"
2100 GF=1
2110 GU=-1
2120 GOTO 1330
2130 GOSUB 1610
2140 PRINT "THE ARROW MISSED
 THE GRUE!!!"
2150 GOTO 1330
2160 REM
2170 REM ***END***
2180 REM
2190 IF (WL=0)*(GF)THEN 2200
 ELSE 2230
2200 PRINT "YOU HAVE REACHED
 THE EXIT   WITH YOUR GRUE!!
! YOU WILL  HAVE A FILLING S
UPPER"
2210 PRINT "TONIGHT FOR SURE
!!!!"
2220 RETURN
2230 IF WL<>0 THEN 2260
2240 PRINT "YOU HAVE REACHED
 THE EXIT   WITHOUT ANY GRUE
!!!! YOU ARESURE TO STARVE!!
!!"
2250 RETURN
2260 IF WL<>1 THEN 2290
2270 PRINT "YOU BUMPED INTO
THE GRUE!!!!HE ATE YOU BEFOR
E YOU COULD MOVE!!!!"
2280 RETURN
2290 IF WL=2 THEN 2300
2300 PRINT "YOU FELL INTO A
PIT!!!!"
2310 PRINT "YOU FELL A LOOOO
OONG WAY..."
2320 RETURN
```

A clever premise overshadows an interesting game. IRSman serves as an excellent mathematical teaching device, while being both challenging and entertaining. Everyone wants to beat the IRS. This is your big chance!

To play, choose a number between 1 and 50 (we'll call it X). The digits 1 through X will be appear on the screen. Now, select a number from the list to be added to your total (we'll call it Y). But, and here's the real crux of the game, every number remaining on the list which is a factor of Y (divides evenly into it) is added to the coffers of the IRSman. Once a number has been used, it is removed from the list. The object is to garner as much money as possible (the numbers are considered money) while being as stingy as possible with your nemesis, the IRSman. Another important rule is that you must always pay. You may not claim a number on the list unless the IRSman gets something too. When none of the remaining numbers on the list has a factor remaining on the list (other than itself), the game is over, and all remaining numbers revert to the IRSman!

The following example should demonstrate the basis of good play. Assume that you are playing and have chosen to list numbers 1 through 6. The entire list will be displayed (1 2 3 4 5 6). If you opt to remove 6 first, then the IRSman scores for the remaining factors of 6. They are 1, 2 and 3. The IRSman scores six (1+2+3) and you score six (6). The revised list (4 5) contains numbers without factors in the list, so the game is over. These unused numbers are added to the IRSman's total. Final score: IRSman-15 You-6. You've been shellacked!

There are many keys to winning play. First, never waste 1, the universal factor. Your first choice for removal from the list should be the largest prime number. A prime number is only divisible by itself and 1. Examples are 1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31 and so on. To circumvent the loss of 5 in the above game, choose it first. After the first selection, the score would be: IRSman-1 You-5. The revised list would be (2 3 4 6). If you now chose 6, the IRSman would score 2 and 3 (2*3=6). Score: IRSman-6 You-11. The one remaining number, 4, reverts to the IRSman. Final score: IRSman-10 (1+2+3+4) You-11 (5+6). You win, but you could have done better. If your second selection was 4 instead of 6, then the IRSman would score 2 for a total of 3 (1+2), to your 9 (5+4). Now the list would be (3 6). Finally, 6 is removed from the list. The IRSman gets 3, but the game is over. Final score: IRSman-6 (1+2+3) You-15 (5+4+6). You've turned the tables and shellacked the IRSman! It is possible to win with almost any number list (1-2-3 being the exception) so try your luck.

```
10 REM ********************
20 REM ***              ***
30 REM ***    IRSMAN    ***
40 REM ***              ***
50 REM ********************
60 GOSUB 110
70 GOSUB 610
80 GOSUB 1000
90 GOSUB 2300
100 END
110 REM
120 REM ***INSTRUCTIONS***
130 REM
140 GOSUB 570
150 PRINT "THIS IS THE GAME
OF IRSMAN. TO WIN, YOU MUST
ACCUMULATE MORE MONEY THAN Y
OUR NEMESISTHE IRSMAN."
160 PRINT
170 PRINT "ENTER A NUMBER BE
TWEEN 1    AND 50.  YOU WILL
 BE GIVEN  A CONSECUTIVE NUM
BER STRING"
180 PRINT "STARTING AT 1 AND
 CONTINUINGTHROUGH TO THE NU
MBER YOU   SELECTED."
190 PRINT
200 PRINT "YOU WILL THEN CHO
OSE HOW    MUCH MONEY (WHICH
 NUMBER)   YOU WANT TO REMOV
E FROM THE LIST."
210 FOR I=1 TO 4
220 PRINT
230 NEXT I
```

```
240 GOSUB 550
250 GOSUB 570
260 PRINT "BUT, AND HERE'S T
HE FUN     PART, THE IRSMAN
GETS ALL OFTHE REMAINING NUM
BERS ON THE"
270 PRINT "LIST WHICH ARE FA
CTORS OF   THE NUMBER YOU CH
OSE.  THAT IS HOW THE IRSMAN
 GETS HIS  MONEY."
280 PRINT
290 PRINT "IF YOU CHOSE 6, F
OR EXAMPLE,THE IRSMAN GETS A
LL OF THE  REMAINING FACTORS
 OF 6"
300 PRINT "(POTENTIALLY 1, 2
 AND 3)."
310 FOR I=1 TO 8
320 PRINT
330 NEXT I
340 GOSUB 550
350 GOSUB 570
360 PRINT "YOU CANNOT REMOVE
 A NUMBER  THAT HAS NO REMAI
NING       FACTORS ON THE LI
ST, BECAUSE"
370 PRINT "YOU MUST ALWAYS P
AY THE IRS."
380 PRINT
390 PRINT "WHEN YOU CAN NO L
ONGER      REMOVE ANY OF THE
 REMAINING NUMBERS, THE UNUS
ED MONEY"
400 PRINT "(NUMBERS) IS RETU
RNED TO THECOFFER OF THE IRS
MAN."
410 FOR I=1 TO 10
420 PRINT
430 NEXT I
440 GOSUB 550
450 GOSUB 570
460 PRINT "YOU MUST ALWAYS M
AKE A TWO- DIGIT GUESS.  IF
YOUR GUESS IS BETWEEN 1 AND
9, THEN"
470 PRINT "TYPE A LEADING ZE
RO: 01, 02,03, 04, ... 08, 09."
480 PRINT
490 PRINT "WHEN YOU ARE THRO
UGH PLAYINGTHE GAME, PRESS Q
.  A FINAL TALLY WILL BE SHO
WN."
```

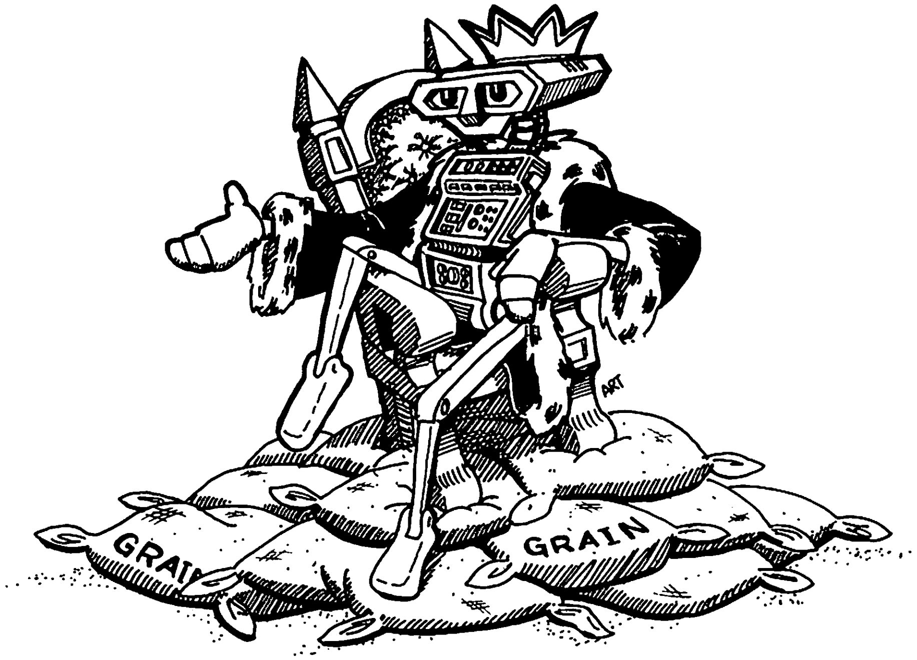
```
500 PRINT
510 PRINT "REMEMBER, Q FOR Q
UIT."
520 FOR I=1 TO 9
530 PRINT
540 NEXT I
550 INPUT "PRESS ENTER WHEN
READY TO   CONTINUE: ":ANS$
560 RETURN
570 CALL CLEAR
580 PRINT TAB(9);"*** IRSMAN
 ***"
590 PRINT
600 RETURN
610 REM
620 REM ***SETUP***
630 REM
640 CALL CLEAR
650 DIM LI(50),M1(18),M2(17)
,M3(28),M4(11)
660 PRINT "HOW MANY NUMBERS
DO YOU WANTIN THE LIST (1-50
) ?"
670 FOR I=1 TO 12
680 PRINT
690 NEXT I
700 INPUT A
710 IF A<0 THEN 700
720 IF A>50 THEN 700
730 IF A<>INT(A)THEN 700
740 FOR I=1 TO A
750 LI(I)=I
760 NEXT I
770 NU=A
780 Z=1
790 MM1$="WHICH DO YOU WANT? "
800 FOR I=1 TO 18
810 M1(I)=ASC(SEG$(MM1$,I,1))
820 NEXT I
830 LC=0
840 MM2$="IRSMAN:      YOU:"
850 FOR I=1 TO 17
860 M2(I)=ASC(SEG$(MM2$,I,1))
870 NEXT I
880 MM3$="YOU CAN'T CHOOSE T
HAT NUMBER"
890 FOR I=1 TO 28
900 M3(I)=ASC(SEG$(MM3$,I,1))
910 NEXT I
920 HS=0
930 YS=0
```

```
940 EOG=0
950 MM4$="FINAL SCORE"
960 FOR I=1 TO 11
970 M4(I)=ASC(SEG$(MM4$,I,1))
980 NEXT I
990 RETURN
1000 REM
1010 REM ***PLAY***
1020 REM
1030 GOSUB 570
1040 FOR I=1 TO 4
1050 PRINT
1060 NEXT I
1070 PRINT "    HERE IS THE
LIST:"
1080 PRINT
1090 PRINT
1100 FOR I=1 TO NU
1110 PRINT I;
1120 IF I=9 THEN 1160
1130 IF I-Z*7<>9 THEN 1180
1140 LC=LC+2
1150 Z=Z+1
1160 PRINT
1170 PRINT
1180 NEXT I
1190 IF NU>9 THEN 1210
1200 LC=-2
1210 FOR I=1 TO 12-LC
1220 PRINT
1230 NEXT I
1240 FOR I=1 TO 17
1250 CALL HCHAR(24,I+6,M2(I)
)
1260 NEXT I
1270 FOR I=1 TO 18
1280 CALL HCHAR(4,I+6,M1(I))
1290 NEXT I
1300 CALL KEY(0,G1,T)
1310 IF T=0 THEN 1750
1320 IF T=-1 THEN 1300
1330 IF G1=81 THEN 2190
1340 CALL HCHAR(4,26,G1)
1350 IF G1<48 THEN 1750
1360 IF G1>53 THEN 1750
1370 CALL KEY(0,G2,T)
1380 IF T=-1 THEN 1370
1390 IF T=0 THEN 1370
1400 CALL HCHAR(4,27,G2)
1410 IF G2<48 THEN 1780
1420 IF G2>57 THEN 1780
```

```
1430 G=(G1-48)*10+G2-48
1440 IF G=00 THEN 1750
1450 IF G>NU THEN 1750
1460 IF LI(G)=NU+1 THEN 2100
1470 FOR I=1 TO INT(G/2)
1480 IF G/LI(I)=INT(G/LI(I))
THEN 1920
1490 NEXT I
1500 IF GT=0 THEN 2100
1510 HS=HS+GT
1520 YS=YS+G
1530 IF HS>999 THEN 1800
1540 A=0
1550 IF HS>99 THEN 1830
1560 B=0
1570 IF HS>9 THEN 1860
1580 C=0
1590 GOTO 1890
1600 Q1=INT((G-3)/7)
1610 IF G<10 THEN 2170
1620 FOR I=4 TO 5
1630 CALL HCHAR(10+2*Q1,(G-Q
1*7-3)*4+I,32)
1640 NEXT I
1650 IF YS>99 THEN 2030
1660 A=0
1670 IF YS>9 THEN 2050
1680 B=0
1690 GOTO 2070
1700 IF EOG=1 THEN 2290
1710 LC=0
1720 CALL HCHAR(4,3,32,25)
1730 GT=0
1740 GOTO 1270
1750 CALL HCHAR(4,26,95)
1760 CALL HCHAR(4,27,95)
1770 GOTO 1300
1780 CALL HCHAR(4,27,95)
1790 GOTO 1370
1800 A=INT(HS/1000)
1810 CALL HCHAR(24,14,A+48)
1820 A=A*1000
1830 B=INT((HS-A)/100)
1840 CALL HCHAR(24,15,B+48)
1850 B=B*100+A
1860 C=INT((HS-B)/10)
1870 CALL HCHAR(24,16,C+48)
1880 C=C*10+B
1890 D=HS-C
1900 CALL HCHAR(24,17,D+48)
1910 GOTO 1600
```

```
1920 GT=GT+LI(I)
1930 IF I<10 THEN 2000
1940 Q=INT((LI(I)-3)/7)
1950 CALL HCHAR(10+2*Q,(LI(I
)-Q*7-3)*4+4,32)
1960 CALL HCHAR(10+2*Q,(LI(I
)-Q*7-3)*4+5,32)
1970 LI(I)=NU+1
1980 GOTO 1490
1990 IF T=0 THEN 1980
2000 CALL HCHAR(10,LI(I)*3+1
,32)
2010 LI(I)=NU+1
2020 GOTO 1490
2030 A=INT(YS/100)
2040 CALL HCHAR(24,25,A+48)
2050 B=INT((YS-(A*100))/10)
2060 CALL HCHAR(24,26,B+48)
2070 C=YS-A*100-B*10
2080 CALL HCHAR(24,27,C+48)
2090 GOTO 1700
2100 FOR I=1 TO 28
2110 CALL HCHAR(4,2+I,M3(I))
2120 NEXT I
2130 FOR I=1 TO 200
2140 NEXT I
2150 CALL HCHAR(4,3,32,28)
2160 GOTO 1270
2170 CALL HCHAR(10,G*3+1,32)
2180 GOTO 1650
2190 FOR I=1 TO NU
2200 IF LI(I)=NU+1 THEN 2220
2210 HS=HS+I
2220 NEXT I
2230 EOG=1
2240 FOR I=1 TO 11
2250 CALL HCHAR(23,I+10,M4(I
))
2260 NEXT I
2270 HS=HS-YS
2280 GOTO 1530
2290 RETURN
2300 REM
2310 REM ***END***
2320 REM
2330 FOR I=1 TO 500
2340 NEXT I
2350 CALL CLEAR
2360 IF YS>HS THEN 2450
2370 IF YS=HS THEN 2430
2380 PRINT "YOU HAVE BEEN BE
```

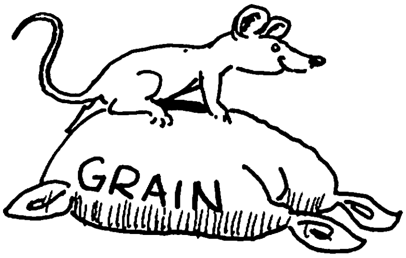
```
ATEN BY THE IRSMAN"
2390 PRINT
2400 PRINT
2410 PRINT "  BETTER LUCK NE
XT TIME!"
2420 GOTO 2490
2430 PRINT "INCREDIBLY, YOU
AND THE     IRSMAN EVENLY DI
VIDED THE   MONEY."
2440 GOTO 2490
2450 PRINT "     CONGRATULAT
IONS!!!"
2460 PRINT
2470 PRINT
2480 PRINT "YOU'VE DEFEATED
THE DREADED IRSMAN!"
2490 FOR I=1 TO 12
2500 PRINT
2510 NEXT I
2520 RETURN
```



Kingdom

This game tests your leadership ability. You are given a ten year reign, during which time you try to guide the kingdom towards greater health and prosperity. There are a number of uncontrollable variables used throughout the game. One of the uncontrollable variables is the price of acreage (in bushels of grain). The price varies between 17 and 26 bushels per acre. When the price of land is high (25 or 26 bushels per), it may be wise to trade in your overalls and become a land broker. That is, you may sell all but one acre of land (you must keep one), and hope that the price of land goes down. You must also beware of the ravenous rats. If you can rebuy the acreage at a reduced price, then you will have made a profit. When the price of land is low (below 20 bushels per acre), it is recommended that you buy as many acres as possible, while retaining enough grain to sow your fields and feed your people. You can better understand the program by altering the variables found in lines 370-460.

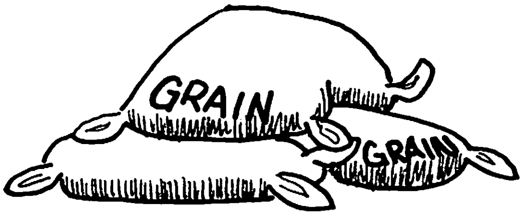
```
10 REM ********************
20 REM ***              ***
30 REM ***    KINGDOM   ***
40 REM ***              ***
50 REM ********************
60 GOSUB 110
70 GOSUB 310
80 GOSUB 520
90 GOSUB 1820
100 END
110 REM
120 REM ***INSTRUCTIONS***
130 REM
140 GOSUB 270
150 PRINT "THIS IS A SIMULAT
ION OF THE COUNTRY OF SUMERI
A.  YOU ARETHE SOVEREIGN RUL
ER, AND"
160 PRINT "YOUR REIGN WILL L
AST FOR TENYEARS."
170 PRINT
180 PRINT "THE DECISIONS YOU
 MAKE WILL AFFECT HUNDREDS O
F PEOPLE.  YOUR DICTATORIAL
SKILLS WILL"
190 PRINT "BE RATED AT THE E
ND OF YOUR RULE."
200 PRINT
210 PRINT "YOU WILL BE ASKED
 TO MAKE   SEVERAL KEY DECIS
IONS EACH  YEAR."
220 FOR I=1 TO 5
230 PRINT
240 NEXT I
250 INPUT "PRESS ENTER WHEN
READY TO   CONTINUE: ":ANS$
260 RETURN
270 CALL CLEAR
280 PRINT TAB(8);"*** KINGDO
M ***"
290 PRINT
300 RETURN
310 REM
320 REM ***SETUP***
330 REM
340 CALL CLEAR
350 RANDOMIZE
360 DIM NU$(11)
370 P=95
380 S=2800
390 H=3000
```



```
400 E=H-S
410 Y=3
420 A=H/Y
430 I=5
440 Q=1
450 D=0
460 Z=0
470 FOR X=1 TO 11
480 READ NU$(X)
490 NEXT X
500 DATA FIRST,SECOND,THIRD,
FOURTH,FIFTH,SIXTH,SEVENTH,E
IGHTH,NINTH,TENTH,ELEVENTH
510 RETURN
520 REM
530 REM ***PLAY***
540 REM
550 GOSUB 270
560 FOR X=1 TO 3
570 PRINT
580 NEXT X
590 Z=Z+1
600 PRINT "HAMURABI, I BEG T
O REPORT TOYOU:"
610 PRINT
620 PRINT "IN THE ";NU$(Z);"
 YEAR,";D;"PEOPLE";
630 PRINT "STARVED;";I;"CAME
 TO THE ";"CITY."
640 P=P+I
650 IF Q THEN 670
660 GOSUB 1380
670 PRINT
680 PRINT "THE POPULATION IS
";P;"AND"
690 PRINT "THE CITY OWNS";A;
"ACRES."
700 PRINT "YOU HARVESTED";Y;
"BUSHELS PER"
710 PRINT "ACRE. RATS ATE";E
;"BUSHELS."
720 PRINT "YOU HAVE";S;
730 PRINT "BUSHELS IN ";"RES
ERVE."
740 PRINT
750 PRINT
760 IF Z=11 THEN 1810
770 Y=INT(RND*10)+17
780 PRINT "LAND IS TRADING A
T";Y
790 PRINT "BUSHELS PER ACRE.
```

```
  HOW MANY ACRES DO YOU WISH
 TO BUY?"
800 INPUT Q
810 PRINT
820 IF Q<0 THEN 1460
830 IF Y*Q>S THEN 1490
840 IF Q>0 THEN 1520
850 INPUT "HOW MANY ACRES DO
 YOU WISH  TO SELL? ":Q
860 PRINT
870 IF Q<0 THEN 1560
880 IF Q=A THEN 1590
890 IF Q>A THEN 1610
900 A=A-Q
910 S=S+Y*Q
920 C=0
930 PRINT
940 PRINT "OF THE";S;"BUSHEL
S THAT"
950 INPUT "ARE LEFT, HOW MAN
Y DO YOU   WISH  TO FEED TO
YOUR       PEOPLE? ":Q
960 PRINT
970 IF Q<1 THEN 1650
980 IF Q=S THEN 1670
990 IF Q>S THEN 1690
1000 S=S-Q
1010 PRINT
1020 PRINT "OF THE";A;"ACRES
 YOU NOW"
1030 INPUT "OWN, HOW MANY DO
 YOU WISH TOPLANT WITH SEED?
 ":D
1040 PRINT
1050 IF D<1 THEN 1720
1060 IF D>A THEN 1740
1070 IF D/2>S THEN 1760
1080 IF D>10*P THEN 1780
1090 S=S-INT(D/2)
1100 Y=INT(RND*5)+1
1110 H=D*Y
1120 E=0
1130 IF INT(Y/2)*2=Y THEN 13
60
1140 S=S-E+H
1150 I=INT(Y*(20*A+S)/P/100+
1)
1160 C=INT(Q/20)
1170 Q=INT(10*(2*RND-.3))
1180 IF P<C THEN 1250
1190 D=P-C
```

```
1200 IF D>.5*P THEN 1270
1210 P1=((Z-1)*P1+D*100/P)/Z
1220 P=C
1230 D1=D1+D
1240 GOTO 550
1250 D=0
1260 GOTO 550
1270 GOSUB 270
1280 PRINT "YOU STARVED";D;"
PEOPLE IN"
1290 PRINT "ONE YEAR.  YOU H
AVE DONE    SUCH A MISERABLE
 JOB THAT"
1300 PRINT "YOU HAVE BEEN OV
ERTHROWN ANDEXILED TO A DESE
RTED ISLAND."
1310 FOR I=1 TO 13
1320 PRINT
1330 NEXT I
1340 WL=5
1350 GOTO 1810
1360 E=INT(S/Y)
1370 GOTO 1140
1380 CALL CLEAR
1390 PRINT "  A HORRIBLE PLA
GUE STRUCK!!"
1400 FOR I=1 TO 5
1410 PRINT
1420 NEXT I
1430 PRINT "HALF OF YOUR PEO
PLE DIED...."
1440 P=INT(P/2)
1450 RETURN
1460 PRINT "HAMURABI, YOU CA
N'T DO THAT,TO SELL LAND, FI
RST BUY ZEROACRES."
1470 PRINT
1480 GOTO 780
1490 PRINT "HAMURABI, THINK
AGAIN!"
1500 PRINT
1510 GOTO 780
1520 A=A+Q
1530 S=S-Q*Y
1540 C=0
1550 GOTO 930
1560 PRINT "HAMURABI, YOU CA
N'T DO THAT.IF YOU DO NOT WI
SH TO SELL, THEN SELL ZERO A
CRES."
1570 PRINT
```



```
1580 GOTO 850
1590 PRINT "HAMURABI, YOU MU
ST KEEP AT  LEAST ONE ACRE O
F LAND!"
1600 GOTO 850
1610 PRINT "HAMURABI, YOU ON
LY OWN";A
1620 PRINT "ACRES."
1630 PRINT
1640 GOTO 850
1650 PRINT "HAMURABI, THE PE
OPLE WILL   STARVE! YOU MUST
 FEED THEM  SOMETHING."
1660 GOTO 930
1670 PRINT "HAMURABI, YOU MU
ST KEEP AT  LEAST ONE BUSHEL
 TO PLANT."
1680 GOTO 930
1690 PRINT "HAMURABI, YOU ON
LY OWN";S
1700 PRINT "BUSHELS."
1710 GOTO 930
1720 PRINT "HAMURABI, YOU MU
ST PLANT    SOMETHING SO THE
RE WILL BE  FOOD FOR NEXT YE
AR."
1730 GOTO 1010
1740 PRINT "YOU ONLY HAVE";A
;"ACRES!"
1750 GOTO 1010
1760 PRINT "HAMURABI, THAT I
S TOO MUCH  TO PLANT."
1770 GOTO 1010
1780 PRINT "YOU CAN ONLY FOR
CE ONE MAN  TO WORK 10 ACRES
 OF LAND.   YOUR POPULATION
OF";P
1790 PRINT "JUST ISN'T BIG E
NOUGH."
1800 GOTO 1010
1810 RETURN
1820 REM
1830 REM ***END***
1840 REM
1850 IF WL=5 THEN 2090
1860 INPUT "PRESS ENTER TO S
EE YOUR     RATING.":A$
1870 GOSUB 270
1880 PRINT "IN YOUR 10 YEARS
 OF RULE,   ";P1;"PERCENT OF
 THE"
```

```
1890 PRINT "POPULATION STARV
ED PER YEAR (ON THE AVERAGE)
; A TOTAL OF";D1;"PEOPLE DIE
D."
1900 L=A/P
1910 PRINT
1920 PRINT "YOU STARTED WITH
 10 ACRES   PER PERSON AND Y
OU ENDED    WITH";L;"ACRES P
ER PERSON."
1930 IF P1>33 THEN 2010
1940 IF L<7 THEN 2010
1950 IF P1>10 THEN 2030
1960 IF L<9 THEN 2030
1970 IF P1>3 THEN 2050
1980 IF L<11 THEN 2050
1990 PRINT "A TRULY INSPIRED
 JOB.  THE  PEOPLE LOVE AND
ADMIRE YOU."
2000 GOTO 2060
2010 PRINT "YOU ARE A DISGRA
CE!!!  THE  PEOPLE CHEERED W
HEN YOU LEFTOFFICE."
2020 GOTO 2060
2030 PRINT "YOU RULE LIKE AT
ILLA THE     HUN!!!  MOST OF
YOUR         SUBJECTS WOULD D
ANCE AT YOUR FUNERAL."
2040 GOTO 2060
2050 PRINT "YOU COULD HAVE D
ONE BETTER. FEW PEOPLE CARE
TO SEE YOU  RULE AGAIN."
2060 FOR I=1 TO 5
2070 PRINT
2080 NEXT I
2090 RETURN
```

In this game you try to alphabetize a matrix of scrambled letters. One square is empty to allow movement into it. The object of the game is to unscramble the letters in as few tries as possible. Remember, practice makes perfect!

Before looking at the grapics, it seems appropriate to mention that proper graphics alignment is rarely a matter of trial and error, but entails working with graph paper and plotting the proposed screen output. Then, after figuring out which points to plot (and with which color), writing the program can take place. The beautiful pictures you see on the screen are the result of painstaking work rather than of brilliance.

Getting back to MAGIC SQUARES, let's look at some of the graphics. Line 510 draws the letters. Capital letters have ASCII values beginning at 65 (A) and ending at 90 (Z). To confirm this, copy and run the following program:

```
10 CALL CLEAR
20 FOR X=1 TO 26
30 CALL HCHAR(12,2+X,64+X)
40 NEXT X
```

```
10 REM ********************
20 REM ***              ***
30 REM *** MAGIC SQUARES***
40 REM ***              ***
50 REM ********************
60 GOSUB 110
70 GOSUB 340
80 GOSUB 890
90 GOSUB 1540
100 END
110 REM
120 REM ***INSTRUCTIONS***
130 REM
140 GOSUB 300
150 PRINT "IN THIS GAME YOU
HAVE A     FOUR BY FOUR GAME
BOARD THAT CONTAINS THE LETT
ERS A-O."
160 PRINT
170 PRINT "THE OBJECT IS TO
HORIZONTAL-LY ALPHABETIZE TH
E SCRAMBLEDLETTERS.  YOU CAN
 MOVE A"
180 PRINT "PIECE SIDEWAYS OR
 UP AND    DOWN, AS LONG AS
THE EMPTY  SQUARE IS NEXT TO
 IT."
190 PRINT
200 PRINT "THE KEY IS TO THI
NK AHEAD!"
210 PRINT
220 PRINT "YOU WILL BE SHOWN
 AN ALPHA- BETIZED GAMEBOARD
 BEFORE    THE SCRAMBLING TA
KES PLACE."
230 PRINT
240 PRINT "TO QUIT THE GAME,
 ENTER Q."
250 FOR I=1 TO 2
260 PRINT
270 NEXT I
280 INPUT "PRESS ENTER WHEN
READY TO   CONTINUE: ":ANS$
290 RETURN
300 CALL CLEAR
310 PRINT TAB(5);"*** MAGIC
SQUARES ***"
320 PRINT
330 RETURN
340 REM
350 REM ***SETUP***
```

```
360 REM
370 CALL CLEAR
380 RANDOMIZE
390 DIM GB(5,5),M1(26)
400 CALL COLOR(9,5,5)
410 CALL COLOR(10,3,3)
420 REM  DRAW GAMEBOARD
430 PRINT "TYPE THE LETTER Y
OU WISH TO MOVE.  TYPE Q TO
QUIT."
440 FOR X=5 TO 13
450 CALL HCHAR(X,14,96,9)
460 NEXT X
470 FOR X=2 TO 8 STEP 2
480 FOR Y=2 TO 8 STEP 2
490 A=A+1
500 IF A=16 THEN 520
510 CALL HCHAR(4+X,13+Y,64+A
)
520 NEXT Y
530 NEXT X
540 CALL HCHAR(12,21,32)
550 A=0
560 FOR I=1 TO 4
570 FOR J=1 TO 4
580 A=A+1
590 GB(I,J)=64+A
600 NEXT J
610 NEXT I
620 GB(4,4)=32
630 REM  SCRAMBLE THE BOARD

640 FOR I=1 TO 4
650 FOR J=1 TO 4
660 X=INT(RND*4)+1
670 Y=INT(RND*4)+1
680 T=GB(I,J)
690 GB(I,J)=GB(X,Y)
700 GB(X,Y)=T
710 NEXT J
720 NEXT I
730 FOR I=1 TO 4
740 FOR J=1 TO 4
750 CALL HCHAR(4+2*I,13+2*J,
GB(I,J))
760 NEXT J
770 NEXT I
780 MM1$="THAT LETTER CAN'T
BE MOVED"
790 FOR I=1 TO 26
800 M1(I)=ASC(SEG$(MM1$,I,1)
)
```

```
810 NEXT I
820 FOR I=0 TO 5 STEP 5
830 FOR J=1 TO 4
840 GB(I,J)=99
850 GB(J,I)=99
860 NEXT J
870 NEXT I
880 RETURN
890 REM
900 REM ***PLAY***
910 REM
920 CALL KEY(0,N,T)
930 IF T=0 THEN 920
940 IF N=81 THEN 1530
950 IF (N<65)+(N>79)THEN 920
960 X1=1
970 Y1=1
980 IF GB(X1,Y1)=32 THEN 1030
990 Y1=Y1+1
1000 IF Y1<5 THEN 980
1010 X1=X1+1
1020 GOTO 970
1030 IF GB(X1+1,Y1)<>N THEN
1090
1040 T=GB(X1+1,Y1)
1050 GB(X1+1,Y1)=32
1060 GB(X1,Y1)=T
1070 CX=1
1080 GOTO 1260
1090 IF GB(X1-1,Y1)<>N THEN
1150
1100 T=GB(X1-1,Y1)
1110 GB(X1-1,Y1)=32
1120 GB(X1,Y1)=T
1130 CX=-1
1140 GOTO 1260
1150 IF GB(X1,Y1+1)<>N THEN
1210
1160 T=GB(X1,Y1+1)
1170 GB(X1,Y1+1)=32
1180 GB(X1,Y1)=T
1190 CY=1
1200 GOTO 1260
1210 IF GB(X1,Y1-1)<>N THEN
1410
1220 T=GB(X1,Y1-1)
1230 GB(X1,Y1-1)=32
1240 GB(X1,Y1)=T
1250 CY=-1
1260 CALL HCHAR(4+2*X1,13+2*
Y1,GB(X1,Y1))
```

```
1270 CALL HCHAR(4+2*(X1+CX),
13+2*(Y1+CY),GB(X1+CX,Y1+CY))
1280 CX=0
1290 CY=0
1300 A=0
1310 FOR I=1 TO 4
1320 FOR J=1 TO 4
1330 IF GB(I,J)<>65+A THEN 1
350
1340 TC=TC+1
1350 A=A+1
1360 NEXT J
1370 NEXT I
1380 IF (TC=15)*(GB(4,4)=32)
THEN 1500
1390 TC=0
1400 GOTO 920
1410 FOR I=1 TO 26
1420 CALL HCHAR(20,I+2,M1(I))
1430 NEXT I
1440 FOR I=1 TO 100
1450 NEXT I
1460 FOR I=1 TO 26
1470 CALL HCHAR(20,I+2,32)
1480 NEXT I
1490 GOTO 920
1500 W=1
1510 FOR I=1 TO 400
1520 NEXT I
1530 RETURN
1540 REM
1550 REM ***END***
1560 REM
1570 CALL CLEAR
1580 IF W<>1 THEN 1650
1590 PRINT TAB(9);"YOU DID I
T!!!"
1600 FOR I=1 TO 12
1610 PRINT
1620 NEXT I
1630 PRINT "YOU HAVE DISPLAY
ED BRAINS,  SKILL, AND FORTI
TUDE!!"
1640 GOTO 1690
1650 PRINT "SORRY YOU HAD TO
 QUIT EARLY.BETTER LUCK NEXT
 TIME."
1660 FOR I=1 TO 12
1670 PRINT
1680 NEXT I
1690 RETURN
```

Numbers Away

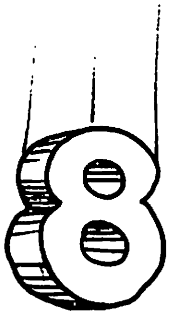
This game is based on a popular game show. The object is to eliminate as many numbers from the list as you can before getting stymied. To begin, you are given a series of numbers from 1 to 9. A pair of dice are rolled. The total (2-12) of the dice must then be subtracted from the list. Most of the numbers can be removed from the list in a multitude of ways. If the first number rolled is 10, there are ten possible ways to total exactly 10. They are:

1,2,3,4; 1,2,7; 1,3,6; 1,4,5; 1,9; 2,3,5; 2,8; 3,7 and 4,6.

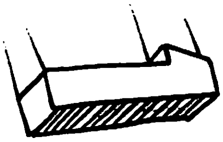
According to the rules, you can remove any combination which equals the total on the dice. Once you have used a number, it is removed from the list.

Let's look at the graphics. Line 590 sets all characters in Character Set #13 to gray (color=15). Lines 650-720 draw the dice. The program goes on to use sophisticated string functions and nested loops, so if it appears confusing, don't be disheartened.

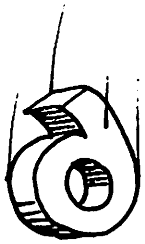
```
10 REM ********************
20 REM ***              ***
30 REM *** NUMBERS AWAY ***
40 REM ***              ***
50 REM ********************
60 GOSUB 110
70 GOSUB 400
80 GOSUB 840
90 GOSUB 2020
100 END
110 REM
120 REM ***INSTRUCTIONS***
130 REM
140 GOSUB 360
150 PRINT "IN THIS GAME YOU
WILL BE     PRESENTED WITH A
SERIES OF  NUMBERS BETWEEN 1
 AND 9."
160 PRINT
170 PRINT "A PAIR OF DICE WI
LL BE      ROLLED, AND THE T
OTAL WILL  BE NOTED.  YOU MU
ST REMOVE"
180 PRINT "FROM THE LIST A C
OMBINATION OF NUMBERS WHOSE
SUM MATCHESTHE TOTAL ON THE
DICE."
190 FOR I=1 TO 10
200 PRINT
210 NEXT I
220 GOSUB 340
230 GOSUB 360
240 PRINT "FOR EXAMPLE, IF A
 SEVEN WAS ROLLED, YOU COULD
 REMOVE    FROM THE LIST (1,
2,4),"
250 PRINT "(1,6), (2,5), (3,
4) OR JUST PLAIN (7)."
260 PRINT
270 PRINT "WHEN YOU SELECT E
NOUGH      NUMBERS TO REACH
THE TOTAL  ON THE DICE, THE
COMPUTER"
280 PRINT "WILL AGAIN ROLL T
HE DICE FORYOUR NEXT TRY."
290 PRINT
300 PRINT "IF THE NUMBERS YO
U CHOOSE   EXCEED THE TOTAL
ON THE     DICE, THEN THE LI
ST WILL BE RESTORED."
310 FOR I=1 TO 4
```



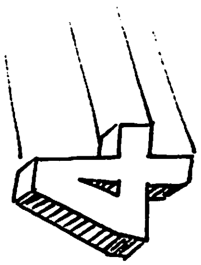
```
320 PRINT
330 NEXT I
340 INPUT "PRESS ENTER WHEN
READY TO   CONTINUE: ":ANS$
350 RETURN
360 CALL CLEAR
370 PRINT TAB(5);"*** NUMBER
S AWAY ***"
380 PRINT
390 RETURN
400 REM
410 REM ***SETUP***
420 REM
430 CALL CLEAR
440 DIM L(9),M1(16),M2(20),Y
G(5)
450 FOR I=1 TO 9
460 L(I)=I
470 PRINT I;
480 NEXT I
490 FOR I=1 TO 4
500 PRINT
510 NEXT I
520 PRINT " PRESS THE NUMBER
S YOU          CHOOSE BUT WAIT
UNTIL EACH    ONE DISAPPEARS B
EFORE          CONTINUING"
530 PRINT
540 PRINT TAB(12);"PRESS Q T
O QUIT"
550 FOR I=1 TO 8
560 PRINT
570 NEXT I
580 CALL COLOR(9,5,5)
590 CALL COLOR(13,15,15)
600 CALL COLOR(14,13,13)
610 CALL VCHAR(2,3,96,22)
620 CALL VCHAR(2,30,96,22)
630 CALL HCHAR(2,4,96,26)
640 CALL HCHAR(23,4,96,26)
650 CALL HCHAR(18,5,130,5)
660 CALL VCHAR(19,5,130,3)
670 CALL HCHAR(22,5,130,5)
680 CALL VCHAR(19,9,130,3)
690 CALL HCHAR(18,11,130,5)
700 CALL VCHAR(19,11,130,3)
710 CALL HCHAR(22,11,130,5)
720 CALL VCHAR(19,15,130,3)
730 M$="THAT'S TOO MUCH!"
740 FOR I=1 TO 16
750 M1(I)=ASC(SEG$(M$,I,1))
```



```
760 NEXT I
770 M$="THAT'S NOT AVAILABLE
"
780 FOR I=1 TO 20
790 M2(I)=ASC(SEG$(M$,I,1))
800 NEXT I
810 GT=0
820 Z=0
830 RETURN
840 REM
850 REM ***PLAY***
860 REM
870 GOSUB 1110
880 CALL KEY(0,A,T)
890 IF ((T=0)+(T=-1))THEN 88
0
900 IF A=81 THEN 1970
910 A=A-48
920 IF (A<1)+(A>9)THEN 880
930 IF L(A)=0 THEN 1020
940 Z=Z+1
950 YG(Z)=A
960 GT=GT+A
970 IF GT>DD THEN 1650
980 CALL HCHAR(6,3*A+1,32)
990 L(A)=0
1000 IF GT=DD THEN 1770
1010 GOTO 880
1020 FOR I=1 TO 20
1030 CALL HCHAR(16,I+4,M2(I)
)
1040 NEXT I
1050 FOR I=1 TO 200
1060 NEXT I
1070 FOR I=1 TO 20
1080 CALL HCHAR(16,I+4,32)
1090 NEXT I
1100 GOTO 880
1110 REM
1120 REM  ROLL DEM BONES!
1130 REM
1140 FOR I=19 TO 21
1150 CALL HCHAR(I,6,32,3)
1160 CALL HCHAR(I,12,32,3)
1170 NEXT I
1180 RANDOMIZE
1190 D1=INT(RND*6+1)
1200 D2=INT(RND*6+1)
1210 ON D1 GOSUB 1250,1270,1
300,1330,1380,1410
1220 ON D2 GOSUB 1450,1470,1
```



```
500,1530,1580,1610
1230 DD=D1+D2
1240 RETURN
1250 CALL HCHAR(20,7,42)
1260 RETURN
1270 CALL HCHAR(19,7,42)
1280 CALL HCHAR(21,7,42)
1290 RETURN
1300 GOSUB 1250
1310 GOSUB 1270
1320 RETURN
1330 CALL HCHAR(19,6,42)
1340 CALL HCHAR(19,8,42)
1350 CALL HCHAR(21,6,42)
1360 CALL HCHAR(21,8,42)
1370 RETURN
1380 GOSUB 1250
1390 GOSUB 1330
1400 RETURN
1410 GOSUB 1330
1420 CALL HCHAR(20,6,42)
1430 CALL HCHAR(20,8,42)
1440 RETURN
1450 CALL HCHAR(20,13,42)
1460 RETURN
1470 CALL HCHAR(19,13,42)
1480 CALL HCHAR(21,13,42)
1490 RETURN
1500 GOSUB 1450
1510 GOSUB 1470
1520 RETURN
1530 CALL HCHAR(19,12,42)
1540 CALL HCHAR(19,14,42)
1550 CALL HCHAR(21,12,42)
1560 CALL HCHAR(21,14,42)
1570 RETURN
1580 GOSUB 1450
1590 GOSUB 1530
1600 RETURN
1610 GOSUB 1530
1620 CALL HCHAR(20,12,42)
1630 CALL HCHAR(20,14,42)
1640 RETURN
1650 FOR I=1 TO 16
1660 CALL HCHAR(16,I+4,M1(I)
)
1670 NEXT I
1680 FOR I=1 TO 200
1690 NEXT I
1700 FOR I=1 TO 16
1710 CALL HCHAR(16,I+4,32)
```



```
1720 NEXT I
1730 GOSUB 1970
1740 GT=0
1750 Z=0
1760 GOTO 880
1770 FOR I=1 TO 200
1780 NEXT I
1790 J=0
1800 FOR I=1 TO 9
1810 IF L(I)=I THEN 1830
1820 J=J+1
1830 NEXT I
1840 IF J>=9 THEN 1880
1850 GT=0
1860 Z=0
1870 GOTO 840
1880 FOR I=1 TO 200
1890 NEXT I
1900 CALL CLEAR
1910 PRINT TAB(6);"CONGRATUL
ATIONS!!!"
1920 FOR I=1 TO 12
1930 PRINT
1940 NEXT I
1950 YW=1
1960 GOTO 2010
1970 FOR I=1 TO Z
1980 CALL HCHAR(6,3*YG(I)+1,
YG(I)+48)
1990 L(YG(I))=YG(I)
2000 NEXT I
2010 RETURN
2020 REM
2030 REM ***END***
2040 REM
2050 IF YW=1 THEN 2390
2060 GT=0
2070 FOR I=1 TO 9
2080 IF L(I)=I THEN 2100
2090 GT=GT+I
2100 NEXT I
2110 GOSUB 360
2120 FOR I=1 TO 3
2130 PRINT
2140 NEXT I
2150 PRINT "OUT OF A POSSIBL
E 45, YOU   GOT";GT;"OFF THE
 BOARD."
2160 PRINT
2170 PRINT "THAT IS ";
2180 ON INT(GT/5)GOSUB 2230,
2250,2270,2290,2310,2330,235
0,2370
2190 FOR I=1 TO 10
2200 PRINT
2210 NEXT I
2220 GOTO 2450
2230 PRINT "INCREDIBLY BAD L
UCK!"
2240 RETURN
2250 PRINT "EXTREMELY POOR!"
2260 RETURN
2270 PRINT "TERRIBLE!"
2280 RETURN
2290 PRINT "VERY POOR..."
2300 RETURN
2310 PRINT "JUST SO-SO."
2320 RETURN
2330 PRINT "PRETTY GOOD!"
2340 RETURN
2350 PRINT "GREAT!"
2360 RETURN
2370 PRINT "FANTASTIC!!!"
2380 RETURN
2390 FOR I=1 TO 200
2400 NEXT I
2410 PRINT TAB(8);"A GREAT E
FFORT!"
2420 FOR I=1 TO 13
2430 PRINT
2440 NEXT I
2450 RETURN
```

This game can be frustrating, challenging, and entertaining all at the same time. The object is to arrange the numbers in ascending order (0 1 2 3 4 5 6 7 8 9) using a reversing technique. Lines 640-660 set up the original array table whereby L1(0)=0, L1(1)=1, . . . L1(9)=9. Lines 670-720 scramble the various integer locations. Lines 1060-1100 handle the actual reversing (which begins at the column you chose at 970). The following program is a shortened version of REVERSER.

```
 10 CALL CLEAR
 20 DIM N(4)
 30 FOR I=0 TO 4
 40 N(I)=I
 50 NEXT I
 60 PRINT
 70 FOR I=0 TO 4
 80 A=INT(RND*5)
 90 T=N(I)
100 N(I)=N(A)
110 N(A)=T
120 NEXT I
130 PRINT "INPUT A COLUMN (0-4)"
140 FOR I=0 TO 4
150 CALL HCHAR(10,10+2*I,48+I)
160 CALL HCHAR(11,10+2*I,48+N(I))
170 NEXT I
200 REM THE ACTUAL GAME
205 CALL KEY(0,RC,T)
210 IF T=0 THEN 205
215 RC=RC-48
220 IF (RC<0)+(RC>4) THEN 205
```

```
230 FOR I=RC TO RC+INT((4-RC)/2)
240 REM T=TEMPORARY
250 T=N(I)
260 N(I)=N(4+RC-I)
270 N(4+RC-I)=T
280 NEXT I
300 REM PRINT OUT THE NEW LIST
310 FOR I=0 TO 4
320 CALL HCHAR(11,10+2*I,48+N(I))
330 NEXT I
340 GOTO 205
```

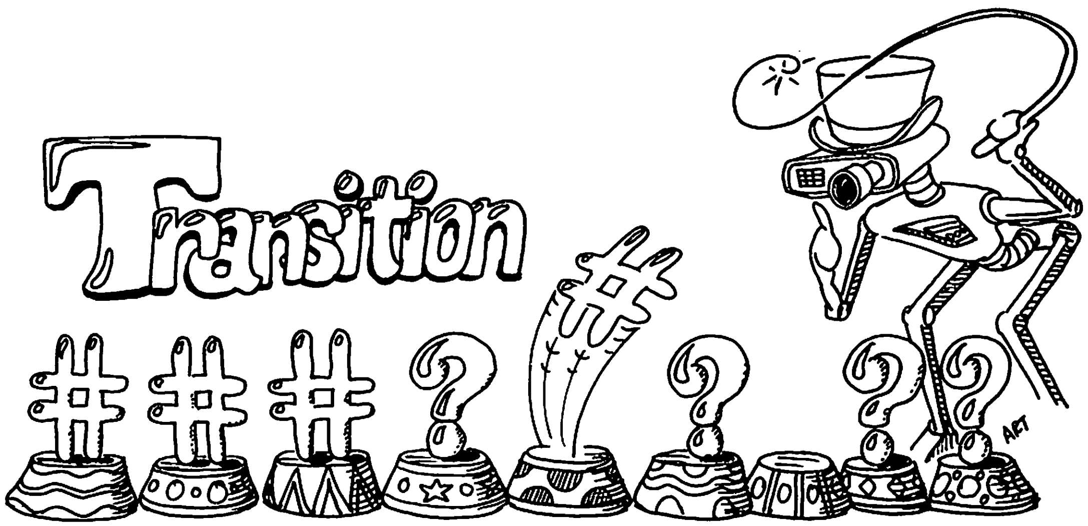
The above program has umpteen shortcomings. Copy and run this inferior version. It should help you to understand (and appreciate) the longer, more elaborate program.

```
10 REM ********************
20 REM ***              ***
30 REM ***   REVERSER   ***
40 REM ***              ***
50 REM ********************
60 GOSUB 110
70 GOSUB 580
80 GOSUB 860
90 GOSUB 1260
100 END
110 REM
120 REM ***INSTRUCTIONS***
130 REM
140 GOSUB 540
150 PRINT "IN THIS GAME YOU
ARE GIVEN  A LIST OF DIGITS
FROM 0-9.  THE LIST IS IN RA
NDOM ORDER,"
160 PRINT "YOUR TASK IS TO G
ET THE LISTINTO ASCENDING OR
DER BY     USING A REVERSING
 TECHNIQUE."
170 PRINT
180 PRINT "YOU ENTER THE NUM
BER OF THE COLUMN WHERE YOU
WANT THE   REVERSAL TO BEGIN
, AND THAT"
190 PRINT "COLUMN THROUGH CO
LUMN NINE  WILL BE REVERSED.
"
200 FOR I=1 TO 8
210 PRINT
```

```
220 NEXT I
230 GOSUB 520
240 GOSUB 540
250 PRINT "IF YOU HAD THE FO
LLOWING    SCRAMBLED LIST:"
260 PRINT
270 PRINT "     0 1 2 3 4 5
6 7 8 9"
280 PRINT
290 PRINT "     0 1 9 8 7 6
5 2 3 4"
300 PRINT
310 PRINT "AND YOU REVERSED
AT COLUMN  SEVEN, THEN THE N
EW LIST    WOULD LOOK LIKE T
HIS:"
320 PRINT
330 PRINT "     0 1 2 3 4 5
6 7 8 9"
340 PRINT
350 PRINT "     0 1 9 8 7 6
5 4 3 2"
360 FOR I=1 TO 6
370 PRINT
380 NEXT I
390 GOSUB 520
400 GOSUB 540
410 PRINT "A FINAL REVERSAL
AT COLUMN 2WOULD COMPLETE TH
E LIST."
420 PRINT
430 PRINT TAB(11);"COLUMNS"
440 PRINT "    0 1 2 3 4 5 6
 7 8 9"
450 PRINT
460 PRINT "    0 1 9 8 7 6 5
 4 3 2         ---------
------          0 1 2 3 4 5 6
 7 8 9"
470 PRINT
480 PRINT "SINCE THE LIST IS
 NOW IN    PROPER ASCENDING
ORDER, THE GAME IS WON.  GO
TO IT!"
490 FOR I=1 TO 7
500 PRINT
510 NEXT I
520 INPUT "PRESS ENTER WHEN
READY TO   CONTINUE: ":ANS$
530 RETURN
540 CALL CLEAR
```

```
550 PRINT TAB(7);"*** REVERS
ER ***"
560 PRINT
570 RETURN
580 REM
590 REM ***SETUP***
600 REM
610 GOSUB 540
620 RANDOMIZE
630 DIM LI(9)
640 FOR I=0 TO 9
650 LI(I)=I
660 NEXT I
670 FOR I=0 TO 9
680 X=INT(RND*10)
690 T=LI(I)
700 LI(I)=LI(X)
710 LI(X)=T
720 NEXT I
730 PRINT
740 PRINT TAB(11);"COLUMNS"
750 PRINT
760 PRINT "    0 1 2 3 4 5 6
 7 8 9"
770 FOR I=1 TO 12
780 PRINT
790 NEXT I
800 PRINT "PRESS THE NUMBER
OF THE     COLUMN WHERE YOU
WOULD LIKE TO REVERSE  (0-9)
."
810 PRINT
820 PRINT "      PRESS Q TO
QUIT"
830 C=0
840 Q=0
850 RETURN
860 REM
870 REM ***PLAY***
880 REM
890 FOR I=0 TO 9
900 CALL HCHAR(10,I*2+7,LI(I
)+48)
910 NEXT I
920 FOR I=0 TO 9
930 IF LI(I)<>I THEN 950
940 C=C+1
950 NEXT I
960 IF C=10 THEN 1160
970 CALL KEY(0,A,T)
980 IF T=0 THEN 970
```

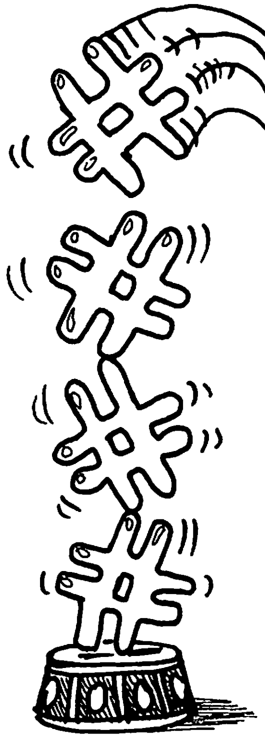
```
990 IF A=81 THEN 1140
1000 A=A-48
1010 IF A<0 THEN 970
1020 IF A>9 THEN 970
1030 FOR I=A TO 9
1040 CALL HCHAR(10,2*I+7,32)
1050 NEXT I
1060 FOR I=A TO A+INT((9-A)/
2)
1070 T=LI(I)
1080 LI(I)=LI(9+A-I)
1090 LI(9+A-I)=T
1100 NEXT I
1110 MOVE=MOVE+1
1120 C=0
1130 GOTO 890
1140 Q=1
1150 GOTO 1250
1160 FOR I=1 TO 200
1170 NEXT I
1180 CALL CLEAR
1190 PRINT "     CONGRATULAT
IONS!!!"
1200 FOR I=1 TO 12
1210 PRINT
1220 NEXT I
1230 FOR I=1 TO 200
1240 NEXT I
1250 RETURN
1260 REM
1270 REM ***END***
1280 REM
1290 CALL CLEAR
1300 IF Q=1 THEN 1370
1310 PRINT TAB(10);"YOU DID
IT!"
1320 PRINT
1330 PRINT
1340 PRINT "AND IT TOOK YOU"
;MOVE;"MOVES."
1350 FOR I=1 TO 300
1360 NEXT I
1370 RETURN
```



Impossible! It may seem impossible, but it's not. Deriving the key to this challenging game is very satisfying indeed. The object is to transpose this list: # # # # . ? ? ? ? so that it looks like this: ? ? ? ? . # # # #. Question marks (?) can only move to the left while pound signs (#) can only move to the right. Either sign may be moved during a turn, subject to the following specifications. A sign may only move directly into the space occupied by the period (referred to as the blank space), or jump over one opposing man and land in the blank space.

Look up the ASCII values for the three graphic symbols (#,?,.). Knowing which number represents which character will help you understand the program.

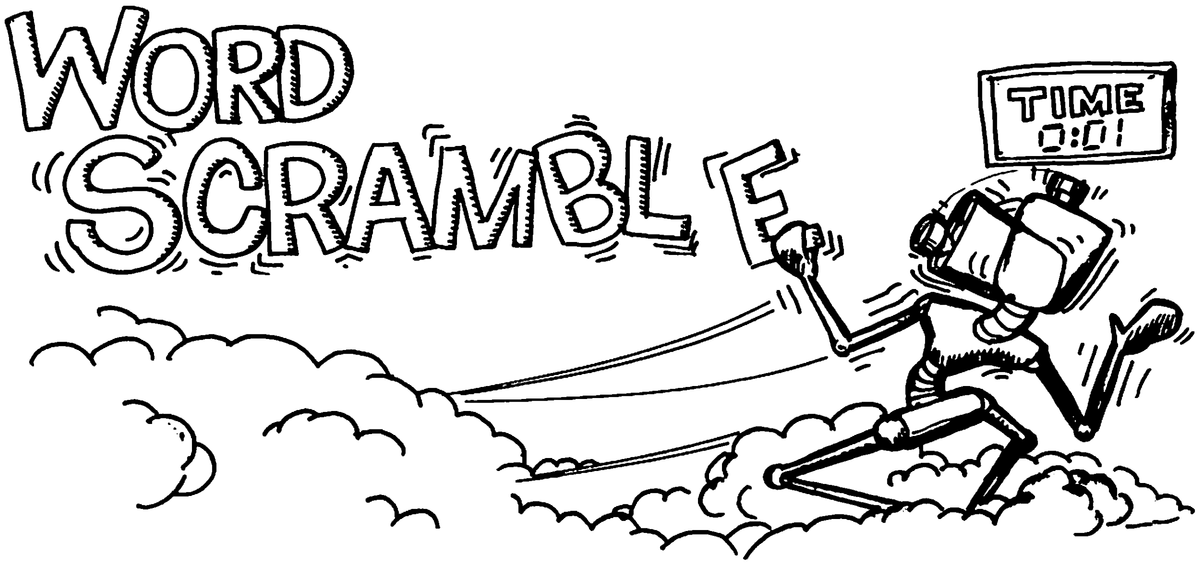
```
10 REM ********************
20 REM ***              ***
30 REM ***  TRANSITION  ***
40 REM ***              ***
50 REM ********************
60 GOSUB 110
70 GOSUB 440
80 GOSUB 650
90 GOSUB 1530
100 END
110 REM
120 REM ***INSTRUCTIONS***
130 REM
140 GOSUB 400
150 PRINT "IN THIS GAME YOU
WILL BE    GIVEN A SERIES OF
 NINE      CHARACTERS.  THE
LIST WILL"
160 PRINT "LOOK LIKE THIS:"
170 PRINT
180 PRINT "      1 2 3 4 5 6
 7 8 9"
190 PRINT "      # # # # . ?
 ? ? ?"
200 PRINT
210 PRINT "THE OBJECT IS TO
TRANSPOSE  THE ORIGINAL CHAR
ACTER      POSITIONS.  TRY T
O REVERSE"
220 PRINT "THE POUND SIGNS (
#) AND THE QUESTION MARKS IN
TO ONE     ANOTHER'S PREVIOU
S POSITION."
230 FOR I=1 TO 6
240 PRINT
250 NEXT I
260 GOSUB 380
270 GOSUB 400
280 PRINT "THE '#' CHARACTER
 CAN ONLY  MOVE TO THE RIGHT
.  THE     QUESTION MARKS CA
N ONLY MOVE"
290 PRINT "TO THE LEFT."
300 PRINT
310 PRINT "A MOVE IS MADE BY
 MOVING    INTO THE EMPTY SP
ACE OR BY  JUMPING ONE OPPOS
ING PIECE."
320 PRINT
330 PRINT "TO MAKE A MOVE, E
NTER THE   POSITION NUMBER O
F THE PIECETO BE MOVED.  TO
```



```
QUIT, ENTER"
340 PRINT "ZERO (0)."
350 FOR I=1 TO 7
360 PRINT
370 NEXT I
380 INPUT "PRESS ENTER WHEN
READY TO   CONTINUE: ":ANS$
390 RETURN
400 CALL CLEAR
410 PRINT TAB(7);"*** TRANSI
TION ***"
420 PRINT
430 RETURN
440 REM
450 REM ***SETUP***
460 REM
470 CALL CLEAR
480 DIM LI(10),M1(19),M2(26)
490 FOR I=1 TO 4
500 LI(I)=35
510 LI(10-I)=63
520 NEXT I
530 LI(5)=46
540 MM1$="THAT SPACE IS EMPT
Y"
550 FOR I=1 TO 19
560 M1(I)=ASC(SEG$(MM1$,I,1)
)
570 NEXT I
580 MM2$="IT CANNOT MOVE FUR
THER ..."
590 FOR I=1 TO 26
600 M2(I)=ASC(SEG$(MM2$,I,1)
)
610 NEXT I
620 NM=0
630 WL=1
640 RETURN
650 REM
660 REM ***PLAY***
670 REM
680 GOSUB 400
690 FOR I=1 TO 3
700 PRINT
710 NEXT I
720 PRINT "  ";
730 FOR I=1 TO 9
740 PRINT I;
750 NEXT I
760 FOR I=1 TO 14
770 PRINT
```

```
780 NEXT I
790 PRINT TAB(10);"MOVE (1-9
) :"
800 PRINT
810 PRINT TAB(8);"PRESS 0 TO
 STOP"
820 Y=5
830 FOR I=1 TO 9
840 CALL HCHAR(12,Y,LI(I))
850 Y=Y+3
860 NEXT I
870 C=0
880 FOR I=1 TO 4
890 IF LI(I)<>63 THEN 910
900 C=C+1
910 IF LI(10-I)<>35 THEN 930
920 C=C+1
930 NEXT I
940 IF LI(5)<>46 THEN 970
950 C=C+1
960 IF C=9 THEN 1420
970 CALL KEY(0,A,T)
980 IF T<>1 THEN 970
990 A=A-48
1000 IF A<0 THEN 970
1010 IF A>9 THEN 970
1020 IF A=0 THEN 1520
1030 IF LI(A)=46 THEN 1210
1040 IF LI(A)<>35 THEN 1070
1050 D=1
1060 IF LI(A+1)=35 THEN 1280
 ELSE 1100
1070 IF LI(A)<>63 THEN 1100
1080 D=-1
1090 IF LI(A-1)=63 THEN 1280
1100 IF LI(A+D)=46 THEN 1350
1110 IF A+2*D>9 THEN 1280
1120 IF A+2*D<0 THEN 1280
1130 IF LI(A+D)<>LI(A)THEN 1
150
1140 GOTO 1280
1150 IF LI(A+2*D)=46 THEN 11
70
1160 GOTO 1280
1170 LI(A+2*D)=LI(A)
1180 LI(A)=46
1190 NM=NM+1
1200 GOTO 820
1210 FOR I=1 TO 19
1220 CALL HCHAR(18,7+I,M1(I)
)
```

```
1230 NEXT I
1240 FOR I=1 TO 200
1250 NEXT I
1260 CALL HCHAR(18,8,32,19)
1270 GOTO 970
1280 FOR I=1 TO 26
1290 CALL HCHAR(18,4+I,M2(I))
1300 NEXT I
1310 FOR I=1 TO 200
1320 NEXT I
1330 CALL HCHAR(18,5,32,26)
1340 GOTO 820
1350 LI(A)=46
1360 NM=NM+1
1370 IF D=1 THEN 1400
1380 LI(A-1)=63
1390 GOTO 820
1400 LI(A+1)=35
1410 GOTO 820
1420 FOR I=1 TO 200
1430 NEXT I
1440 CALL CLEAR
1450 PRINT TAB(12);" WOW!!!"
1460 FOR I=1 TO 12
1470 PRINT
1480 NEXT I
1490 FOR I=1 TO 200
1500 NEXT I
1510 WL=99
1520 RETURN
1530 REM
1540 REM ***END***
1550 REM
1560 IF WL=99 THEN 1640
1570 CALL CLEAR
1580 PRINT "YOU GOT STUCK AF
TER";NM
1590 PRINT "MOVES.  BETTER L
UCK NEXT    TIME!"
1600 FOR I=1 TO 10
1610 PRINT
1620 NEXT I
1630 GOTO 1700
1640 PRINT TAB(9);"YOU DID IT!"
1650 PRINT
1660 PRINT "AND IT ONLY TOOK
";NM;"MOVES!"
1670 FOR I=1 TO 12
1680 PRINT
1690 NEXT I
1700 RETURN
```



For all you word buffs, here is a game of anagrams geared for any skill level. For those of you not familiar with the term, an anagram is a scrambled word. The object is to unscramble the word. The following list should help you to understand the rating system.

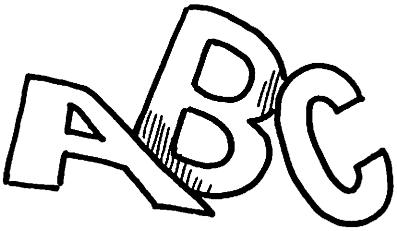
SCRAMBLED	UNSCRAMBLED	RATING
XOB	BOX	1 (EASY)
TAHB	BATH	2
MUHBT	THUMB	3 (LIGHT)
SLIPL	SPILL	4
NURDIG	DURING	5
TPLES	SLEPT	6 (TOUGH)
SPUMLIE	IMPULSE	7
TYRCLAS	CRYSTAL	8 (HARD)
MEMINSCON	MNEMONICS	9
PRESKULEN	SPELUNKER	10 (VERY HARD)

Lines 570-640 contain the data, so if you ever want to add or change words, just alter these lines.

```
10 REM ********************
20 REM ***              ***
30 REM *** WORD SCRAMBLE***
40 REM ***              ***
50 REM ********************
60 GOSUB 110
70 GOSUB 270
80 GOSUB 720
90 GOSUB 1660
100 END
110 REM
120 REM ***INSTRUCTIONS***
130 REM
140 GOSUB 230
150 PRINT "IN THIS GAME, THE
 COMPUTER  WILL CHOOSE A WOR
D AND SHOW YOU A SCRAMBLED V
ERSION."
160 PRINT
170 PRINT "YOUR GOAL IS TO U
NSCRAMBLE  THE WORD BEFORE T
HE ALLOTTEDTIME EXPIRES."
180 FOR I=1 TO 13
190 PRINT
200 NEXT I
210 INPUT "PRESS ENTER WHEN
READY TO   CONTINUE: ":ANS$
220 RETURN
230 CALL CLEAR
240 PRINT TAB(5);"*** WORD S
CRAMBLE ***"
250 PRINT
260 RETURN
270 REM
280 REM ***SETUP***
290 REM
300 RANDOMIZE
310 DIM TC(4),WC$(20),WO$(10
0),WR$(15),WS$(15)
320 GOSUB 230
330 FOR I=1 TO 4
340 PRINT
350 NEXT I
360 PRINT "THE FOLLOWING ARE
 AVAILABLE:"
370 PRINT
380 PRINT "   1) VERY EASY"
390 PRINT "   2) EASY"
400 PRINT "   3) INTERMEDIAT
E"
410 PRINT "   4) HARD"
```

```
420 PRINT "   5) VERY HARD"
430 FOR I=1 TO 8
440 PRINT
450 NEXT I
460 INPUT "WHICH OPTION (1-5
) ? ":ANS
470 IF ANS<1 THEN 460
480 IF ANS>5 THEN 460
490 IF ANS<>INT(ANS)THEN 460
500 FOR I=1 TO 20
510 WC$(I)=""
520 NEXT I
530 FOR I=1 TO 100
540 READ WO$(I)
550 NEXT I
560 RESTORE
570 DATA DOT,CAT,TREE,GIN,LI
P,GOOD,BOX,MAN,WALL,PEN,PAD,
CUP,DIG,TIE,SEA,EYE,SING,BET
,FAR,MOP
580 DATA WIND,WARM,LEAF,GREE
N,JOKE,SLIP,DRIP,MAZE,PARK,L
INE,GAME,HIGH,RISK,RAIN,CARD
,VERY,MOLE,ITCH,DATA,VERY
590 DATA PAPER,PHONE,LOOSE,T
IRED,PROOF,BLACK,AFTER,KNOCK
,BAGEL,RETURN,ENTRY,INPUT,ST
ART,LISTEN,HORSE,GROUND
600 DATA SHINE,FIGURE,CONVOY
,CRAZY
610 DATA ORIGINAL,FASTEN,PER
FORM,PENCIL,SOFTWARE,NORMAL,
STORAGE,WINDOW,SYSTEM,GAMBIT
,REALIZE,COMBINE,TANGENT
620 DATA EVALUATE,SPECIFY,ME
MORY,IGNORE,BEFORE,HOWEVER,I
NSIDE
630 DATA TRAJECTORY,DENY,TEC
HNIQUE,EMULATE,CONICAL,APPRO
PRIATE,IRIDESCENT,SEQUOIA,MA
TRIX,CIRCUMVENT,STIMULUS
640 DATA SUBSTANTIAL,MAGNETI
C,VITAMINS,PROJECTION,PROBAB
ILITY,CONTINUE,SUBSCRIPT,ENV
ELOPED,COORDINATE
650 CALL COLOR(9,15,15)
660 WC=0
670 TC1$="TIME"
680 FOR I=1 TO 4
690 TC(I)=ASC(SEG$(TC1$,I,1))
700 NEXT I
```

```
710 RETURN
720 REM
730 REM ***PLAY***
740 REM
750 WO=INT(RND*20)+(ANS-1)*2
0+1
760 WL=LEN(WO$(WO))
770 FOR I=1 TO WL
780 WR$(I)=SEG$(WO$(WO),I,1)
790 WS$(I)=WR$(I)
800 NEXT I
810 FOR I=1 TO WL
820 TI=INT(RND*WL+1)
830 SW$=WS$(I)
840 WS$(I)=WS$(TI)
850 WS$(TI)=SW$
860 NEXT I
870 LC=0
880 FOR I=1 TO WL
890 IF WS$(I)=WR$(I)THEN 158
0
900 NEXT I
910 IF LC=WL THEN 810
920 GOSUB 230
930 FOR I=1 TO 15
940 PRINT
950 NEXT I
960 PRINT "THIS IS THE SCRAM
BLED WORD:"
970 PRINT
980 PRINT "STARTING AT THE L
EFT, ENTER A LETTER (A-Z)."
990 PRINT
1000 PRINT "PRESS ENTER WHEN
 FINISHED."
1010 FOR A=1 TO 4
1020 CALL HCHAR(8,A+26,TC(A))
1030 NEXT A
1040 FOR I=1 TO WL
1050 CALL HCHAR(9,I+13,ASC(W
S$(I)))
1060 NEXT I
1070 CALL HCHAR(7,11,97,WL+6)
1080 CALL HCHAR(11,11,97,WL+
6)
1090 CALL VCHAR(8,11,97,3)
1100 CALL VCHAR(8,WL+16,97,3)
1110 FOR I=14 TO 13+WL
1120 CALL HCHAR(15,I,95)
1130 NEXT I
1140 WP=1
```



```
1150 FOR I=100*ANS TO 1 STEP
 -1
1160 IF I/20=INT(I/20)THEN 1
320
1170 CALL KEY(5,L,T)
1180 IF T<>1 THEN 1290
1190 IF L=8 THEN 1480
1200 IF L=9 THEN 1530
1210 IF L=13 THEN 1600
1220 IF L<65 THEN 1170
1230 IF L>90 THEN 1170
1240 CALL HCHAR(15,13+WP,L)
1250 WC$(WP)=CHR$(L)
1260 WP=WP+1
1270 FOR DL=1 TO 25
1280 NEXT DL
1290 NEXT I
1300 CALL HCHAR(10,28,48,2)
1310 GOTO 1600
1320 IF (I/20)>9 THEN 1360
1330 CALL HCHAR(10,28,48)
1340 CALL HCHAR(10,29,I/20+4
8)
1350 GOTO 1170
1360 A=INT(I/200)
1370 A1=A
1380 FOR AA=1 TO A
1390 A=AA+48
1400 NEXT AA
1410 B=(I/20)-(A1*10)
1420 FOR BB=0 TO B
1430 B=BB+48
1440 NEXT BB
1450 CALL HCHAR(10,28,A)
1460 CALL HCHAR(10,29,B)
1470 GOTO 1170
1480 WP=WP-1
1490 IF WP<1 THEN 1510
1500 GOTO 1290
1510 WP=1
1520 GOTO 1290
1530 WP=WP+1
1540 IF WP>WL THEN 1560
1550 GOTO 1290
1560 WP=WL
1570 GOTO 1290
1580 LC=LC+1
1590 GOTO 900
1600 FOR I=1 TO WL
1610 CALL HCHAR(9,13+I,ASC(W
R$(I)))
```

```
1620 NEXT I
1630 FOR DELAY=1 TO 700
1640 NEXT DELAY
1650 RETURN
1660 REM
1670 REM ***END***
1680 REM
1690 CALL CLEAR
1700 FOR I=1 TO WL
1710 IF WC$(I)<>WR$(I)THEN 1
730
1720 WC=WC+1
1730 NEXT I
1740 IF WC=WL THEN 1800
1750 PRINT "OUT OF";WL;"YOU
GOT";WC;"CORRECT"
1760 FOR I=1 TO 12
1770 PRINT
1780 NEXT I
1790 GOTO 1830
1800 FOR I=1 TO 18
1810 CALL HCHAR(12,I+7,ASC(S
EG$("CONGRATULATIONS!!!",I,1
)))
1820 NEXT I
1830 INPUT "DO YOU WISH TO P
LAY AGAIN?":ANS$
1840 IF SEG$(ANS$,1,1)="Y" T
HEN 70
1850 RETURN
```

Beware! This game is addicting! Within the nine by six arena are 40 food points. The object is to consume all 40 squares before the Mubble Eaters consume you. If on the first run you do not complete the mission, you still have two more Mubbles in reserve. It is up to you to safely guide the Mubble to gluttony. Go to it!

This program is explained in depth. It was chosen because the use of graphics is such that most of the other programs can be understood if this program is understood.

10 REM is short for REMark. Anything may be construed as a remark. In this case, the REM statement is used to allow the program name to be written. Naturally, ****************** is not really a remark, but in this case the REM statement allows the programmer to write the title in computer-acceptable form.

20-50 These REMarks merely finish the program title.

60 GOSUB 110 tells the computer to branch to line 110 (referred to as the target line) and to continue from there until the RETURN statement is encountered. When it is, the program will RETURN to line 60, and, having completed 60, drop down to the next line.

70-90 These lines are the same format as line 60. This uniform coding is the key to structured programming. In each program you will find the main program loop followed by the instructions, the setup, the game play, and the game end.

100 When line 90 has been fulfilled, the program will drop down to 100, and terminate. The END statement may appear anywhere in the program (it may even appear more than once). Just be sure that it isn't performed before the proper time.

120 This line informs the reader that the INSTRUCTIONS are to follow.

140 Jumps temporarily to perform program lines 380-410, then returns to continue program execution with line 150. The effect of this routine is to clear the screen (line 380) and print the game title (390) and a blank line (400).

150 The PRINT statement is perhaps the most versatile of all computer instructions. There are four uses which will be detailed. One, the PRINT statement will display mathematical results. Two, anything enclosed by quotation marks will be reproduced verbatim. Three, PRINT operates as a carriage return. Four, a TAB command may follow PRINT. Copy and run the following program.

```
 10 CALL CLEAR
 20 REM MATH RESULTS
 30 PRINT 3+4;6*8;3-1
 40 REM A MESSAGE IN QUOTES
 50 PRINT "THIS IS AN EXAMPLE."
 60 PRINT "3+4";"6*8";"3-1"
 70 REM PRINT ALWAYS ACTS AS A CARRIAGE RETURN
 80 PRINT
 90 REM TAB FOLLOWING PRINT
100 PRINT TAB(7);"HELLO"
110 PRINT "HELLO"
```

RUN

```
7 48 2
THIS IS AN EXAMPLE.
3+46*83-1

      HELLO
HELLO
```

Now to explain. The first line (7 48 2) is the result of three mathematical functions. The semicolon (;) serves two purposes: It separates each equation and it tells the computer to PRINT beginning in the next available print space. So why are there spaces between the three answers? The TI allows a space for a sign (+ or –), whether the sign is present or not. Moving down, the PRINT statement at line 50 acts as a carriage return, thereby skipping down to the next line. The message in quotes is then printed. Line 60 contains three equations, but because they are in quotes, the computer does not perform the math and it does not allow for a plus (+) or minus (–) sign.

Line 80 is referred to as an empty PRINT statement. Its only function is to skip to the beginning of the next line. The purpose of 80 is not seen until line 100 is performed. Remember, PRINT always acts as a carriage return. Therefore, line 100's PRINT skips to the beginning of a new line, even though the previous line was blank (or empty). TAB(7) instructs the computer to skip to the seventh PRINT position, then the message "HELLO" is PRINTed. Line 110 serves to show how TAB functions by comparing it to line 100.

210-230 The concept of looping is intended to save time and paper. There are different methods, but the FOR/NEXT loop is one of the best. A FOR statement accomplishes two things. First, the variable (or counter) is specified. Second, the range of the counter is specified (1 to 7). The following three programs demonstrate the importance of looping.

```
10 REM X IS THE COUNTER
20 FOR X=1 TO 10
30 PRINT X
40 NEXT X
50 END
```

```
RUN

 1
 2
 3
 4
 5
 6
 7
 8
 9
 10
```

```
10 REM AGAIN, X IS THE COUNTER
20 X=1
30 PRINT X
40 IF X=10 THEN 70
50 X=X+1
60 GOTO 30
70 END
```

RUN

```
 1
 2
 3
 4
 5
 6
 7
 8
 9
10
```

```
 10 PRINT  1
 20 PRINT  2
 30 PRINT  3
 40 PRINT  4
 50 PRINT  5
 60 PRINT  6
 70 PRINT  7
 80 PRINT  8
 90 PRINT  9
100 PRINT 10
```

RUN

```
 1
 2
 3
 4
 5
 6
 7
 8
 9
10
```

These three programs all perform the same task, but they're not the same. The first program is the most efficient. Following is an explanation of the computer flow (of the first program).

10 The computer ignores the REMark.
20 A loop has begun. The counter is identified as X with X=1 (FOR X=1.......).
30 The value of X is printed.
40 The NEXT statement increments X and returns the program to line 20.
20 The computer checks to see if X is still between 1 and 10 (X=2).
30 The value of X is printed.
40 X is incremented. The program branches to 20.
20 The computer verifies that X is between 1 and 10 (X=3).
30 X is printed.
40 X is incremented. Flow is sent to line 20.
20 ....

Finally, when X equals 10

40 X is incremented. The program is sent to line 20.
20 The computer checks to see if X is between 1 and 10. Since it's not (X=11), the program skips past the NEXT command to the following instruction.

50 END program execution.

Getting back to 210-230, the loop consists of seven passes (I=1,2,3,4,5,6,7). Line 220 operates as a carriage return (seven times). The net affect is to "skip" (or PRINT) seven empty lines.

240 Jumps temporarily to perform program lines 360-370, then returns to continue program execution with line 250. This routine prints the message in quotes (line 360) and waits for the user to press ENTER before continuing.

370 As stated in line 60, the SUBroutine which began at 110 would be completed by RETURN, which returns program flow to 60. From there, the program skips to the next instruction at line 70.

450 During program execution, screen color is medium green (color 3). The CALL SCREEN command can change the screen color to any of the sixteen colors. Copy and run the following program:

```
10 CALL CLEAR
20 FOR I=1 TO 16
30 CALL SCREEN(I)
40 FOR DELAY=1 TO 100
50 NEXT DELAY
60 NEXT I
```

470 TI BASIC has a built-in random number generator. But, like most of the others, it will generate the same random number (or sequence) over and over again — unless told otherwise. The RANDOMIZE statement ensures constant random generation. The command for a random number, RND, yields an eight digit number between zero and one — .00000000-.99999999. How to convert this unwieldy number into integer form will be discussed later. To better understand RaNDom number generation, copy and run the following program. Run it five times.

```
10 CALL CLEAR
30 FOR I=1 TO 4
40 PRINT RND
50 NEXT I
```

You will note that on each run you get the same set of four random numbers. Add the following line to the program — 20 RANDOMIZE. Run the new program five times.

480 A computer typically reserves one memory location per variable. That is, if a program reads: 10 X=5 20 X=2, then as far as the computer is concerned, X equals 2. What if you want X to equal 5 and 2? The solution is to DIMension memory to allow for two separate values for X. This is done with a DIM statement. The following programs will clarify the situation.

```
10 DIM X(2)          10 REM NO DIM
20 X(1)=5            20 X=5
30 X(2)=2            30 X=2
40 PRINT X(1)        40 PRINT X
50 PRINT X(2)        50 PRINT X
```

RUN

```
5                    2
2                    2
```

490 A CALL COLOR statement contains three keys in parentheses. The general purpose of this command is to assign a color to a certain Character Set. The first number identifies the set — CALL COLOR(CHARACTER SET,2,2). The second number specifies the character set's foreground color (color 2 is black) — CALL COLOR(CHARACTER SET, FOREGROUND COLOR, 2). The third number specifies the Character Set's background color — CALL COLOR(CHARACTER SET, FOREGROUND COLOR, BACKGROUND COLOR). To paraphrase: Draw the eight characters in Character Set #2 in black with a black background.

520-570 Here is an example of a nested loop. The term "nested" means "contained within." Each time the main (outside) loop performs a single pass, the inside loop is performed from start to finish. To help you understand, copy and run the following program.

```
10 CALL CLEAR
20 FOR X=1 TO 5
30 FOR Y=1 TO 5
40 PRINT "X =";X
50 PRINT "Y =";Y
60 NEXT Y
70 NEXT X
80 PRINT "ALL DONE!"
```

530 The CALL HCHAR (or VCHAR) statement is similar to a PRINT statement. The numbers in parentheses explain what to print, where to print it, and how many times to repeat the process. The first number specifies the row. The second number specifies the column. The third number is the ASCII value of the character to be PRINTed. The fourth number (which is often omitted) specifies the number of repetitions. A simple way to remember is: CALL HCHAR(ROW, COLUMN, ASCII VALUE, REPETITIONS). There is one major difference between CALL HCHAR (Horizontal CHARacter) and CALL VCHAR (Vertical CHARacter). The difference has to do with repetitions. Here's how. Line 530 reads: CALL HCHAR(I+1,2, 40,30). The printer moves to row I+1 (I=1 TO 21), column 2, prints the character with ASCII value 40, and moves horizontally right while repeating the PRINTing 30 times. With I=1, the repetitions will occur at Row,Column: 2,2; 2,3; 2,4; 2,5; 2,6; etc. If the instruction had specified VCHAR, then the repetitions would have been at 2,2; 3,2; 4,2; 5,2; 6,2; etc.

If you look for the character with ASCII value 40, you will find that ASCII 40 corresponds to a left parenthesis. So why don't you see any left parentheses on the screen? Line 490 set all Character Set CHNO2 characters (ASCII values 40 through 47) to PRINT in black — on a black background! The effect is a solid black square. That's why.

580 This line merits discussion for one reason — STEP 3. In a previous explanation you were told that NEXT increments the counter. Unless specified, the incrementation is by 1 (1 is added to the counter). If the counter is to be incremented at a rate other than 1, the STEP option is used. You may add any number to the counter: positive, negative, integer or floating point.

690 This line elaborates on the typical RND function. In English, 690 reads: R1 equals the INTeger value of a RaNDom number (between 0 and 1) multiplied by 6, with that integer added to 9. Read the chart below.

NUMBER	INTEGER EQUIVALENT
3.4	3
3.9	3
563.772	563
.8	0
-3.4	-4
-3.002	-4
-57.23	-58

As you can see, with positive numbers, everything to the right of the decimal point is truncated (cut off). With negative numbers, the number is changed to the smallest integer equivalent.

960 Wow! This line is really messed up! It may look that way, but everything is really fine. Looking ahead, line 1250 sets D1 equal to D. Line 880 sets D1 to 68. ASCII 68 corresponds to D (->). This conglomeration of lines checks to see if an impossible request was made of the computer. The four impossible requests are: D1=68 (which means: move right) and (*) the Mubble's Y coordinate is the rightmost (30), D1=69 (E or up arrow) and the Mubble's X coordinate is the top border, D1=83 (S or left arrow) and the Mubble's Y coordinate (MY) is equal to the left border, D1=88 (X or down arrow) and MX equals the bottom border. These are the four ways of trying to move off the gameboard; and they are all forbidden.

970 This line will cause program flow to skip to 1010 unless D equals 68.

980 Since D=68 corresponds to a right arrow, the X coordinate incrementer (which controls the up and down movements) is set to zero.

990 When moving to the right, the Y coordinate is incremented.

1120-1170 These lines adjust the coordinates of the three piece Mubble.

1180-1200 These lines draw the Mubble in its new position.

1210 The CALL KEY statement has a very specific function. The items in parentheses control the action. The first item, 0, is a code which instructs the computer on the type of input to accept. The CALL KEY statement is designed to read one keyboard character. Whether to accept UPPER CASE, lower case, both, numbers, etc. is determined by the code. The second item in parentheses, D, is the variable name into which a keyboard character's ASCII code will be placed. The third item, T, is a variable name into which a special code number is placed. T can contain -1, 0 or 1. If T equals -1, then the computer has read an ASCII value into D, but it is the value left over from the previous pass. If T equals zero (0), then no value has been read into D. If T equals one (1), then a new value has been moved into D.

1280 The Texas Instruments computer has an extensive musical scope. The CALL SOUND instruction is the programmer's way of accessing it. The three numbers in parentheses are Duration, Pitch, and Volume. The Duration ranges from 1/1000 of a second to 4250/1000 (or 4 1/4) seconds. The Pitch is calculated according to a specific frequency scale — the Hertz scale. The scale ranges from 110 (the lowest, deepest note) to 44733 (a note far too high for humans to hear). The Volume ranges from 0 (the loudest) to 30 (virtually inaudible). A CALL SOUND statement may play many different notes, but they all reflect the same Duration. For example: CALL SOUND(Duration, Pitch, Volume, Pitch 2, Volume 2, Pitch 3, Volume 3). What this means is, the second pitch and the second volume (Pitch 2 and Volume 2) and the third pitch and the third volume (Pitch 3 and Volume 3) will have the same Duration as the first note.

The rest of the program employs instructions that have already been explained.

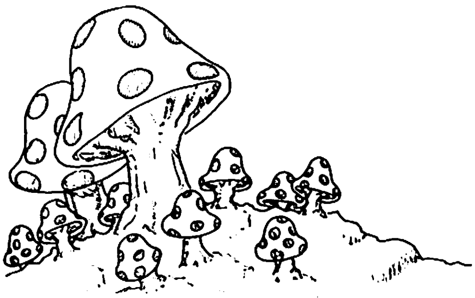
```
10 REM ********************
20 REM ***              ***
30 REM *** MUBBLE CHASE ***
40 REM ***              ***
50 REM ********************
60 GOSUB 110
70 GOSUB 420
80 GOSUB 920
90 GOSUB 2010
100 END
110 REM
120 REM ***INSTRUCTIONS***
130 REM
140 GOSUB 380
150 PRINT "IN THIS EXCITING
GAME YOU   CONTROL THE MOVEM
ENT OF THE HUNGRY LITTLE CRE
ATURE WE"
160 PRINT "CALL THE MUBBLE."
170 PRINT
180 PRINT "THE MUBBLE SCURRI
ES THROUGH A MAZE, TRYING TO
 EAT UP ALLOF THE FOOD POINT
S."
190 PRINT
200 PRINT "UNFORTUNATELY, TH
ERE ARE    THREE MUBBLE EATE
RS WHO WANTTO CATCH AND EAT
THE POOR   MUBBLE."
210 FOR I=1 TO 7
220 PRINT
230 NEXT I
240 GOSUB 360
250 GOSUB 380
260 PRINT "YOU MUST MANEUVER
 THE MUBBLETO THE FOOD POINT
S AND AWAY FROM THE MUBBLE E
ATERS."
270 PRINT
280 PRINT "YOU ARE ALLOWED T
O LOSE TWO MUBBLES, BUT WHEN
 THE THIRD MUBBLE IS EATEN,
THE GAME ISOVER."
290 PRINT
300 PRINT "MOVEMENT OF THE M
UBBLE IS   CONTROLLED BY THE
 FOUR ARROWKEYS."
310 PRINT
320 PRINT "              E
                        S D
                         X"
330 FOR I=1 TO 4
```

```
340 PRINT
350 NEXT I
360 INPUT "PRESS ENTER WHEN
READY TO   CONTINUE: ":ANS$
370 RETURN
380 CALL CLEAR
390 PRINT TAB(5);"*** MUBBLE
 CHASE ***"
400 PRINT
410 RETURN
420 REM
430 REM ***SETUP***
440 REM
450 CALL SCREEN(15)
460 CALL CLEAR
470 RANDOMIZE
480 DIM GB(21,30),MX(3),MY(3
),EX(3),EY(3)
490 CALL COLOR(2,2,2)
500 CALL COLOR(5,5,5)
510 CALL COLOR(7,7,7)
520 FOR I=1 TO 21
530 CALL HCHAR(I+1,2,40,30)
540 FOR J=1 TO 30
550 GB(I,J)=15
560 NEXT J
570 NEXT I
580 FOR I=3 TO 21 STEP 3
590 CALL HCHAR(I,4,32,26)
600 NEXT I
610 FOR I=3 TO 30 STEP 3
620 CALL VCHAR(3,I,32,19)
630 NEXT I
640 FOR I=9 TO 15
650 CALL COLOR(I,I,I)
660 NEXT I
670 FOR R=6 TO 18 STEP 3
680 FOR J=6 TO 27 STEP 3
690 R1=INT(RND*6)+9
700 GB(R,J)=R1
710 CALL HCHAR(R,J,R1*8+30)
720 NEXT J
730 NEXT R
740 FOR R=1 TO 3
750 MX(R)=21
760 CALL HCHAR(21,2+R,70)
770 MY(R)=2+R
780 NEXT R
790 EX(1)=21
800 EY(1)=30
810 EX(2)=3
```

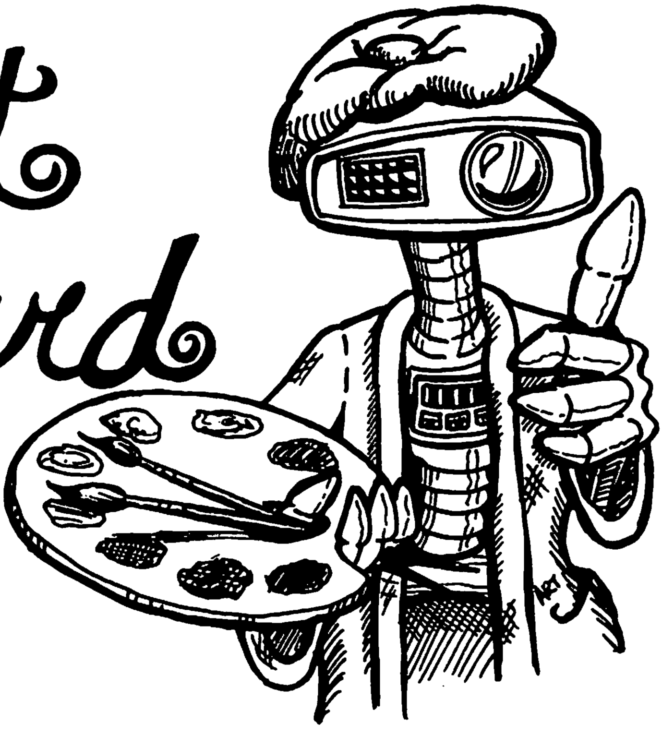
```
820 EY(2)=30
830 EX(3)=3
840 EY(3)=3
850 FOR R=1 TO 3
860 CALL HCHAR(EX(R),EY(R),8
4)
870 NEXT R
880 D1=68
890 YW=0
900 EM=0
910 RETURN
920 REM
930 REM ***PLAY***
940 REM
950 Q=0
960 IF ((D1=68)*(MY(3)=30))+
((D1=69)*(MX(3)=3))+((D1=83)
*(MY(3)=3))+((D1=88)*(MX(3)=
21))THEN 1210
970 IF D1>68 THEN 1010
980 X=0
990 Y=1
1000 GOTO 1110
1010 IF D1>69 THEN 1050
1020 X=-1
1030 Y=0
1040 GOTO 1110
1050 IF D1>83 THEN 1090
1060 X=0
1070 Y=-1
1080 GOTO 1110
1090 X=1
1100 Y=0
1110 CALL HCHAR(MX(1),MY(1),
150)
1120 MX(1)=MX(2)
1130 MY(1)=MY(2)
1140 MX(2)=MX(3)
1150 MY(2)=MY(3)
1160 MX(3)=MX(3)+X
1170 MY(3)=MY(3)+Y
1180 CALL HCHAR(MX(1),MY(1),
70)
1190 CALL HCHAR(MX(2),MY(2),
70)
1200 CALL HCHAR(MX(3),MY(3),
70)
1210 CALL KEY(0,D,T)
1220 IF ((D=69)*(MY(3)/3=INT
(MY(3)/3))*(MX(3)<>3))+((D=6
8)*(MX(3)/3=INT(MX(3)/3))*(M
Y(3)<>30))THEN 1250
1230 IF ((D=83)*(MX(3)/3=INT
(MX(3)/3))*(MY(3)<>3))+((D=8
8)*(MY(3)/3=INT(MY(3)/3))*(M
X(3)<>21))THEN 1250
1240 GOTO 1260
1250 D1=D
1260 IF GB(MX(3),MY(3))=15 T
HEN 1310
1270 GB(MX(3),MY(3))=15
1280 CALL SOUND(200,262,2,33
0,2)
1290 SC=SC+1
1300 IF SC>39 THEN 1970
1310 FOR I=1 TO 3
1320 CALL HCHAR(EX(I),EY(I),
GB(EX(I),EY(I))*8+30)
1330 R=INT(RND*7+1)
1340 R4=INT(RND*4+1)
1350 ON R GOSUB 1530,1530,15
30,1530,1780,1780,1780
1360 CALL HCHAR(EX(I),EY(I),
86)
1370 FOR Z=1 TO 3
1380 IF (EX(I)=MX(Z))*(EY(I)
=MY(Z))THEN 1390 ELSE 1420
1390 ME=ME+1
1400 IF ME>2 THEN 2000
1410 Q=1
1420 NEXT Z
1430 IF Q<>1 THEN 1510
1440 CALL SOUND(500,140,2,12
8,2)
1450 FOR P=1 TO 3
1460 CALL HCHAR(MX(P),MY(P),
150)
1470 CALL HCHAR(EX(P),EY(P),
GB(EX(P),EY(P)))
1480 NEXT P
1490 GOSUB 740
1500 GOTO 920
1510 NEXT I
1520 GOTO 920
1530 IF (EX(I)/3=INT(EX(I)/3
))*(EY(I)/3=INT(EY(I)/3))THE
N 1570
1540 IF EY(I)/3=INT(EY(I)/3)
THEN 1560
1550 ON R4 GOTO 1580,1580,16
30,1630
1560 ON R4 GOTO 1680,1680,17
30,1730
```

```
1570 ON R4 GOTO 1580,1630,16
80,1730
1580 IF (EX(I)<>MX(3))*(EY(I
)=MY(3))*(EY(I)/3=INT(EY(I)/
3))THEN 1560
1590 IF EY(I)=3 THEN 1660
1600 IF EY(I)<MY(3)THEN 1660
1610 EY(I)=EY(I)-1
1620 GOTO 1770
1630 IF (EX(I)<>MX(3))*(EY(I
)=MY(3))*(EY(I)/3=INT(EY(I)/
3))THEN 1560
1640 IF EY(I)=30 THEN 1610
1650 IF EY(I)>MY(3)THEN 1610
1660 EY(I)=EY(I)+1
1670 GOTO 1770
1680 IF (EY(I)<>MY(3))*(EX(I
)=MX(3))*(EX(I)/3=INT(EX(I)/
3))THEN 1550
1690 IF EX(I)=3 THEN 1760
1700 IF EX(I)<MX(3)THEN 1760
1710 EX(I)=EX(I)-1
1720 GOTO 1770
1730 IF (EY(I)<>MY(3))*(EX(I
)=MX(3))*(EX(I)/3=INT(EX(I)/
3))THEN 1550
1740 IF EX(I)=21 THEN 1710
1750 IF EX(I)>MX(3)THEN 1710
1760 EX(I)=EX(I)+1
1770 RETURN
1780 IF (EX(I)/3=INT(EX(I)/3
))*(EY(I)/3=INT(EY(I)/3))THE
N 1840
1790 IF EX(I)/3=INT(EX(I)/3)
THEN 1830
1800 IF EY(I)/3=INT(EY(I)/3)
THEN 1820
1810 RETURN
1820 ON R4 GOTO 1850,1850,18
80,1880
1830 ON R4 GOTO 1910,1910,19
40,1940
1840 ON R4 GOTO 1850,1880,19
10,1940
1850 IF EX(I)=3 THEN 1880
1860 EX(I)=EX(I)-1
1870 GOTO 1960
1880 IF EX(I)=21 THEN 1850
1890 EX(I)=EX(I)+1
1900 GOTO 1960
1910 IF EY(I)=3 THEN 1940
```

```
1920 EY(I)=EY(I)-1
1930 GOTO 1960
1940 IF EY(I)=30 THEN 1910
1950 EY(I)=EY(I)+1
1960 RETURN
1970 FOR I=1 TO 500
1980 NEXT I
1990 YW=1
2000 RETURN
2010 REM
2020 REM ***END***
2030 REM
2040 CALL CLEAR
2050 FOR I=2 TO 9
2060 CALL COLOR(I,2,15)
2070 NEXT I
2080 IF YW=0 THEN 2180
2090 PRINT "     CONGRATULAT
IONS!!!"
2100 FOR I=1 TO 4
2110 PRINT
2120 NEXT I
2130 PRINT "YOU'VE DONE AN E
XCELLENT    JOB.  THE MUBBLE
 IS SAFE ANDNO LONGER HUNGRY
."
2140 FOR I=1 TO 4
2150 PRINT
2160 NEXT I
2170 GOTO 2220
2180 PRINT "WELL, I GUESS TH
E MUBBLE    EATERS WON THIS
TIME.  MAYBENEXT TIME THEY W
ON'T BE SO  LUCKY!"
2190 FOR I=1 TO 10
2200 PRINT
2210 NEXT I
2220 RETURN
```



Artist Board



Grab an easel, a palette and a canvas, or sit down to your computer. Either way, colorful fun will be the result.

As with any game, analysis is more meaningful after you run the program.

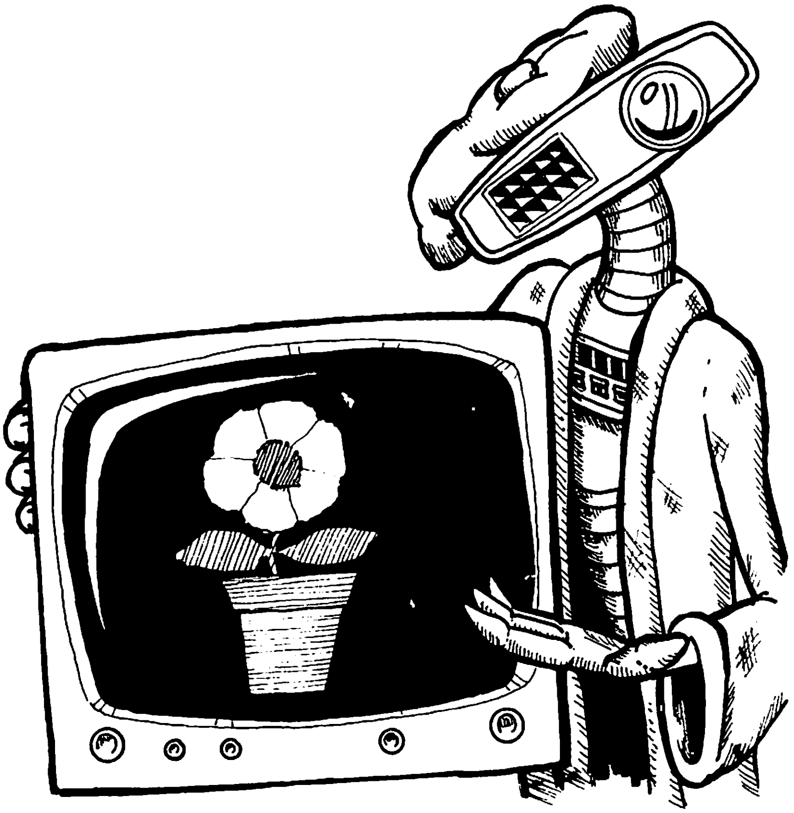
List 690-790. Here the program checks to see if any relevant keyboard character was pressed. A CALL KEY statement contains three items in parentheses. The following description should help you see the relationship between 690 and 710-780. The instruction, CALL KEY(A,B,C), is very meaningful. "A" is a digit-code which tells the computer the type of input to accept; lower case, UPPER CASE, both, numeric only, etc. "B" stores the ASCII value of the first character key pressed. Check your manual to find Upper Case ASCII values (A+65, B=66, C=67, . . . Y=89, Z-90). The final variable, "C", assumes a value of -1, 0 or 1. Minus one (-1) indicates that the key being pressed is the same key as was entered into B the previous time the line was performed. Zero (0) indicates that no key is being pressed. One (1) indicates that a new value has been entered into B. As you can see, lines 710-780 check the ASCII value, which is stored in N.

```
10 REM ********************
20 REM ***              ***
30 REM *** ARTIST BOARD ***
40 REM ***              ***
50 REM ********************
60 GOSUB 100
70 GOSUB 490
80 GOSUB 650
90 END
100 REM
110 REM ***INSTRUCTIONS***
120 REM
130 GOSUB 450
140 PRINT "BY USING ARTIST B
OARD AND   YOUR CREATIVE TAL
ENT, YOU   CAN CREATE PICTUR
ES IN LOW-"
150 PRINT "RESOLUTION GRAPHI
CS."
160 PRINT
170 PRINT "THE 4 ARROW KEYS
DETERMINE  WHERE THE COLOR(S
) WILL     OCCUR.  YOU HAVE
THE TOOLS"
180 PRINT "TO CREATE SOME VE
RY NICE    DRAWINGS."
190 PRINT
200 PRINT "CHECK YOUR MANUAL
 TO SEE    WHICH COLOR IS RE
PRESENTED  BY WHICH NUMBER (
1-16)."
210 FOR I=1 TO 6
220 PRINT
230 NEXT I
240 GOSUB 430
250 GOSUB 450
260 PRINT "FOR PROGRAMMING R
EASONS,    ONLY THE FOLLOWIN
G COLORS   WILL BE AVAILABLE
; NOT 1 0R 2, BUT 3-16."
270 PRINT
280 PRINT "TO CHANGE COLORS,
 TYPE C    FOLLOWED BY THE N
UMBER OF   THE NEW COLOR."
290 FOR I=1 TO 12
300 PRINT
310 NEXT I
320 GOSUB 430
330 GOSUB 450
340 PRINT "TYPE M TO MOVE, Q
 TO QUIT,  AND Z TO ERASE."
```

```
350 PRINT
360 PRINT "IF YOU PRESS ONE
OF THE FOURARROW KEYS BEFORE
 TYPING M, THEN THE PREVIOUS
LY USED"
370 PRINT "COLOR WILL BE DRA
WN AT EACH SPOT YOU MOVE OVE
R."
380 PRINT
390 PRINT "THE BRUSH WILL BE
GIN IN THE MIDDLE OF THE SCR
EEN."
400 FOR I=1 TO 9
410 PRINT
420 NEXT I
430 INPUT "PRESS ENTER WHEN
READY TO   CONTINUE: ";ANS$
440 RETURN
450 CALL CLEAR
460 PRINT TAB(5);"*** ARTIST
 BOARD ***"
470 PRINT
480 RETURN
490 REM
500 REM ***SETUP***
510 REM
520 CALL CLEAR
530 FOR I=3 TO 16
540 CALL COLOR(I,I,I)
550 NEXT I
560 X1=12
570 Y1=16
580 CO=70
590 CALL HCHAR(12,16,70)
600 CALL HCHAR(1,2,66,30)
610 CALL HCHAR(24,2,66,30)
620 CALL VCHAR(1,2,66,24)
630 CALL VCHAR(1,32,66,24)
640 RETURN
650 REM
660 REM ***PLAY***
670 REM
680 GOSUB 1310
690 CALL KEY(0,N,T)
700 IF T=0 THEN 680
710 IF N=81 THEN 1340
720 IF N=77 THEN 800
730 IF N=90 THEN 1010
740 IF N=69 THEN 1030
750 IF N=83 THEN 1060
760 IF N=68 THEN 1090
```

```
770 IF N=88 THEN 1120
780 IF N=67 THEN 1150
790 GOTO 680
800 CALL KEY(0,N,T)
810 GOSUB 1310
820 IF T=0 THEN 800
830 CALL SOUND(100,500,3)
840 IF N=68 THEN 890
850 IF N=69 THEN 920
860 IF N=83 THEN 950
870 IF N=88 THEN 980
880 GOTO 710
890 IF Y1=31 THEN 800
900 Y1=Y1+1
910 GOTO 800
920 IF X1=2 THEN 800
930 X1=X1-1
940 GOTO 800
950 IF Y1=3 THEN 800
960 Y1=Y1-1
970 GOTO 800
980 IF X1=23 THEN 800
990 X1=X1+1
1000 GOTO 800
1010 CO=32
1020 GOTO 680
1030 IF X1=2 THEN 680
1040 X1=X1-1
1050 GOTO 1280
1060 IF Y1=3 THEN 680
1070 Y1=Y1-1
1080 GOTO 1280
1090 IF Y1=31 THEN 680
1100 Y1=Y1+1
1110 GOTO 1280
1120 IF X1=23 THEN 680
1130 X1=X1+1
1140 GOTO 1280
1150 CALL KEY(0,N1,T)
1160 IF T=0 THEN 1150
1170 IF T=-1 THEN 1150
1180 N1=N1-48
1190 IF (N1<1)+(N1=2)+(N1>9)
THEN 1150
1200 IF N1>1 THEN 1260
1210 N1=10
1220 CALL KEY(0,N2,T)
1230 IF T<>1 THEN 1220
1240 N2=N2-48
1250 IF (N2<1)+(N2>6)THEN 12
20
```

```
1260 CO=(N1+N2)*8+24
1270 N2=0
1280 CALL HCHAR(X1,Y1,CO)
1290 CALL SOUND(100,500,3)
1300 GOTO 680
1310 CALL GCHAR(X1,Y1,X2)
1320 CALL HCHAR(X1,Y1,42)
1330 CALL HCHAR(X1,Y1,X2)
1340 RETURN
```



Nim

Nim is a game of subtle logic that will take you minutes to learn and years to master. The concept is simple. The program will help you create two to six piles, each containing one to fifteen matches. Each turn, you will be asked to select a number of matches from a pile. When making your move, there are only two rules to follow: you must take at least one match each turn, and all the matches you take must be from the same pile. Once you have made your move, the computer will take its move, following the same rules that you do. To beat the computer, you have to take the last match from the board. You decide how many piles there will be and how many matches in each pile, so it should be easy, right? Well play a few games first and then decide!

The routines that draw the numbers on the piles are contained in lines 720-1520. Lines 2020-2500 contain the logic for the computer move. If you take a good look at these lines, you might be able to get some idea of how the computer makes its move.

```
10 REM ********************
20 REM ***              ***
30 REM ***      NIM     ***
40 REM ***              ***
50 REM ********************
60 GOSUB 110
70 GOSUB 460
80 GOSUB 1530
90 GOSUB 3580
100 END
110 REM ***INSTRUCTIONS***
120 GOSUB 380
130 PRINT "WELCOME TO THE GA
ME OF NIM. THIS GAME TESTS Y
OUR ABILITYTO THINK LOGICALL
Y.  FIRST,"
140 PRINT "YOU WILL BE ASKED
 TO CHOOSE THE NUMBER OF PIL
ES (2-6),  AND THEN HOW MANY
 MATCHES"
150 PRINT "IN EACH PILE (1-1
5)."
160 PRINT
170 PRINT "YOU WILL TAKE TUR
NS WITH THECOMPUTER REMOVING
 MATCHES   FROM A SELECTED P
ILE.  THE"
180 PRINT "ONE TO REMOVE THE
 LAST MATCHWINS."
190 PRINT
200 PRINT "PRESS THE SPACE B
AR TO      SELECT A PILE.  T
HE CURRENT PILE WILL BE HIGH
LIGHTED."
210 X=3
220 GOSUB 420
230 GOSUB 360
240 GOSUB 380
250 PRINT "NEXT, ENTER THE N
UMBER OF   MATCHES YOU WISH
TO TAKE.   IF YOU TAKE ONLY
ONE MATCH,"
260 PRINT "YOU MUST PRESS 'E
NTER' AFTERTYPING THE ONE."
270 PRINT
280 PRINT "THE COMPUTER WILL
 TAKE ITS  TURN NEXT, AND UP
DATE THE   SCREEN."
290 PRINT
300 PRINT "THIS GAME USES A
STRATEGY   THAT MAY TAKE SOM
E TIME TO  MASTER, BUT ONCE
YOU KNOW"
310 PRINT "THE SECRET, IT'S
```

```
STILL FUN."
320 PRINT
330 PRINT "GOOD LUCK!"
340 X=4
350 GOSUB 420
360 INPUT "PRESS ENTER WHEN
READY TO   CONTINUE: ":ANS$
370 RETURN
380 CALL CLEAR
390 PRINT TAB(9);"*** NIM **
*"
400 PRINT
410 RETURN
420 FOR I=1 TO X
430 PRINT
440 NEXT I
450 RETURN
460 REM ***SETUP***
470 CALL CLEAR
480 RANDOMIZE
490 DIM P(6),M0(27),M1(22),M
2(27),M3(20)
500 FOR I=10 TO 15
510 CALL COLOR(I,1,16)
520 NEXT I
530 CALL COLOR(16,16,16)
540 FOR I=104 TO 144 STEP 8
550 CALL CHAR(I,"3E7F7F7F7F7
F3E00")
560 NEXT I
570 FOR I=1 TO 27
580 M0(I)=ASC(SEG$("PRESS SP
ACE TO SELECT PILE.",I,1))
590 M2(I)=ASC(SEG$("THAT'S T
OO MANY, TRY AGAIN.",I,1))
600 NEXT I
610 FOR I=1 TO 22
620 M1(I)=ASC(SEG$("TAKE HOW
 MANY MATCHES?",I,1))
630 NEXT I
640 FOR I=1 TO 20
650 M3(I)=ASC(SEG$("I'LL TAK
E    MATCHES",I,1))
660 NEXT I
670 RETURN
680 FOR K=0 TO 7
690 CALL HCHAR(R+K,C,152,8)
700 NEXT K
710 RETURN
720 CALL VCHAR(R+1,C+4,D,6)
730 CALL HCHAR(R+2,C+3,D)
740 CALL HCHAR(R+6,C+3,D,3)
750 RETURN
760 CALL HCHAR(R+1,C+2,D,3)
```

```
770 CALL VCHAR(R+2,C+5,D,2)
780 CALL HCHAR(R+4,C+3,D,2)
790 CALL HCHAR(R+5,C+2,D)
800 CALL HCHAR(R+6,C+2,D,4)
810 RETURN
820 CALL HCHAR(R+1,C+2,D,3)
830 CALL HCHAR(R+2,C+5,D)
840 CALL HCHAR(R+3,C+3,D,2)
850 CALL VCHAR(R+4,C+5,D,2)
860 CALL HCHAR(R+6,C+2,D,3)
870 RETURN
880 CALL VCHAR(R+1,C+5,D,6)
890 CALL VCHAR(R+1,C+2,D,4)
900 CALL HCHAR(R+4,C+3,D,2)
910 RETURN
920 CALL HCHAR(R+1,C+2,D,4)
930 CALL HCHAR(R+2,C+2,D)
940 CALL HCHAR(R+3,C+2,D,3)
950 CALL VCHAR(R+4,C+5,D,2)
960 CALL HCHAR(R+6,C+2,D,3)
970 RETURN
980 CALL HCHAR(R+1,C+3,D,3)
990 CALL VCHAR(R+2,C+2,D,4)
1000 CALL HCHAR(R+3,C+3,D,2)
1010 CALL VCHAR(R+4,C+5,D,2)
1020 CALL HCHAR(R+6,C+3,D,2)
1030 RETURN
1040 CALL HCHAR(R+1,C+2,D,4)
1050 CALL HCHAR(R+2,C+5,D)
1060 CALL HCHAR(R+3,C+4,D)
1070 CALL VCHAR(R+4,C+3,D,3)
1080 RETURN
1090 CALL HCHAR(R+1,C+3,D,2)
1100 CALL HCHAR(R+2,C+2,D)
1110 CALL HCHAR(R+2,C+5,D)
1120 CALL HCHAR(R+3,C+3,D,2)
1130 CALL VCHAR(R+4,C+2,D,2)
1140 CALL VCHAR(R+4,C+5,D,2)
1150 CALL HCHAR(R+6,C+3,D,2)
1160 RETURN
1170 CALL HCHAR(R+1,C+3,D,2)
1180 CALL VCHAR(R+2,C+2,D,2)
1190 CALL VCHAR(R+2,C+5,D,5)
1200 CALL HCHAR(R+4.C+3,D,2)
1210 RETURN
1220 CALL VCHAR(R+1,C+1,D,6)
1230 CALL HCHAR(R+1,C+4,D,2)
1240 CALL VCHAR(R+2,C+3,D,4)
1250 CALL VCHAR(R+2,C+6,D,4)
1260 CALL HCHAR(R+6,C+4,D,2)
1270 RETURN
1280 CALL VCHAR(R+1,C+2,D,6)
1290 CALL HCHAR(R+2,C+1,D)
1300 CALL VCHAR(R+1,C+5,D,6)
```

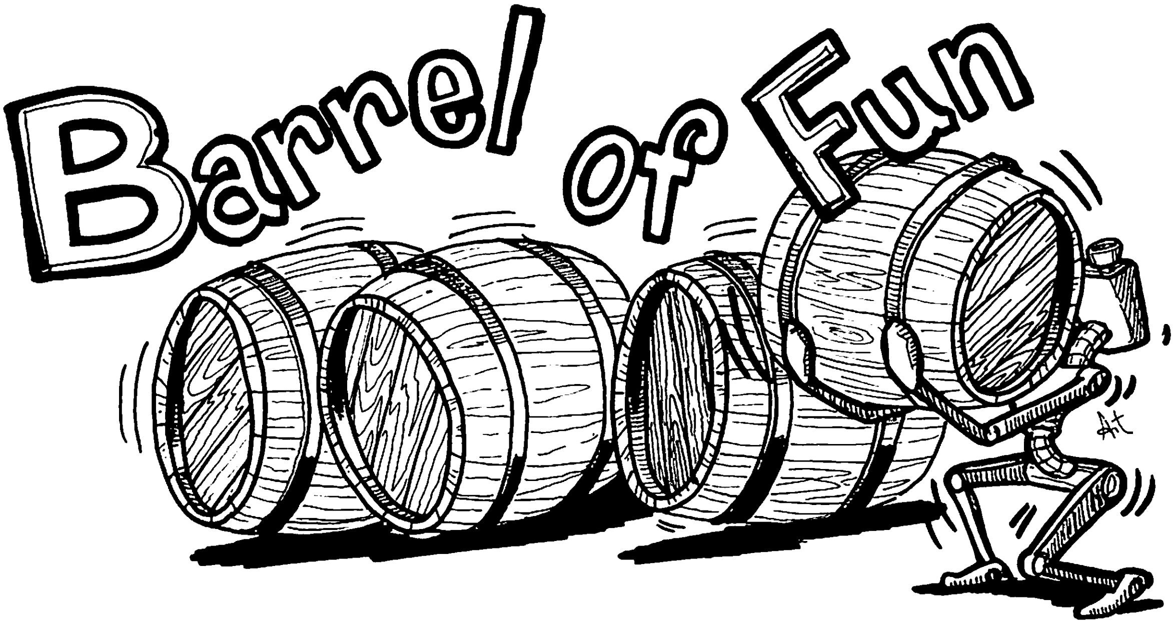
```
1310 CALL HCHAR(R+2,C+4,D)
1320 RETURN
1330 CALL VCHAR(R+1,C+1,D,6)
1340 C=C+1
1350 GOSUB 760
1360 C=C-1
1370 RETURN
1380 CALL VCHAR(R+1,C+1,D,6)
1390 C=C+1
1400 GOSUB 820
1410 C=C-1
1420 RETURN
1430 CALL VCHAR(R+1,C+1,D,6)
1440 C=C+1
1450 GOSUB 880
1460 C=C-1
1470 RETURN
1480 CALL VCHAR(R+1,C+1,D,6)
1490 C=C+1
1500 GOSUB 920
1510 C=C-1
1520 RETURN
1530 REM ***PLAY***
1540 CALL CLEAR
1550 FL=0
1560 INPUT "HOW MANY PILES?
(2-6) ":PL
1570 IF (PL<2)+(PL>6)+(PL<>I
NT(PL))THEN 1560
1580 PRINT
1590 FOR I=1 TO PL
1600 PRINT "HOW MANY MATCHES
 IN PILE";I;"?"
1610 INPUT "(1-15) ":P(I)
1620 IF (P(I)<1)+(P(I)>15)+(
P(I)<>INT(P(I)))THEN 1610
1630 NEXT I
1640 CALL CLEAR
1650 GOSUB 2590
1660 GOSUB 2620
1670 WH=0
1680 WH=WH+1
1690 IF WH>PL THEN 1670
1700 IF P(WH)=0 THEN 1680
1710 FOR I=1 TO PL
1720 IF I=WH THEN 1740
1730 CALL COLOR(9+I,1,16)
1740 NEXT I
1750 CALL COLOR(9+WH,2,16)
1760 CALL KEY(0,N,S)
1770 IF S<>1 THEN 1760
1780 IF N=32 THEN 1680
1790 N=N-48
1800 IF (N<1)+(N>9)THEN 1760
```

```
1810 CALL HCHAR(22,26,N+48)
1820 IF N>1 THEN 1900
1830 CALL KEY(0,N2,S)
1840 IF S<>1 THEN 1830
1850 IF N2=13 THEN 1900
1860 N2=N2-48
1870 IF (N2<0)+(N2>5)THEN 18
30
1880 CALL HCHAR(22,27,N2+48)
1890 N=N2+10
1900 IF P(WH)>=N THEN 1930
1910 GOSUB 2690
1920 GOTO 1760
1930 P(WH)=P(WH)-N
1940 CALL COLOR(9+WH,1,16)
1950 GOSUB 2590
1960 CALL HCHAR(20,3,32,28)
1970 CALL HCHAR(22,3,32,28)
1980 GOSUB 2530
1990 IF WL THEN 2020
2000 WL=1
2010 GOTO 3570
2020 REM  MACHINE'S MOVE
2030 GOSUB 3130
2040 FOR I=1 TO PL
2050 IF P(I)=0 THEN 2080
2060 NU=P(I)
2070 GOSUB 2510
2080 NEXT I
2090 QT=ABS(B3*8+B2*4+B1*2+B
0)
2100 IF QT>0 THEN 2160
2110 WH=INT(RND*PL)+1
2120 IF P(WH)<1 THEN 2110
2130 QT=INT(RND*P(WH))+1
2140 P(WH)=P(WH)-QT
2150 GOTO 2430
2160 J=1
2170 IF QT<2 THEN 2230
2180 J=2
2190 IF QT<4 THEN 2230
2200 J=4
2210 IF QT<8 THEN 2230
2220 J=8
2230 WH=0
2240 FOR I=1 TO PL
2250 IF P(I)<J THEN 2290
2260 IF WH=0 THEN 2280
2270 IF INT(RND*2)THEN 2290
2280 WH=I
2290 NEXT I
2300 IF P(WH)<>QT THEN 2330
2310 P(WH)=0
2320 GOTO 2430
```

```
2330 GOSUB 3130
2340 NU=P(WH)
2350 GOSUB 2510
2360 NU=QT
2370 GOSUB 2510
2380 QT=P(WH)
2390 P(WH)=ABS(B3*8+B2*4+B1*
2+B0)
2400 IF P(WH)<=QT THEN 2420
2410 P(WH)=P(WH)-2*J
2420 QT=QT-P(WH)
2430 CALL COLOR(9+WH,2,16)
2440 GOSUB 2770
2450 GOSUB 2590
2460 CALL HCHAR(20,3,32,20)
2470 GOSUB 2530
2480 IF WL=0 THEN 3570
2490 GOSUB 2620
2500 GOTO 1700
2510 ON NU GOSUB 3110,3060,3
100,3020,2930,3050,3090,2980
,2900,2970,2960,3010,3000,30
40,3080
2520 RETURN
2530 WL=0
2540 FOR I=1 TO PL
2550 IF P(I)=0 THEN 2570
2560 WL=-1
2570 NEXT I
2580 RETURN
2590 ON PL-1 GOSUB 3240,3330
,3420,3470,3520
2600 FL=1
2610 RETURN
2620 FOR I=1 TO 27
2630 CALL HCHAR(20,2+I,M0(I)
)
2640 NEXT I
2650 FOR I=1 TO 22
2660 CALL HCHAR(22,2+I,M1(I)
)
2670 NEXT I
2680 RETURN
2690 FOR I=1 TO 27
2700 CALL HCHAR(24,2+I,M2(I)
)
2710 NEXT I
2720 FOR I=1 TO 200
2730 NEXT I
2740 CALL HCHAR(24,3,32,28)
2750 CALL HCHAR(22,26,32,2)
2760 RETURN
2770 LE=18
2780 IF QT=1 THEN 2840
```

```
2790 LE=20
2800 IF QT<10 THEN 2840
2810 M3(11)=49
2820 M3(12)=38+QT
2830 GOTO 2860
2840 M3(11)=32
2850 M3(12)=48+QT
2860 FOR I=1 TO LE
2870 CALL HCHAR(20,2+I,M3(I)
)
2880 NEXT I
2890 RETURN
2900 B3=(B3=0)
2910 B0=(B0=0)
2920 RETURN
2930 B2=(B2=0)
2940 B0=(B0=0)
2950 RETURN
2960 B0=(B0=0)
2970 B1=(B1=0)
2980 B3=(B3=0)
2990 RETURN
3000 B0=(B0=0)
3010 B3=(B3=0)
3020 B2=(B2=0)
3030 RETURN
3040 B3=(B3=0)
3050 B2=(B2=0)
3060 B1=(B1=0)
3070 RETURN
3080 B3=(B3=0)
3090 B2=(B2=0)
3100 B1=(B1=0)
3110 B0=(B0=0)
3120 RETURN
3130 B0=0
3140 B1=0
3150 B2=0
3160 B3=0
3170 RETURN
3180 IF (FL=1)*(I<>WH)THEN 3
230
3190 GOSUB 680
3200 D=96+8*I
3210 IF P(I)=0 THEN 3230
3220 ON P(I)GOSUB 720,760,82
0,880,920,980,1040,1090,1170
,1220,1280,1330,1380,1430,14
80
3230 RETURN
3240 I=1
3250 R=6
3260 C=8
3270 FOR J=1 TO 2
```

```
3280 GOSUB 3180
3290 C=C+10
3300 I=I+1
3310 NEXT J
3320 RETURN
3330 I=1
3340 R=6
3350 C=3
3360 FOR J=1 TO 3
3370 GOSUB 3180
3380 C=C+10
3390 I=I+1
3400 NEXT J
3410 RETURN
3420 I=1
3430 R=1
3440 GOSUB 3260
3450 R=11
3460 GOTO 3260
3470 I=1
3480 R=1
3490 GOSUB 3260
3500 R=11
3510 GOTO 3350
3520 I=1
3530 R=1
3540 GOSUB 3350
3550 R=11
3560 GOTO 3350
3570 RETURN
3580 REM ***END***
3590 FOR I=1 TO 200
3600 NEXT I
3610 GOSUB 380
3620 X=4
3630 GOSUB 420
3640 IF WL=1 THEN 3690
3650 PRINT "SORRY, YOU LOST.
.."
3660 PRINT
3670 PRINT "YOU'LL HAVE BETT
ER LUCK      NEXT TIME."
3680 GOTO 3720
3690 PRINT "CONGRATULATIONS!
!!"
3700 PRINT
3710 PRINT "YOU BEAT THE COM
PUTER!       GOOD JOB!!"
3720 X=10
3730 GOSUB 420
3740 RETURN
```



The straightforward use of graphics makes this an excellent program. The object of the game is to align the rows so that each one is a solid color.

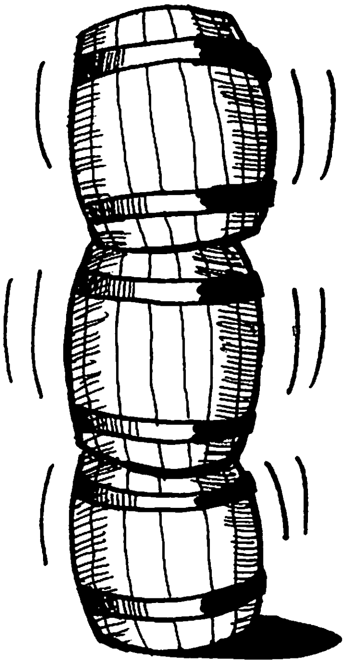
Run the program before continuing with the analysis. Now that you know how the program should look, by making some changes, the line functions should become more clear. Change 750-810 so that each background color is medium green (color 3): 750 CALL COLOR(9,5,3), 760 CALL COLOR(10,7,3), . . . 810 CALL COLOR(15,15,3). Rather than getting a solidly colored print block, you get a character printed on a green background. Change line 870 to: 870 FOR X=11 TO 23 STEP 6. Experiment with any line whose function is unclear.

```
10 REM ********************
20 REM ***              ***
30 REM *** BARREL OF FUN***
40 REM ***              ***
50 REM ********************
60 GOSUB 110
70 GOSUB 560
80 GOSUB 1200
90 GOSUB 1990
100 END
110 REM
120 REM ***INSTRUCTIONS***
130 REM
140 GOSUB 520
150 PRINT "IN THIS GAME YOU
ARE        PRESENTED WITH TH
REE COLUMNSOF SIX COLORFUL B
OXES."
160 PRINT
170 PRINT "THE OBJECT OF THE
 GAME IS   TO ALIGN THE ROWS
 SUCH THAT EACH ROW IS A SOL
ID COLOR."
180 PRINT
190 PRINT "THERE ARE ONLY TW
O DARK RED RECTANGLES."
200 PRINT
210 PRINT "THE EMPTY SPACE W
ILL        COMPLETE THE ABOV
E ROW."
220 FOR I=1 TO 7
230 PRINT
240 NEXT I
250 GOSUB 500
260 GOSUB 520
270 PRINT "EACH COLUMN OF BO
XES CAN    BE ROTATED VERTIC
ALLY BY    ENTERING THE NUMB
ER OF THE"
280 PRINT "COLUMN YOU WISH T
O ROTATE."
290 PRINT
300 PRINT "YOU CAN MOVE A CO
LORED BOX  INTO THE EMPTY SP
ACE BY     USING THE APPROPR
IATE ARROW KEY."
310 PRINT
320 PRINT "FOR INSTANCE, IF
YOU WANT TOMOVE A COLORED BO
X INTO THE EMPTY SPACE ON IT
S RIGHT,"
```

```
330 PRINT "THEN PRESS THE RI
GHT ARROW."
340 FOR I=1 TO 6
350 PRINT
360 NEXT I
370 GOSUB 500
380 GOSUB 520
390 PRINT "NOTE THAT IN THE
FINAL      SOLUTION, ALL HOR
IZONTAL    ROWS MUST BE THE
SAME COLOR."
400 PRINT
410 PRINT "IT DOESN'T MATTER
 WHICH ROW IS WHICH COLOR, A
S LONG AS  ALL COLORS WITHIN
 A ROW ARE THE SAME."
420 PRINT
430 PRINT "THE GAMEBOARD WIL
L BE MIXED UP, AND YOUR TASK
 IS TO     UNSCRAMBLE IT.  T
HE FINAL"
440 PRINT "PRODUCT SHOULD CO
NTAIN SIX  EVENLY COLORED RO
WS."
450 PRINT
460 PRINT "PRESS Q TO QUIT."
470 FOR I=1 TO 4
480 PRINT
490 NEXT I
500 INPUT "PRESS ENTER WHEN
READY TO   CONTINUE: ":ANS$
510 RETURN
520 CALL CLEAR
530 PRINT TAB(4);"*** BARREL
 OF FUN ***"
540 PRINT
550 RETURN
560 REM
570 REM ***SETUP***
580 REM
590 CALL CLEAR
600 RANDOMIZE
610 DIM BA(3,6),CO(6)
620 FOR I=1 TO 6
630 CO(I)=88+8*I
640 NEXT I
650 FOR I=1 TO 6
660 FOR J=1 TO 3
670 BA(J,I)=CO(I)
680 NEXT J
690 NEXT I
```

```
700 PRINT "TYPE 1 TO ROTATE
COLUMN 1,  2 TO ROTATE COLUM
N 2, AND 3 TO ROTATE COLUMN
3. USE THE"
710 PRINT "RIGHT ARROW (D KE
Y) TO MOVE RIGHT, AND THE LE
FT ARROW   (S KEY) TO MOVE L
EFT."
720 FOR I=1 TO 3
730 CALL HCHAR(2,9+4*I,48+I)
740 NEXT I
750 CALL COLOR(9,5,5)
760 CALL COLOR(10,7,7)
770 CALL COLOR(11,16,16)
780 CALL COLOR(12,2,2)
790 CALL COLOR(13,10,10)
800 CALL COLOR(14,14,14)
810 CALL COLOR(15,3,3)
820 FOR I=4 TO 16 STEP 2
830 FOR J=11 TO 23
840 CALL HCHAR(I,J,42)
850 NEXT J
860 NEXT I
870 FOR X=11 TO 23 STEP 4
880 FOR Y=5 TO 15
890 CALL VCHAR(Y,X,42)
900 NEXT Y
910 NEXT X
920 FOR I=1 TO 6
930 FOR J=1 TO 3
940 FOR K=8 TO 10
950 CALL HCHAR(3+2*I,K+J*4,B
A(J,I))
960 NEXT K
970 NEXT J
980 NEXT I
990 BA(3,2)=145
1000 FOR I=20 TO 22
1010 CALL HCHAR(7,I,BA(3,2))
1020 NEXT I
1030 FOR I=1 TO 6
1040 FOR J=1 TO 3
1050 X=INT(RND*6)+1
1060 Y=INT(RND*3)+1
1070 T=BA(J,I)
1080 BA(J,I)=BA(Y,X)
1090 BA(Y,X)=T
1100 NEXT J
1110 NEXT I
1120 FOR I=1 TO 6
1130 FOR J=1 TO 3
```

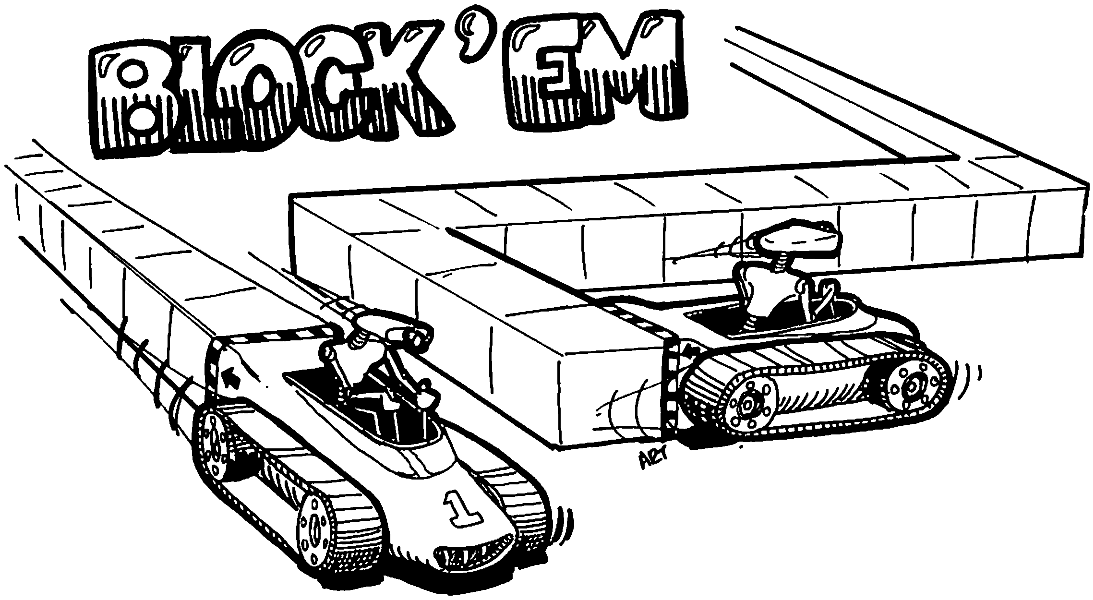
```
1140 FOR K=8 TO 10
1150 CALL HCHAR(3+2*I,K+J*4,
BA(J,I))
1160 NEXT K
1170 NEXT J
1180 NEXT I
1190 RETURN
1200 REM
1210 REM ***PLAY***
1220 REM
1230 CALL KEY(0,N,T)
1240 IF T=0 THEN 1230
1250 IF N=49 THEN 1410
1260 IF N=50 THEN 1530
1270 IF N=51 THEN 1550
1280 IF N=68 THEN 1570
1290 IF N=83 THEN 1570
1300 IF N=81 THEN 1970
1310 GOTO 1860
1320 GT=0
1330 FOR I=1 TO 6
1340 FOR J=1 TO 2
1350 IF BA(J,I)=BA(J+1,I)THE
N 1380
1360 IF BA(3,I)=145 THEN 138
0
1370 GT=88
1380 NEXT J
1390 NEXT I
1400 IF GT=0 THEN 1950 ELSE
1230
1410 C=1
1420 T=BA(C,1)
1430 FOR I=1 TO 5
1440 BA(C,I)=BA(C,I+1)
1450 NEXT I
1460 BA(C,6)=T
1470 FOR I=1 TO 6
1480 FOR K=8 TO 10
1490 CALL HCHAR(3+2*I,K+4*C,
BA(C,I))
1500 NEXT K
1510 NEXT I
1520 GOTO 1320
1530 C=2
1540 GOTO 1420
1550 C=3
1560 GOTO 1420
1570 BX=1
1580 BY=1
1590 IF BA(BY,BX)=145 THEN 1
660
```



```
1600 IF BY=3 THEN 1630
1610 BY=BY+1
1620 GOTO 1590
1630 BY=1
1640 BX=BX+1
1650 GOTO 1590
1660 IF N=68 THEN 1830
1670 IF BY=3 THEN 1860
1680 CC=1
1690 T=BA(BY,BX)
1700 BA(BY,BX)=BA(BY+CC,BX)
1710 BA(BY+CC,BX)=T
1720 FOR K=8 TO 10
1730 CALL HCHAR(3+BX*2,K+BY*
4,BA(BY,BX))
1740 CALL HCHAR(3+BX*2,K+(BY
+CC)*4,BA(BY+CC,BX))
1750 NEXT K
1760 GOTO 1320
1770 FOR I=1 TO 6
1780 FOR K=8 TO 10
1790 CALL HCHAR(3+2*I,K+4*C,
BA(C,I)
1800 NEXT K
1810 NEXT I
1820 GOTO 1320
1830 IF BY=1 THEN 1860
1840 CC=-1
1850 GOTO 1690
1860 FOR I=1 TO 9
1870 CALL HCHAR(1,12+I,ASC(S
EG$("TRY AGAIN",I,1)))
1880 NEXT I
1890 FOR I=1 TO 200
1900 NEXT I
1910 FOR I=1 TO 9
1920 CALL HCHAR(1,12+I,32)
1930 NEXT I
1940 GOTO 1230
1950 W=1
1960 GOTO 1980
1970 W=0
1980 RETURN
1990 REM
2000 REM ***END***
2010 REM
2020 IF W=0 THEN 2150
2030 FOR I=1 TO 200
2040 NEXT I
2050 CALL CLEAR
2060 FOR I=1 TO 18
```

```
2070 CALL HCHAR(12,I+7,ASC(S
EG$("CONGRATULATIONS!!!",I,1
)))
2080 NEXT I
2090 FOR I=1 TO 200
2100 NEXT I
2110 PRINT "YOU DID IT!  YOU
 COMPLETELY UNSCRAMBLED THE
GAME BOARD!           GOOD JO
B!"
2120 FOR I=1 TO 4
2130 PRINT
2140 NEXT I
2150 RETURN
```

BLOCK'EM



The object of this game is to draw a longer line than your opponent draws. If your progress is impeded by a border or by the opponent's line, you lose. This is a two man game.

List 390-420. These lines draw the border around the field of play. Change line 390 to: 390 CALL HCHAR(1,5,120,20). Change line 400 to: 400 CALL HCHAR(24,5,120,20). Type RUN 70 (this will skip the instructions). You will see an ill-fitting border. Change line 410 to: 410 CALL VCHAR(1,5,120,24). Change line 420 to: 420 CALL VCHAR(1,24,120,24). Run the program. The border will be uniform but too narrow.

To undo the changes, reload the program. Next, list 530-630. Line 530 accepts a special numeric value for a single keyboard character and stores the value in N. The 'key-unit' or first number in parentheses tells the computer how the keyboard is being used (mapped) and what number to store in N for each key pressed. The chart below will help to explain lines 530-550 and 580-600.

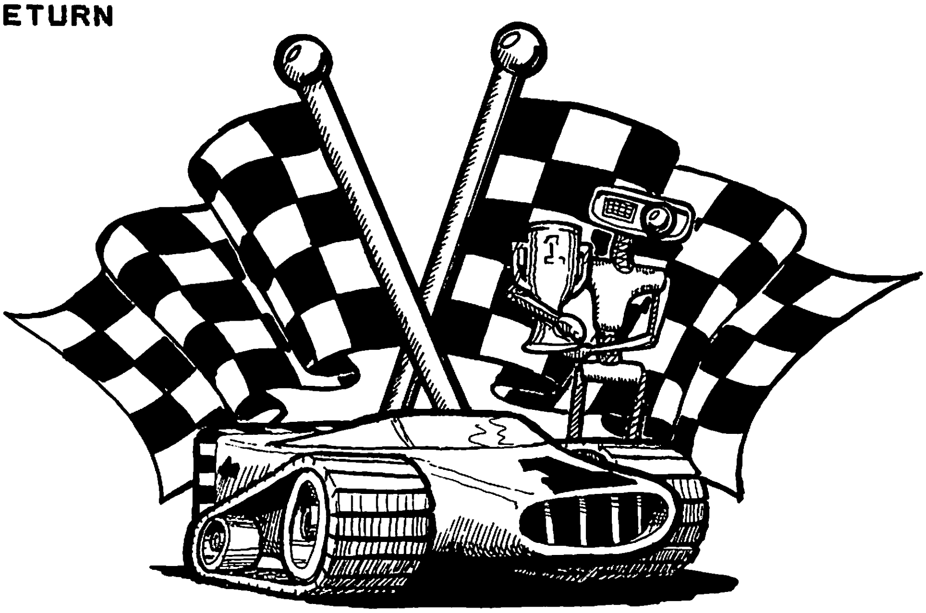
KEY-UNIT	NUMERIC VALUE	CHARACTER	DIRECTION
1	5	E	up
1	2	S	left
1	3	D	right
1	0	X	down
2	5	I	up
2	2	J	left
2	3	K	right
2	0	M	down

The letters E, S, D and X are the direction keys for player #1; I, J, K and M are for player #2.

```
10 REM ********************
20 REM ***              ***
30 REM ***   BLOCK 'EM  ***
40 REM ***              ***
50 REM ********************
60 GOSUB 110
70 GOSUB 320
80 GOSUB 500
90 GOSUB 1050
100 END
110 REM
120 REM ***INSTRUCTIONS***
130 REM
140 GOSUB 280
150 PRINT "THIS GAME PITS TW
O PLAYERS  AGAINST ONE ANOTH
ER."
160 PRINT
170 PRINT "EACH PLAYER DRAWS
 A LINE,   AND THE FIRST PLA
YER TO RUN INTO ANY LINE OR
BOUNDARY   IS THE LOSER."
180 PRINT
190 PRINT "THE CHART BELOW S
HOWS WHICH KEYS ARE RELEVANT
 TO EACH   PLAYER."
200 PRINT
210 PRINT "   PLAYER #1    P
LAYER #2      ---------    -
--------"
220 PRINT
230 PRINT "      E      UP
    I           S D    L<>R
   J K           X      DN
    M"
240 PRINT
250 PRINT
260 INPUT "PRESS ENTER WHEN
READY TO   CONTINUE: ":ANS$
270 RETURN
280 CALL CLEAR
290 PRINT TAB(7);"*** BLOCK
'EM ***"
300 PRINT
310 RETURN
320 REM
330 REM ***SETUP***
340 REM
350 CALL CLEAR
360 CALL COLOR(12,2,2)
370 CALL COLOR(13,5,5)
```

```
380 CALL COLOR(14,7,7)
390 CALL HCHAR(1,2,120,31)
400 CALL HCHAR(24,2,120,31)
410 CALL VCHAR(1,2,120,24)
420 CALL VCHAR(1,32,120,24)
430 X1=5
440 Y1=5
450 X2=18
460 Y2=31
470 P1=3
480 P2=2
490 RETURN
500 REM
510 REM ***PLAY***
520 REM
530 CALL KEY(1,N,T)
540 IF (N<0)+(N>5)THEN 560
550 ON N+1 GOTO 730,560,670,
700,560,640
560 N=P1
570 ON N+1 GOTO 730,580,670,
700,580,640
580 CALL KEY(2,N,T)
590 IF (N<0)+(N>5)THEN 610
600 ON N+1 GOTO 900,610,840,
870,610,810
610 N=P2
620 ON N+1 GOTO 900,630,840,
870,630,810
630 GOTO 530
640 REM  PLAYER #1 UP
650 X1=X1-1
660 GOTO 750
670 REM  PLAYER #1 LEFT
680 Y1=Y1-1
690 GOTO 750
700 REM  PLAYER #1 RIGHT
710 Y1=Y1+1
720 GOTO 750
730 REM  PLAYER #1 DOWN
740 X1=X1+1
750 CALL GCHAR(X1,Y1,X3)
760 IF X3<>32 THEN 1010
770 CALL HCHAR(X1,Y1,130)
780 CALL SOUND(-60,220,0)
790 P1=N
800 GOTO 580
810 REM  PLAYER #2 UP
820 X2=X2-1
830 GOTO 920
840 REM  PLAYER #2 LEFT
```

```
850 Y2=Y2-1
860 GOTO 920
870 REM  PLAYER #2 RIGHT
880 Y2=Y2+1
890 GOTO 920
900 REM  PLAYER #2 DOWN
910 X2=X2+1
920 CALL GCHAR(X2,Y2,X3)
930 IF X3<>32 THEN 1030
940 CALL HCHAR(X2,Y2,138)
950 CALL SOUND(-60,330,0)
960 DC=DC+1
970 FOR I=1 TO 100-DC
980 NEXT I
990 P2=N
1000 GOTO 530
1010 W=2
1020 GOTO 1040
1030 W=1
1040 RETURN
1050 REM
1060 REM ***END***
1070 REM
1080 FOR I=1 TO 200
1090 NEXT I
1100 CALL CLEAR
1110 PRINT "CONGRATULATIONS
TO PLAYER";W;"FOR A WINNING
EFFORT."
1120 FOR I=1 TO 12
1130 PRINT
1140 NEXT I
1150 RETURN
```



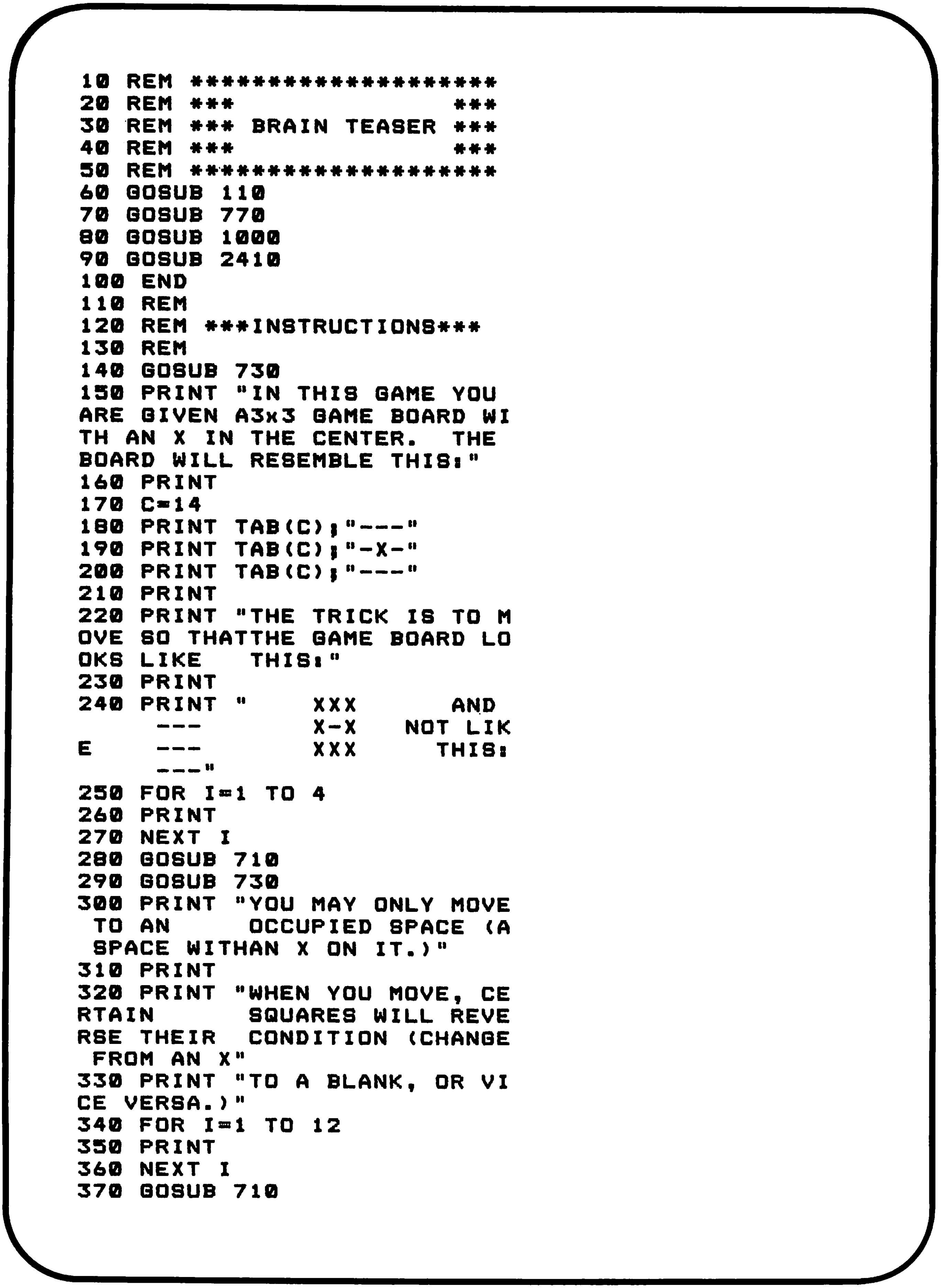
BRAIN TEASER

This game is an intellectual challenge. Random guessing will rarely net a correct solution. The object is to manipulate the board so that eight of the gameboard squares (all but the center square) are filled with X's. This may sound easy, but it's not. After learning the rules and playing a game, you will find it very helpful to conceptualize a basic game plan.

The graphics are straightforward. Look at line 820. Character Set nine (#9) is assigned background and foreground color five (5) or dark blue. Change the line so that it reads: CALL COLOR(9,7,7) or CALL COLOR(9,5,2) or CALL COLOR(4,5,5). Run the program after each change.

Change line 860 so that it reads: CALL VCHAR(7,13+I,84,7). Not only is the block drawn from left to right instead of top to bottom, but the character has been changed from one of the blue on blue characters of Character Set #9, to a T (ASCII value of 84). Run the program to see the changes.

Now list 1030-1110. Line 1030 stores a value in N (the number which is typed should be between 1 and 9). The storage location N will not store 1, 2, . . . 9; instead it will store the corresponding ASCII value (49-57). The two main symbols used in this game are a blank space (ASCII value=32) and an X (ASCII value=88).



```
10 REM ********************
20 REM ***              ***
30 REM *** BRAIN TEASER ***
40 REM ***              ***
50 REM ********************
60 GOSUB 110
70 GOSUB 770
80 GOSUB 1000
90 GOSUB 2410
100 END
110 REM
120 REM ***INSTRUCTIONS***
130 REM
140 GOSUB 730
150 PRINT "IN THIS GAME YOU
ARE GIVEN A3x3 GAME BOARD WI
TH AN X IN THE CENTER.  THE
BOARD WILL RESEMBLE THIS:"
160 PRINT
170 C=14
180 PRINT TAB(C);"---"
190 PRINT TAB(C);"-X-"
200 PRINT TAB(C);"---"
210 PRINT
220 PRINT "THE TRICK IS TO M
OVE SO THATTHE GAME BOARD LO
OKS LIKE   THIS:"
230 PRINT
240 PRINT "    XXX      AND
     ---         X-X   NOT LIK
E    ---         XXX     THIS:
     ---"
250 FOR I=1 TO 4
260 PRINT
270 NEXT I
280 GOSUB 710
290 GOSUB 730
300 PRINT "YOU MAY ONLY MOVE
 TO AN     OCCUPIED SPACE (A
 SPACE WITHAN X ON IT.)"
310 PRINT
320 PRINT "WHEN YOU MOVE, CE
RTAIN      SQUARES WILL REVE
RSE THEIR  CONDITION (CHANGE
 FROM AN X"
330 PRINT "TO A BLANK, OR VI
CE VERSA.)"
340 FOR I=1 TO 12
350 PRINT
360 NEXT I
370 GOSUB 710
```

```
380 GOSUB 730
390 PRINT "IF YOU MOVE TO A
CORNER,    ALL OF THE ADJACE
NT SQUARES REVERSE."
400 PRINT
410 PRINT "IF YOU MOVE TO TH
E MIDDLE OFA SIDE, ALL OF TH
E SQUARES  ON THAT SIDE WILL
 REVERSE."
420 PRINT
430 PRINT "IF YOU CHOOSE THE
 CENTER    SQUARE, THAT BOX
AND THE    FOUR MIDDLE BOXES
 WILL ALL  REVERSE."
440 FOR I=1 TO 8
450 PRINT
460 NEXT I
470 GOSUB 710
480 GOSUB 730
490 PRINT "HERE IS A QUICK R
EVIEW OF   THE VARIOUS MOVES
, AND THE  RESULTING REVERSA
LS."
500 PRINT
510 PRINT "    M*-
    ---        **-      AND
    ---        ---
    *M*"
520 PRINT
530 PRINT "    --*
    -*-        --M      AND
    *M*        --*
    -*-"
540 PRINT
550 PRINT "THE M DENOTES THE
 MOVE, AND THE ASTERISKS DEN
OTE WHICH  SQUARES WILL REVE
RSE."
560 FOR I=1 TO 5
570 PRINT
580 NEXT I
590 GOSUB 710
600 GOSUB 730
610 PRINT "THE SQUARE THAT Y
OU MOVE TO WILL ALSO BE REVE
RSED."
620 PRINT
630 PRINT "THE BOARD IS DESI
GNATED BY  POSITION:"
640 PRINT
650 PRINT TAB(C);"123"
```

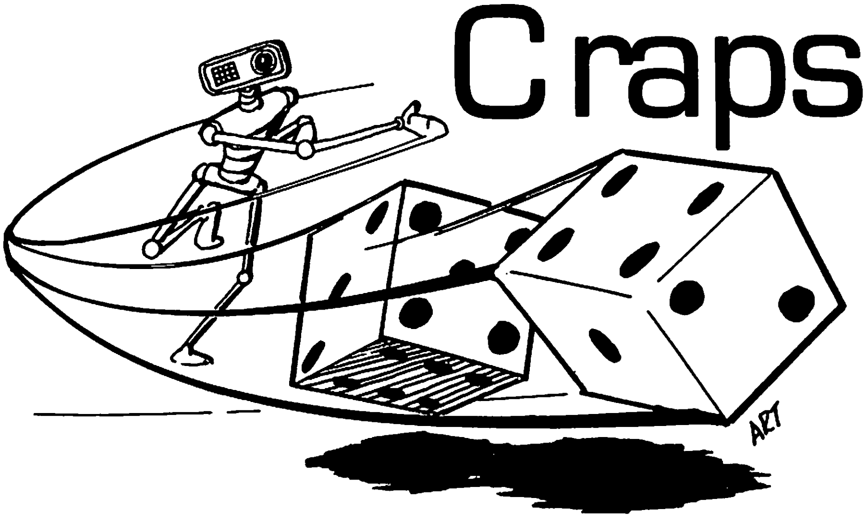
```
660 PRINT TAB(C);"456"
670 PRINT TAB(C);"789"
680 FOR I=1 TO 11
690 PRINT
700 NEXT I
710 INPUT "PRESS ENTER WHEN
READY TO   CONTINUE: ":ANS$
720 RETURN
730 CALL CLEAR
740 PRINT TAB(5);"*** BRAIN
TEASER ***"
750 PRINT
760 RETURN
770 REM
780 REM ***SETUP***
790 REM
800 CALL CLEAR
810 DIM GB(9)
820 CALL COLOR(9,5,5)
830 PRINT "ENTER THE NUMBER
OF THE      SQUARE TO WHICH Y
OU WANT TO MOVE (1-9)."
840 REM  DRAW GAME BOARD
850 FOR I=1 TO 7
860 CALL HCHAR(6+I,14,96,7)
870 NEXT I
880 FOR I=2 TO 6 STEP 2
890 FOR J=2 TO 6 STEP 2
900 CALL HCHAR(6+I,13+J,32)
910 NEXT J
920 NEXT I
930 CALL HCHAR(10,17,88)
940 FOR I=1 TO 9
950 GB(I)=32
960 NEXT I
970 GB(5)=88
980 RV=56
990 RETURN
1000 REM
1010 REM ***PLAY***
1020 REM
1030 CALL KEY(0,N,T)
1040 IF T=0 THEN 1030
1050 N=N-48
1060 IF (N<1)+(N>9)THEN 2210
1070 IF GB(N)<>88 THEN 2210
1080 ON N GOTO 1110,1240,133
0,1460,1550,1630,1720,1850,1
940
1090 IF N=5 THEN 1570
1100 IF N<>1 THEN 1140
```

```
1110 GB(1)=32
1120 IF GB(2)=32 THEN 1140
1130 RV=-56
1140 GB(2)=GB(2)+RV
1150 RV=56
1160 IF GB(4)=32 THEN 1180
1170 RV=-56
1180 GB(4)=GB(4)+RV
1190 RV=56
1200 IF GB(5)=32 THEN 1220
1210 RV=-56
1220 GB(5)=GB(5)+RV
1230 GOTO 2060
1240 GB(2)=32
1250 IF GB(1)=32 THEN 1270
1260 RV=-56
1270 GB(1)=GB(1)+RV
1280 RV=56
1290 IF GB(3)=32 THEN 1310
1300 RV=-56
1310 GB(3)=GB(3)+RV
1320 GOTO 2060
1330 GB(3)=32
1340 IF GB(2)=32 THEN 1360
1350 RV=-56
1360 GB(2)=GB(2)+RV
1370 RV=56
1380 IF GB(5)=32 THEN 1400
1390 RV=-56
1400 GB(5)=GB(5)+RV
1410 RV=56
1420 IF GB(6)=32 THEN 1440
1430 RV=-56
1440 GB(6)=GB(6)+RV
1450 GOTO 2060
1460 GB(4)=32
1470 IF GB(1)=32 THEN 1490
1480 RV=-56
1490 GB(1)=GB(1)+RV
1500 RV=56
1510 IF GB(7)=32 THEN 1530
1520 RV=-56
1530 GB(7)=GB(7)+RV
1540 GOTO 2060
1550 GB(5)=32
1560 FOR I=2 TO 8 STEP 2
1570 IF GB(I)=32 THEN 1590
1580 RV=-56
1590 GB(I)=GB(I)+RV
1600 RV=56
1610 NEXT I
```

```
1620 GOTO 2060
1630 GB(6)=32
1640 IF GB(3)=32 THEN 1660
1650 RV=-56
1660 GB(3)=GB(3)+RV
1670 RV=56
1680 IF GB(9)=32 THEN 1700
1690 RV=-56
1700 GB(9)=GB(9)+RV
1710 GOTO 2060
1720 GB(7)=32
1730 IF GB(4)=32 THEN 1750
1740 RV=-56
1750 GB(4)=GB(4)+RV
1760 RV=56
1770 IF GB(5)=32 THEN 1790
1780 RV=-56
1790 GB(5)=GB(5)+RV
1800 RV=56
1810 IF GB(8)=32 THEN 1830
1820 RV=-56
1830 GB(8)=GB(8)+RV
1840 GOTO 2060
1850 GB(8)=32
1860 IF GB(7)=32 THEN 1880
1870 RV=-56
1880 GB(7)=GB(7)+RV
1890 RV=56
1900 IF GB(9)=32 THEN 1920
1910 RV=-56
1920 GB(9)=GB(9)+RV
1930 GOTO 2060
1940 GB(9)=32
1950 IF GB(5)=32 THEN 1970
1960 RV=-56
1970 GB(5)=GB(5)+RV
1980 RV=56
1990 IF GB(6)=32 THEN 2010
2000 RV=-56
2010 GB(6)=GB(6)+RV
2020 RV=56
2030 IF GB(8)=32 THEN 2050
2040 RV=-56
2050 GB(8)=GB(8)+RV
2060 A=0
2070 FOR I=1 TO 3
2080 FOR J=1 TO 3
2090 A=A+1
2100 CALL HCHAR(6+2*I,13+2*J
,GB(A))
2110 NEXT J
```

```
2120 NEXT I
2130 RV=56
2140 FOR I=1 TO 9
2150 IF GB(I)=32 THEN 2170
2160 GT=GT+1
2170 NEXT I
2180 IF (GB(5)=32)*(GT=8)THE
N 2370
2190 GT=0
2200 GOTO 2300
2210 FOR I=1 TO 9
2220 CALL HCHAR(3,12+I,ASC(S
EG$("TRY AGAIN",I,1)))
2230 NEXT I
2240 FOR I=1 TO 100
2250 NEXT I
2260 FOR I=1 TO 9
2270 CALL HCHAR(3,12+I,32)
2280 NEXT I
2290 GOTO 1030
2300 FOR I=1 TO 9
2310 IF GB(I)=32 THEN 2330
2320 GT=99
2330 NEXT I
2340 IF GT=0 THEN 2400
2350 GT=0
2360 GOTO 1030
2370 W=5
2380 FOR I=1 TO 400
2390 NEXT I
2400 RETURN
2410 REM
2420 REM ***END***
2430 REM
2440 IF W<>5 THEN 2540
2450 CALL CLEAR
2460 A$="CONGRATULATIONS!!!"
2470 FOR I=1 TO LEN(A$)
2480 CALL HCHAR(12,I+7,ASC(S
EG$(A$,I,1)))
2490 NEXT I
2500 FOR I=1 TO 200
2510 NEXT I
2520 PRINT "GOOD JOB!"
2530 PRINT "THIS GAME IS NOT
 EASY."
2540 RETURN
```

Craps



Craps is a simplified version of the popular dice game. The rules are as follows: The dice are rolled. If the total is 7 or 11, you win. If the total is 2, 3 or 12, you lose. If the total is 4, 5, 6, 8, 9 or 10, then the total is referred to as your point. The game continues until you either roll your point (you win!) or until you roll a 7 (you crap out). The game ends in one of two ways: Either you run out of money, or you bet 0000 (nothing).

Type: LIST 570-620. Line 570 sets Character Set CHNO10 to print blue on blue (color 5 is deep blue). Lines 590-620 draw the blue frame on the screen. Lines 590 and 600 are slightly different from other CALL VCHAR statements we have seen.

It has been explained that a CALL HCHAR or CALL VCHAR requires three pieces of information (Row, Column, ASCII value of the character to be printed). Not only does line 590 contain these three necessary items (2, 3, 110), but it also includes a fourth number - 22. What this does is to print character 110 (110 is a lower case "n") in blue on a blue background, twenty-two times, in a vertical column. The first two numbers (2, 3) specify the beginning print position.

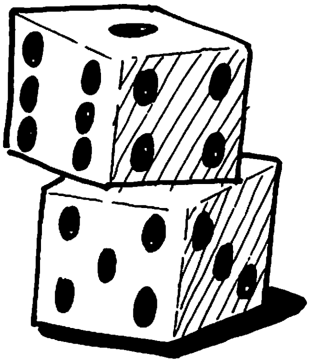
In other words, beginning at row 2 column 3, print 22 blue on blue lower case n's in a vertical column. Notice that lines 610 and 620 CALL a Horizontal CHARacter (HCHAR). The difference between HCHAR and VCHAR is the direction in which repetitions are plotted. HCHAR causes repetitions to be drawn in a row, VCHAR causes repetitions to be drawn in a column.

To better understand the function of any one line, type in the line number and then press ENTER. Now when you run the program, the lack of some attribute may highlight the line's function. Once you have a solid understanding of this program, you can "fix" the dice in any number of ways.

```
10 REM ********************
20 REM ***              ***
30 REM ***     CRAPS    ***
40 REM ***              ***
50 REM ********************
60 GOSUB 110
70 GOSUB 510
80 GOSUB 810
90 GOSUB 2360
100 END
110 REM
120 REM ***INSTRUCTIONS***
130 REM
140 GOSUB 470
150 PRINT "THIS IS A DICE GA
ME CALLED"
160 PRINT
170 PRINT TAB(12);"'CRAPS'"
180 PRINT
190 PRINT "TO PLAY, YOU WAGE
R A PORTIONOF YOUR MONEY ON
A ROLL OF  THE DICE.  HERE A
RE THE     RULES...."
200 PRINT
210 PRINT "YOU WIN IF THE FI
RST ROLL ISA 7 OR AN 11. CON
VERSELY,   YOU LOSE IF THE F
IRST ROLL  IS 2, 3 OR 12."
220 FOR I=1 TO 7
230 PRINT
240 NEXT I
250 GOSUB 450
260 GOSUB 470
270 PRINT "IF YOU GET A 4,5,
6,8,9, OR  10 ON YOUR FIRST
ROLL, IT ISREFERRED TO AS YO
UR 'POINT.'"
280 PRINT
290 PRINT "YOU MUST CONTINUE
 ROLLING   UNTIL YOU 1) ROLL
 A SEVEN   (CRAP OUT), OR 2)
 ROLL A"
300 PRINT "NUMBER EQUAL TO Y
OUR POINT  (YOU WIN!)."
310 PRINT
320 PRINT "TO QUIT THE GAME,
 BET ZERO  (0000) DOLLARS."
330 FOR I=1 TO 8
340 PRINT
350 NEXT I
360 GOSUB 450
```

```
370 GOSUB 470
380 PRINT "WHENEVER YOU MAKE
 A WAGER,  YOU MUST ENTER EX
ACTLY FOUR NUMBERS. FOR EXAM
PLE:"
390 PRINT
400 PRINT "   WAGER    INPUT
              -----    -----
                100     0100
"
410 PRINT "       50     0050
               2000     2000
                  1     0001
"
420 FOR I=1 TO 10
430 PRINT
440 NEXT I
450 INPUT "PRESS ENTER WHEN
READY TO   CONTINUE:":ANS$
460 RETURN
470 CALL CLEAR
480 PRINT TAB(9);"*** CRAPS
***"
490 PRINT
500 RETURN
510 REM
520 REM ***SETUP***
530 REM
540 CALL CLEAR
550 RANDOMIZE
560 DIM M1$(21)
570 CALL COLOR(10,5,5)
580 CALL COLOR(15,15,15)
590 CALL VCHAR(2,3,110,22)
600 CALL VCHAR(2,32,110,22)
610 CALL HCHAR(2,3,110,30)
620 CALL HCHAR(23,3,110,30)
630 CALL HCHAR(18,12,148,5)
640 CALL HCHAR(18,18,148,5)
650 CALL HCHAR(22,12,148,5)
660 CALL HCHAR(22,18,148,5)
670 CALL VCHAR(18,12,148,5)
680 CALL VCHAR(18,16,148,5)
690 CALL VCHAR(18,18,148,5)
700 CALL VCHAR(18,22,148,5)
710 A=1
720 B=5
730 C=0
740 D=0
750 GT=1500
760 MM$="PRESS ANY KEY TO RO
LL"
```

```
770 FOR I=1 TO 21
780 M1$(I)=SEG$(MM$,I,1)
790 NEXT I
800 RETURN
810 REM
820 REM ***PLAY***
830 REM
840 P=0
850 CALL HCHAR(12,5,32,23)
860 CALL HCHAR(14,5,32,23)
870 FOR I=1 TO 23
880 CALL HCHAR(12,4+I,ASC(SE
G$("YOU HAVE        DOLLARS"
,I,1)))
890 NEXT I
900 CALL HCHAR(16,13,32,8)
910 CALL HCHAR(6,5,32,25)
920 IF (A=0)*(B=0)*(C=0)THEN
 980
930 IF (A=0)*(B=0)THEN 970
940 IF A=0 THEN 960
950 CALL HCHAR(12,15,A+48)
960 CALL HCHAR(12,16,B+48)
970 CALL HCHAR(12,17,C+48)
980 CALL HCHAR(12,18,D+48)
990 FOR I=1 TO 23
1000 CALL HCHAR(14,4+I,ASC(S
EG$("HOW MUCH WILL YOU RISK?
",I,1)))
1010 NEXT I
1020 N1=0
1030 N2=0
1040 N3=0
1050 N4=0
1060 CALL KEY(0,N1,T)
1070 IF (T=0)+(T=-1)THEN 106
0
1080 N1=N1-48
1090 IF (N1<0)+(N1>9)THEN 11
10
1100 IF N1*1000<=A*1000 THEN
 1130
1110 GOSUB 2180
1120 GOTO 1060
1130 CALL HCHAR(16,15,N1+48)
1140 CALL KEY(0,N2,T)
1150 IF (T=0)+(T=-1)THEN 114
0
1160 N2=N2-48
1170 IF (N2<0)+(N2>9)THEN 11
90
```

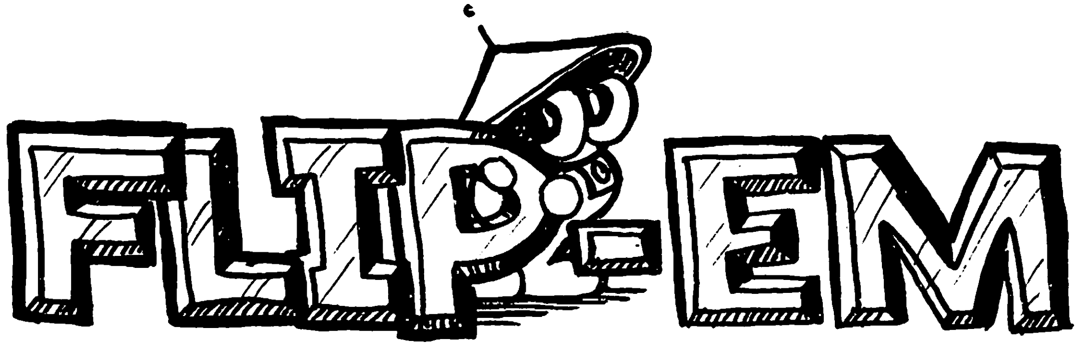


```
1180 IF N1*1000+N2*100<=A*10
00+B*100 THEN 1210
1190 GOSUB 2180
1200 GOTO 1140
1210 CALL HCHAR(16,16,N2+48)
1220 CALL KEY(0,N3,T)
1230 IF (T=0)+(T=-1)THEN 122
0
1240 N3=N3-48
1250 IF (N3<0)+(N3>9)THEN 12
70
1260 IF N1*1000+N2*100+N3*10
<=A*1000+B*100+C*10 THEN 129
0
1270 GOSUB 2180
1280 GOTO 1220
1290 CALL HCHAR(16,17,N3+48)
1300 CALL KEY(0,N4,T)
1310 IF (T=0)+(T=-1)THEN 130
0
1320 N4=N4-48
1330 IF (N4<0)+(N4>9)THEN 13
70
1340 IF (N1=0)*(N2=0)*(N3=0)
*(N4=0)THEN 2160
1350 IF N1*1000+N2*100+N3*10
+N4<=A*1000+B*100+C*10+D THE
N 1390
1360 GOTO 1390
1370 GOSUB 2180
1380 GOTO 1300
1390 CALL HCHAR(16,18,N4+48)
1400 REM
1410 REM ***ROLL DICE***
1420 REM
1430 FOR I=1 TO 21
1440 CALL HCHAR(10,4+I,ASC(M
1$(I)))
1450 NEXT I
1460 CALL KEY(0,AK,T)
1470 IF (T=0)+(T=-1)THEN 146
0
1480 CALL HCHAR(10,5,32,21)
1490 FOR Z=1 TO RND*4+2
1500 D1=INT(RND*6+1)
1510 FOR X=1 TO 3
1520 CALL HCHAR(18+X,13,32,3
)
1530 CALL HCHAR(18+X,19,32,3
)
1540 NEXT X
```

```
1550 CC=0
1560 ON D1 GOSUB 1960,1980,2
010,2040,2090,2120
1570 D2=INT(RND*6+1)
1580 CC=6
1590 ON D2 GOSUB 1960,1980,2
010,2040,2090,2120
1600 NEXT Z
1610 IF P<>0 THEN 1740
1620 IF (D1+D2=7)+(D1+D2=11)
THEN 1770
1630 IF (D1+D2=2)+(D1+D2=3)+
(D1+D2=12)THEN 2250
1640 P=D1+D2
1650 FOR I=1 TO 13
1660 CALL HCHAR(6,4+I,ASC(SE
G$("YOUR POINT IS",I,1)))
1670 NEXT I
1680 IF D1+D2<10 THEN 1720
1690 CALL HCHAR(6,19,49)
1700 CALL HCHAR(6,20,48)
1710 GOTO 1430
1720 CALL HCHAR(6,20,D1+D2+4
8)
1730 GOTO 1430
1740 IF D1+D2=7 THEN 2250
1750 IF P=D1+D2 THEN 1770
1760 GOTO 1430
1770 GT=GT+N1*1000+N2*100+N3
*10+N4
1780 CALL HCHAR(6,5,32,25)
1790 FOR I=1 TO 12
1800 CALL HCHAR(6,4+I,ASC(SE
G$("GOOD JOB!!!!",I,1)))
1810 NEXT I
1820 CALL SOUND(400,330,6)
1830 CALL SOUND(1200,440,4)
1840 IF GT<10000 THEN 1900
1850 A=9
1860 B=9
1870 C=9
1880 D=9
1890 GOTO 1940
1900 A=INT(GT/1000)
1910 B=INT((GT-A*1000)/100)
1920 C=INT((GT-A*1000-B*100)
/10)
1930 D=GT-A*1000-B*100-C*10
1940 CALL HCHAR(8,5,32,18)
1950 GOTO 840
1960 CALL HCHAR(20,14+CC,42)
```

```
1970 RETURN
1980 CALL HCHAR(19,14+CC,42)
1990 CALL HCHAR(21,14+CC,42)
2000 RETURN
2010 GOSUB 1960
2020 GOSUB 1980
2030 RETURN
2040 CALL HCHAR(19,13+CC,42)
2050 CALL HCHAR(19,15+CC,42)
2060 CALL HCHAR(21,13+CC,42)
2070 CALL HCHAR(21,15+CC,42)
2080 RETURN
2090 GOSUB 1960
2100 GOSUB 2040
2110 RETURN
2120 GOSUB 2040
2130 CALL HCHAR(20,13+CC,42)
2140 CALL HCHAR(20,15+CC,42)
2150 RETURN
2160 YW=2
2170 GOTO 2350
2180 FOR I=1 TO 10
2190 CALL HCHAR(8,10+I,ASC(S
EG$("TRY AGAIN!",I,1)))
2200 NEXT I
2210 FOR I=1 TO 100
2220 NEXT I
2230 CALL HCHAR(8,11,32,10)
2240 RETURN
2250 CALL HCHAR(6,5,32,25)
2260 FOR I=1 TO 23
2270 CALL HCHAR(6,4+I,ASC(SE
G$("YOU CRAPPED OUT, SORRY!"
,I,1)))
2280 NEXT I
2290 FOR I=1 TO 100
2300 NEXT I
2310 GT=GT-(N1*1000+N2*100+N
3*10+N4)
2320 YW=1
2330 IF GT=0 THEN 2350
2340 GOTO 1900
2350 RETURN
2360 REM
2370 REM ***END***
2380 REM
2390 A=INT(GT/1000)
2400 CALL CLEAR
2410 PRINT "YOUR GRAND TOTAL
 IS";GT;"DOLLARS...."
2420 IF YW=2 THEN 2460
```

```
2430 PRINT
2440 PRINT "I GUESS THE DICE
 GODS WERE  ANGRY. MAYBE NEX
T TIME      YOU'LL HAVE BETT
ER LUCK."
2450 GOTO 2480
2460 IF A=0 THEN 2710
2470 ON A GOSUB 2530,2550,25
70
2480 FOR I=1 TO 12
2490 PRINT
2500 NEXT I
2510 INPUT "WOULD YOU LIKE T
O PLAY ONE  MORE TIME (Y/N)?
 ":ANS$
2520 IF ANS$="Y" THEN 70 ELS
E 2730
2530 PRINT "NOT TOO SHABBY."
2540 RETURN
2550 PRINT "I LIKE YOUR STYL
E!"
2560 RETURN
2570 PRINT "GOOD PLAYING!!"
2580 RETURN
2590 PRINT "MY, AREN'T WE TH
E HOT SHOT!"
2600 RETURN
2610 PRINT "I'M GENUINELY IM
PRESSED!!!"
2620 RETURN
2630 PRINT "THE DICE GODS AR
E SMILING ON YOU!"
2640 RETURN
2650 PRINT "GREAT EFFORT!  M
AYBE YOU    SHOULD PLAY DICE
 FOR REAL!"
2660 RETURN
2670 PRINT "FANTASTIC!!!  LA
S VEGAS     TREMBLES WHEN TH
EY HEAR YOURNAME!"
2680 RETURN
2690 PRINT "INCREDIBLE!!!  A
T THIS RATE YOU COULD RETIRE
 ON YOUR    WINNINGS!"
2700 RETURN
2710 PRINT "AT LEAST YOU QUI
T BEFORE YOUWENT BROKE!"
2720 GOTO 2480
2730 RETURN
```



Flip-em is a strategy game for two. It implements some of the best qualities of chess and bridge, and virtually anyone can play! The rules are quite simple. To move, just enter the coordinates of the square you wish to occupy. There are two restrictions: One, you must move to a square directly adjacent to a square occupied by the opponent. Two, you must be able (as a direct result of the move) to flip at least one of the opponent's men. The way that you earn the right to flip is this: If the tile you play, in partnership with another one of your tiles, encloses any opposing pieces in any row (vertically, horizontally, or diagonally), you flip them over to your color. This is done by entering the coordinates of the piece (or pieces) to be flipped. The example below should help you to understand.

		C	O	L	U	M	N	
	1	2	3	4	5	6	7	8
	1–	–	–	–	–	–	–	–
	2–	–	–	–	–	–	–	–
R	3–	Y	O	Y	Y	O	–	–
O	4–	–	Y	O	O	O	–	–
W	5–	O	O	Y	O	O	O	–
	6–	–	O	Y	Y	Y	Y	–
	7–	Y	–	Y	–	–	–	–
	8–	–	–	–	–	–	–	–

The upper left-hand corner is 1,1 and the lower left-hand corner is 8,1 (and so on). In the above game, (Y=You O=Opponent) assume that it is your turn (Y). The possible moves are: 2,2; 2,3; 2,6; 2,7; 3,7; 4,1; 4,2; 4,7; 4,8; 5,1; 5,8; 6,1; 6,2; 6,8 and 7,3. All of these moves satisfy the two basic requirements outlined above. If Y's choice was 4,7, then the following tiles would be flipped: 4,4; 4,5; 4,6 (enclosed by 4,7 and 4,3) 5,6 (enclosed by 6,5 and 4,7) and 5,7 (enclosed by 4,7 and 6,7). Assuming that it is O's turn, the possible moves are: 2,3; 2,4; 2,5; 2,6; 3,1; 4,2; 6,8; 7,3; 7,5; 7,6; 7,7; 7,8; 8,1; 8,3; 8,4 and 8,5. In terms of strategy, the four corners (1,1; 1,8; 8,1 and 8,8) are critically important. Therefore, if O chooses 8,1 then 7,2 gets flipped (enclosed by 8,1 and 6,3). This game is fun for all ages. Enjoy!

```
10 REM ********************
20 REM ***              ***
30 REM ***   FLIP-EM    ***
40 REM ***              ***
50 REM ********************
60 GOSUB 110
70 GOSUB 820
80 GOSUB 1400
90 GOSUB 3450
100 END
110 REM
120 REM ***INSTRUCTIONS***
130 REM
140 CALL SCREEN(14)
150 GOSUB 780
160 PRINT "THIS IS THE GAME
OF FLIP-EM.YOU MAY KNOW IT B
ETTER AS   'OTHELLO'."
170 PRINT
180 PRINT "THE OBJECT IS TO
HAVE MORE  OF YOUR MEN ON TH
E BOARD    THAN YOUR OPPONEN
T DOES AT"
190 PRINT "THE END OF THE GA
ME."
200 PRINT
210 PRINT "THE GAME ENDS WHE
N ALL 64   SQUARES ARE FILLE
D."
220 FOR I=1 TO 9
230 PRINT
240 NEXT I
250 GOSUB 760
260 GOSUB 780
270 PRINT "SOME SIMPLE RULES
:"
280 PRINT
290 PRINT "AFTER EACH MOVE,
YOU MUST BEABLE TO FLIP AN O
PPONENT'S  PIECE. IF YOU CAN
'T, YOU"
300 PRINT "MUST SKIP YOUR TU
RN. THIS ISDONE BY PRESSING
P)ASS."
310 PRINT
320 PRINT "AFTER EACH MOVE,
IF FLIPS   ARE POSSIBLE, YOU
 WILL FLIP ALL OF YOUR OPPON
ENT'S"
330 PRINT "PIECES BETWEEN YO
UR OWN AND THE ONE YOU JUST
```

```
PLACED ON  THE BOARD."
340 FOR I=1 TO 6
350 PRINT
360 NEXT I
370 GOSUB 760
380 GOSUB 780
390 PRINT TAB(11);"EXAMPLE:"
400 PRINT
410 PRINT "  0 X X     0 X X
     0 X X"
420 PRINT
430 PRINT "  0 X    => 0 X 0
  => 0 0 0"
440 PRINT
450 PRINT "  X 0 X     X 0 X
     X 0 X"
460 PRINT
470 PRINT TAB(14);"OR"
480 PRINT
490 PRINT "  0 X       0 X 0
     0 0 0"
500 PRINT
510 PRINT "  0 X X  => 0 X X
  => 0 0 0"
520 PRINT
530 PRINT "  0 0 0     0 0 0
     0 0 0"
540 FOR I=1 TO 5
550 PRINT
560 NEXT I
570 GOSUB 760
580 GOSUB 780
590 PRINT "SHOULD YOU MAKE A
 MISTAKE   AND DO SOMETHING
ON THE     WRONG SQUARE, IT
CAN BE"
600 PRINT "CORRECTED BY PRES
SING       C)HANGE. YOU WILL
 THEN BE   ASKED THE ROW, CO
LOMN AND   DESIRED COLOR."
610 PRINT
620 PRINT "IF YOU SHOULD WIS
H TO MOVE  AFTER A FLIP, PRE
SS M)OVE   AND IT WILL BE YO
UR MOVE    AGAIN."
630 PRINT
640 PRINT "WHEN YOU ARE FINI
SHED       FLIPPING, PRESS D
)ONE AND ITWILL BE YOUR OPPO
NENTS TURN."
650 FOR I=1 TO 4
```

```
660 PRINT
670 NEXT I
680 GOSUB 760
690 GOSUB 780
700 PRINT "YOU MUST ALWAYS M
OVE NEXT TOANOTHER PIECE OR
THE MOVE   WON'T COUNT!!!!"
710 PRINT
720 PRINT "ENJOY!!!"
730 FOR I=1 TO 15
740 PRINT
750 NEXT I
760 INPUT "PRESS ENTER WHEN
READY TO   CONTINUE: ":ANS$
770 RETURN
780 CALL CLEAR
790 PRINT TAB(8);"*** FLIP-E
M ***"
800 PRINT
810 RETURN
820 REM
830 REM ***SETUP***
840 REM
850 CALL CLEAR
860 CALL SCREEN(1)
870 DIM A$(6),ROW(8),COL(8)
880 CALL CHAR(128,"FFFFFFFFF
FFFFFFF")
890 CALL CHAR(129,"FFFFFFFFF
FFFFFFF")
900 CALL CHAR(130,"FFFFFFFFF
FFFFFFF")
910 CALL CHAR(131,"FFFFFFFFF
FFFFFFF")
920 CALL CHAR(136,"031F3F7F7
F7FFFFF")
930 CALL CHAR(137,"C0F8FCFEF
EFEFFFF")
940 CALL CHAR(138,"FFFF7F7F7
F3F1F03")
950 CALL CHAR(139,"FFFFFEFEF
EFCF8C0")
960 CALL CHAR(144,"031F3F7F7
F7FFFFF")
970 CALL CHAR(145,"C0F8FCFEF
EFEFFFF")
980 CALL CHAR(147,"FFFFFEFEF
EFCF8C0")
990 CALL CHAR(146,"FFFF7F7F7
F3F1F03")
1000 CALL COLOR(13,3,1)
```

```
1010 CALL COLOR(14,16,1)
1020 CALL COLOR(15,11,1)
1030 FOR I=2 TO 8
1040 CALL COLOR(I,16,1)
1050 NEXT I
1060 FOR I=1 TO 24 STEP 3
1070 FOR J=1 TO 24 STEP 3
1080 CALL HCHAR(I,J+3,128,2)
1090 CALL HCHAR(I+1,J+3,128,
2)
1100 NEXT J
1110 NEXT I
1120 FOR I=1 TO 28
1130 CALL HCHAR(24,3+I,ASC(S
EG$("C)HANGE  D)ONE  M)OVE  
P)ASS",I,1)))
1140 NEXT I
1150 FOR I=1 TO 5
1160 CALL HCHAR(5,27+I,ASC(S
EG$("MOVE:",I,1)))
1170 NEXT I
1180 FOR I=1 TO 3
1190 CALL HCHAR(11,27+I,ASC(
SEG$("R,C",I,1)))
1200 NEXT I
1210 FOR I=1 TO 8
1220 ROW(I)=(I-1)*3+1
1230 COL(I)=I*3+1
1240 NEXT I
1250 R=5
1260 C=5
1270 CLR=144
1280 GOSUB 2080
1290 R=4
1300 C=4
1310 GOSUB 2080
1320 R=5
1330 C=4
1340 CLR=136
1350 GOSUB 2080
1360 R=4
1370 C=5
1380 GOSUB 2080
1390 RETURN
1400 REM
1410 REM ***PLAY***
1420 REM
1430 MV=0
1440 CLR=144
1450 FOR M=5 TO 64
1460 M$="  "&STR$(M)
```

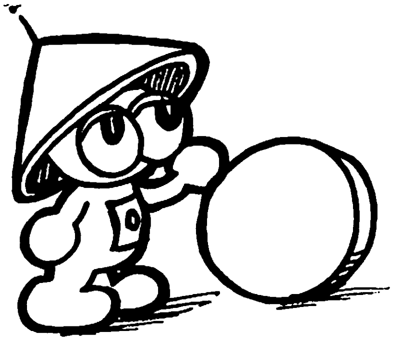
```
1470 CALL HCHAR(1,29,ASC(SEG
$(M$,LEN(M$)-1,1)))
1480 CALL HCHAR(1,30,ASC(SEG
$(M$,LEN(M$),1)))
1490 IF ((MV=1)*(CLR=144))+(
(MV=0)*(CLR=136))THEN 1570
1500 CALL HCHAR(7,27,32,6)
1510 FOR I=1 TO 5
1520 CALL HCHAR(7,26+I,ASC(S
EG$("WHITE",I,1)))
1530 CLR=136
1540 MV=0
1550 NEXT I
1560 GOTO 1630
1570 CALL HCHAR(7,27,32,6)
1580 FOR I=1 TO 6
1590 CALL HCHAR(7,26+I,ASC(S
EG$("ORANGE",I,1)))
1600 CLR=144
1610 MV=0
1620 NEXT I
1630 GOSUB 3250
1640 IF ECK=0 THEN 1680
1650 IF KEY=80 THEN 1490
1660 GOSUB 2430
1670 GOTO 1630
1680 CALL GCHAR(ROW(R),COL(C
),CK)
1690 IF CK=128 THEN 1720
1700 GOSUB 2540
1710 GOTO 1630
1720 GOSUB 2870
1730 IF BMCK=0 THEN 1760
1740 GOSUB 2760
1750 GOTO 1630
1760 GOSUB 2080
1770 CALL HCHAR(7,27,32,6)
1780 FOR I=1 TO 6
1790 CALL HCHAR(7,26+I,ASC(S
EG$(" FLIP ",I,1)))
1800 NEXT I
1810 GOSUB 3250
1820 IF ECK=0 THEN 1960
1830 IF KEY=67 THEN 1880
1840 IF KEY=68 THEN 2060
1850 IF KEY=77 THEN 2050
1860 GOSUB 2430
1870 GOTO 1810
1880 CALL HCHAR(7,27,32,6)
1890 FOR I=1 TO 6
1900 CALL HCHAR(7,26+I,ASC(S
EG$("CHANGE",I,1)))
1910 NEXT I
1920 GOSUB 3250
1930 IF ECK=1 THEN 1770
1940 GOSUB 3010
1950 GOTO 1770
1960 CALL GCHAR(ROW(R),COL(C
),CK)
1970 IF CK<>CLR THEN 2000
1980 GOSUB 2320
1990 GOTO 1810
2000 IF (CK=136)+(CK=144)THE
N 2030
2010 GOSUB 2650
2020 GOTO 1810
2030 GOSUB 2080
2040 GOTO 1810
2050 MV=1
2060 NEXT M
2070 RETURN
2080 REM ***PIECE***
2090 REM
2100 CALL HCHAR(ROW(R),COL(C
),CLR)
2110 CALL HCHAR(ROW(R),COL(C
)+1,CLR+1)
2120 CALL HCHAR(ROW(R)+1,COL
(C),CLR+2)
2130 CALL HCHAR(ROW(R)+1,COL
(C)+1,CLR+3)
2140 RETURN
2150 CALL SOUND(-1000,110,2)
2160 FOR I=1 TO 6
2170 CALL HCHAR(16+I,27,ASC(
SEG$(A$(I),1,1)))
2180 CALL HCHAR(16+I,28,ASC(
SEG$(A$(I),2,1)))
2190 CALL HCHAR(16+I,29,ASC(
SEG$(A$(I),3,1)))
2200 CALL HCHAR(16+I,30,ASC(
SEG$(A$(I),4,1)))
2210 CALL HCHAR(16+I,31,ASC(
SEG$(A$(I),5,1)))
2220 CALL HCHAR(16+I,32,ASC(
SEG$(A$(I),6,1)))
2230 NEXT I
2240 RETURN
2250 FOR I=1 TO 200
2260 NEXT I
2270 FOR I=17 TO 22
2280 CALL HCHAR(I,27,32,6)
```

```
2290 NEXT I
2300 CALL HCHAR(12,28,32,3)
2310 RETURN
2320 REM ***OWN PIECE ERROR*
**
2330 REM
2340 A$(1)="THAT  "
2350 A$(2)="IS    "
2360 A$(3)="YOUR  "
2370 A$(4)="OWN   "
2380 A$(5)="PIECE "
2390 A$(6)="      "
2400 GOSUB 2150
2410 GOSUB 2250
2420 RETURN
2430 REM ***RANGE ERROR***
2440 REM
2450 A$(1)=" MUST "
2460 A$(2)="  BE  "
2470 A$(3)="1 TO 8"
2480 A$(4)="      "
2490 A$(5)="      "
2500 A$(6)="      "
2510 GOSUB 2150
2520 GOSUB 2250
2530 RETURN
2540 REM ***SPACE TAKEN ERRO
R***
2550 REM
2560 A$(1)="THAT  "
2570 A$(2)="SPACE "
2580 A$(3)="IS    "
2590 A$(4)="TAKEN "
2600 A$(5)="TRY   "
2610 A$(6)="AGAIN "
2620 GOSUB 2150
2630 GOSUB 2250
2640 RETURN
2650 REM ***SPACE EMPTY ERRO
R***
2660 REM
2670 A$(1)="THAT  "
2680 A$(2)="SPACE "
2690 A$(3)="IS    "
2700 A$(4)="EMPTY "
2710 A$(5)="TRY   "
2720 A$(6)="AGAIN "
2730 GOSUB 2150
2740 GOSUB 2250
2750 RETURN
2760 REM ***BAD MOVE ERROR**
*
```

```
2770 REM
2780 A$(1)="BAD   "
2790 A$(2)="MOVE  "
2800 A$(3)="TRY   "
2810 A$(4)="AGAIN "
2820 A$(5)="      "
2830 A$(6)="      "
2840 GOSUB 2150
2850 GOSUB 2250
2860 RETURN
2870 REM ***MOVE CHECK***
2880 REM
2890 C1=0
2900 FOR I=(R<>1)TO (R=8)+1
2910 FOR J=(C<>1)TO (C=8)+1
2920 IF (I=0)*(J=0)THEN 2970
2930 C1=C1+1
2940 CALL GCHAR(ROW(R+I),COL
(C+J),CK)
2950 IF (CK=136)+(CK=144)THE
N 2970
2960 C1=C1-1
2970 NEXT J
2980 NEXT I
2990 BMCK=(C1<>0)+1
3000 RETURN
3010 REM ***CHANGE***
3020 REM
3030 AUX=CLR
3040 A$(1)="  TO  "
3050 A$(2)=" WHAT "
3060 A$(3)="      "
3070 A$(4)="G)REEN"
3080 A$(5)="W)HITE"
3090 A$(6)="O)RNGE"
3100 GOSUB 2160
3110 CALL KEY(3,KEY,STAT)
3120 IF STAT=0 THEN 3110
3130 IF (KEY=71)+(KEY=79)+(K
EY=87)THEN 3140 ELSE 3110
3140 IF KEY<>71 THEN 3170
3150 CLR=128
3160 GOTO 3210
3170 IF KEY<>79 THEN 3200
3180 CLR=144
3190 GOTO 3210
3200 CLR=136
3210 GOSUB 2080
3220 GOSUB 2270
3230 CLR=AUX
3240 RETURN
```

```
3250 REM ***INPUT***
3260 REM
3270 ECK=0
3280 CALL HCHAR(12,28,32,3)
3290 CALL KEY(3,KEY,STAT)
3300 IF STAT<>1 THEN 3290
3310 R=KEY-48
3320 IF (R<1)+(R>8)THEN 3430
3330 CALL HCHAR(12,28,KEY)
3340 CALL HCHAR(12,29,44)
3350 CALL KEY(3,KEY,STAT)
3360 IF STAT<>1 THEN 3350
3370 C=KEY-48
3380 IF (C<1)+(C>8)THEN 3410
3390 CALL HCHAR(12,30,KEY)
3400 GOTO 3440
3410 GOSUB 2430
3420 GOTO 3250
3430 ECK=1
3440 RETURN
3450 REM
3460 REM ***END***
3470 REM
3480 CALL SCREEN(14)
3490 FOR I=1 TO 8
3500 FOR J=4 TO 25 STEP 3
3510 CALL GCHAR(ROW(I),J,CK)
3520 IF CK<>136 THEN 3550
3530 WH=WH+1
3540 GOTO 3570
3550 IF CK<>144 THEN 3570
3560 OR=OR+1
3570 NEXT J
3580 NEXT I
3590 CALL SCREEN(1)
3600 CALL CLEAR
3610 IF WH<=OR THEN 3650
3620 PRINT TAB(5);"WHITE IS
THE WINNER!!!"
3630 PRINT TAB(10);WH;"TO";O
R
3640 GOTO 3730
3650 IF OR<=WH THEN 3690
3660 PRINT TAB(5);"ORANGE IS
 THE WINNER!!!"
3670 PRINT TAB(10);OR;"TO";W
H
3680 GOTO 3730
3690 PRINT TAB(10);"INCREDIB
LE!!"
3700 PRINT TAB(4);"YOUR SCOR
```

```
ES ARE EQUAL!!!"
3710 PRINT TAB(8);"IT'S A TI
E GAME"
3720 PRINT TAB(10);OR;"TO";W
H
3730 FOR I=1 TO 10
3740 PRINT
3750 NEXT I
3760 PRINT "I HOPE YOU ENJOY
ED THE GAME!"
3770 RETURN
```



3-D TIC TAC TOE

Many of you know that standard tic-tac-toe has very few points of strategy. What's more, a thorough mastery of the game can be achieved in 10 minutes! Three-Dimensional Tic-Tac-Toe is only a distant relative of the simple game. In 3-D Tic-Tac-Toe, the object is to make a chain of four consecutive pieces, on any one plane or on all four planes, while blocking your opponent's attempt to do the same. The real trick is to create a winning situation a few turns before the four-in-a-row is completed; but few can ever hope to gain this level of sophistication.

The program, though it doesn't use color, uses easy-to-understand graphics. Lines 1030-1090 draw the original configuration of four 4x4 gameboards. At line 1040 you see, in the parentheses, the number 35. This is the ASCII value for a pound sign (#). Calculate I+48 to better understand 1050. Line 1010 writes, "ENTER (12 blank spaces) (1,2,3,4)." The blank spaces allow for the words "gameboard," "row," and "column" to be written before "(1,2,3,4)."

```
10 REM ********************
20 REM ** 3-D TIC-TAC-TOE**
30 REM **                **
40 REM ** BY SCOTT SINGER**
50 REM ********************
60 GOSUB 110
70 GOSUB 780
80 GOSUB 1110
90 GOSUB 2160
100 END
110 REM
120 REM ***INSTRUCTIONS***
130 REM
140 GOSUB 740
150 PRINT "THE OBJECT OF THE
 GAME IS TOGET FOUR IN A ROW
 - ON A    SINGLE PLANE OR O
N ALL FOUR AT ONCE."
160 PRINT
170 PRINT "DON'T GET CONFUSE
D, A SERIESOF EXAMPLES WILL
FOLLOW."
180 PRINT
190 PRINT "TO MOVE, ENTER TH
E NUMBER OFTHE GAMEBOARD (1,
2,3 OR 4)  AND THE POSITION
YOU WOULD"
200 PRINT "LIKE TO OCCUPY (R
OW,COLUMN)."
210 FOR I=1 TO 8
220 PRINT
230 NEXT I
240 GOSUB 720
250 GOSUB 740
260 PRINT "FOR EXAMPLE: 2,1,
3 INSTRUCTSTHE COMPUTER TO P
LACE YOUR  MARK ON THE NUMBE
R 2 GAME-"
270 PRINT "BOARD AT ROW 1 CO
LUMN 3.    THE EXAMPLE BELOW
 SHOULD    HELP."
280 PRINT
290 PRINT "  #1     #2     #
3     #4"
300 PRINT
310 PRINT " X--O   XOO-   X-
-O   ----   ----   -X--   OO
--   -X-X   O---   --O-   --
X-   OOOX"
320 PRINT " X-X-   -O--   --
-X   --OX"
```

```
330 PRINT
340 PRINT "WOULD YOU BELIEVE
 THAT X     ALREADY WON? HERE
'S HOW..."
350 PRINT
360 PRINT "THE FOUR-IN-A-ROW
 ARE:        1,1,1  2,2,2  3,3
,3  4,4,4."
370 PRINT
380 GOSUB 720
390 GOSUB 740
400 PRINT "TO SEE IF YOU'VE
WON, THINK OF ALL FOUR GAMEB
OARDS ONE  ON TOP OF ANOTHER
.  IGNORING"
410 PRINT "THE FACT THAT ONE
 SPACE MAY HAVE BOTH AN X AN
D AN O ON  IT (USING THIS ME
THOD),"
420 PRINT "CHECK TO SEE IF T
HERE ARE   FOUR IN A ROW."
430 PRINT
440 PRINT "THERE IS ONE LIMI
TATION, YOUMAY NOT MOVE TO A
N 0CCUPIED SPACE."
450 FOR I=1 TO 8
460 PRINT
470 NEXT I
480 GOSUB 720
490 GOSUB 740
500 PRINT "THE FOLLOWING ARE
 WINNING   COMBINATIONS:"
510 PRINT
520 PRINT "1,1,1 1,2,2 1,3,3
 1,4,4;     1,2,2 2,2,2 3,2,2
 4,2,2;     1,1,4 2,1,3 3,1,2
 4,1,1;"
530 PRINT
540 PRINT "AND SO ON."
550 PRINT
560 PRINT "YOU ARE ASKED TO
ENTER THREENUMBERS EACH TURN
: # OF      GAMEBOARD,ROW,COL
UMN."
570 FOR I=1 TO 8
580 PRINT
590 NEXT I
600 GOSUB 720
610 GOSUB 740
620 PRINT "TO SIGNIFY A CAT
S GAME,    ENTER A 'C' (FOR
THE          GAMEBOARD NUMBER)
."
630 PRINT
640 PRINT "IF YOU ACCIDENTAL
LY INPUT   THE WRONG NUMBER/
S, TYPE `R'FOR REDO."
650 PRINT
660 PRINT "PLAYER #1 IS ALWA
YS 'O' AND PLAYER #2 IS ALWA
YS 'X'."
670 PRINT
680 PRINT "THE PROGRAM WILL
CALCULATE  ALL WINNING MOVES
. HAVE FUN!"
690 FOR I=1 TO 7
700 PRINT
710 NEXT I
720 INPUT "PRESS ENTER WHEN
READY TO   CONTINUE: ":ANS$
730 RETURN
740 CALL CLEAR
750 PRINT TAB(6);"*** TIC-TA
C-TOE ***"
760 PRINT
770 RETURN
780 REM
790 REM ***SETUP***
800 REM
810 CALL CLEAR
820 DIM GB(4,4,4),G(9),C(6),
R(3)
830 FOR X=1 TO 4
840 FOR Y=1 TO 4
850 FOR Z=1 TO 4
860 GB(X,Y,Z)=45
870 NEXT Z
880 NEXT Y
890 NEXT X
900 FOR I=1 TO 9
910 G(I)=ASC(SEG$("GAMEBOARD
",I,1))
920 NEXT I
930 FOR I=1 TO 6
940 C(I)=ASC(SEG$("COLUMN",I
,1))
950 NEXT I
960 FOR I=1 TO 3
970 R(I)=ASC(SEG$("ROW",I,1)
)
980 NEXT I
990 PRINT TAB(3);"PLAYER #1"
```

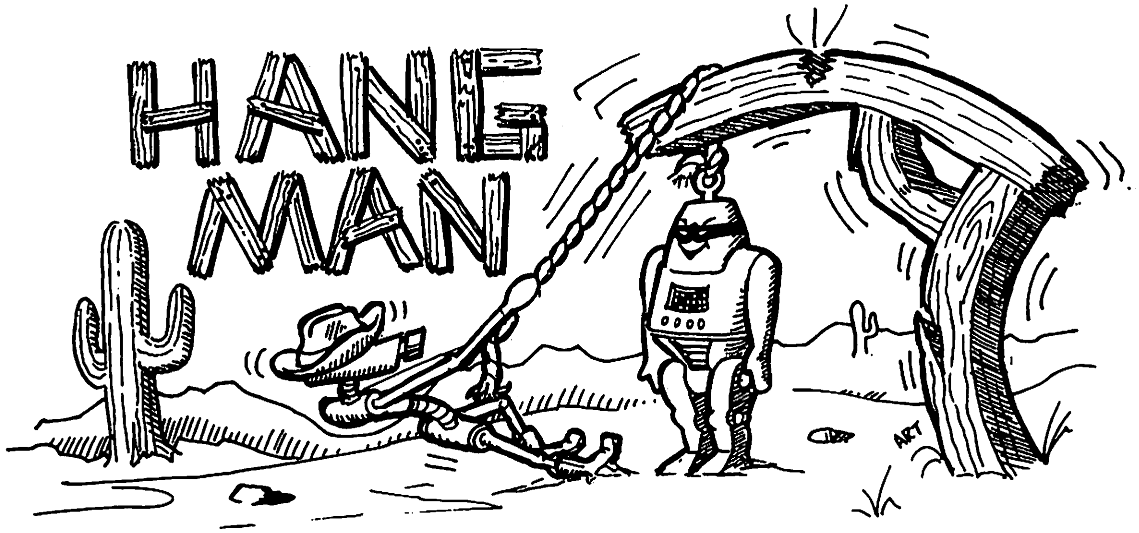
```
1000 PRINT
1010 PRINT TAB(3);"ENTER";TA
B(19);"(1,2,3,4)"
1020 PRINT
1030 FOR I=1 TO 4
1040 CALL HCHAR(7,7*I-1,35)
1050 CALL HCHAR(7,7*I,I+48)
1060 FOR X=1 TO 4
1070 CALL HCHAR(8+X,7*I-2,45
,4)
1080 NEXT X
1090 NEXT I
1100 RETURN
1110 REM
1120 REM ***PLAY***
1130 REM
1140 FOR P=1 TO 2
1150 CALL HCHAR(20,13,P+48)
1160 CALL HCHAR(20,18,32,5)
1170 FOR I=1 TO 9
1180 CALL HCHAR(22,10+I,G(I)
)
1190 NEXT I
1200 GOSUB 1950
1210 IF N=67 THEN 2150
1220 IF N=82 THEN 1160
1230 CALL HCHAR(20,18,N)
1240 CALL HCHAR(20,19,44)
1250 N1=N-48
1260 FOR I=1 TO 3
1270 CALL HCHAR(22,10+I,R(I)
)
1280 NEXT I
1290 GOSUB 1950
1300 IF N=67 THEN 1260
1310 IF N=82 THEN 1160
1320 CALL HCHAR(20,20,N)
1330 CALL HCHAR(20,21,44)
1340 N2=N-48
1350 FOR I=1 TO 6
1360 CALL HCHAR(22,10+I,C(I)
)
1370 NEXT I
1380 GOSUB 1950
1390 IF N=67 THEN 1350
1400 IF N=82 THEN 1160
1410 CALL HCHAR(20,22,N)
1420 N3=N-48
1430 IF GB(N1,N2,N3)<>45 THE
N 2020
1440 CALL HCHAR(8+N2,7*N1+N3
-3,70+P*9)
```

```
1450 GB(N1,N2,N3)=P*9+70
1460 FOR A=1 TO 4
1470 FOR B=1 TO 4
1480 IF GB(A,B,1)=45 THEN 15
00
1490 IF ((GB(A,B,1)=GB(A,B,2
))*(GB(A,B,1)=GB(A,B,3))*(GB
(A,B,1)=GB(A,B,4)))THEN 2090
1500 IF GB(A,1,B)=45 THEN 15
20
1510 IF ((GB(A,1,B)=GB(A,2,B
))*(GB(A,2,B)=GB(A,3,B))*(GB
(A,3,B)=GB(A,4,B)))THEN 2090
1520 IF GB(1,A,B)=45 THEN 15
40
1530 IF ((GB(1,A,B)=GB(2,A,B
))*(GB(2,A,B)=GB(3,A,B))*(GB
(3,A,B)=GB(4,A,B)))THEN 2090
1540 NEXT B
1550 IF GB(A,1,1)=45 THEN 15
70
1560 IF ((GB(A,1,1)=GB(A,2,2
))*(GB(A,2,2)=GB(A,3,3))*(GB
(A,3,3)=GB(A,4,4)))THEN 2090
1570 IF GB(A,4,1)=45 THEN 15
90
1580 IF ((GB(A,4,1)=GB(A,3,2
))*(GB(A,3,2)=GB(A,2,3))*(GB
(A,2,3)=GB(A,1,4)))THEN 2090
1590 NEXT A
1600 IF GB(1,1,1)=45 THEN 16
40
1610 IF ((GB(1,1,1)=GB(2,2,2
))*(GB(2,2,2)=GB(3,3,3))*(GB
(3,3,3)=GB(4,4,4)))THEN 2090
1620 IF ((GB(1,1,1)=GB(2,1,2
))*(GB(2,1,2)=GB(3,1,3))*(GB
(3,1,3)=GB(4,1,4)))THEN 2090
1630 IF ((GB(1,1,1)=GB(2,2,1
))*(GB(2,2,1)=GB(3,3,1))*(GB
(3,3,1)=GB(4,4,1)))THEN 2090
1640 IF GB(1,4,1)=45 THEN 16
80
1650 IF ((GB(1,4,1)=GB(2,3,2
))*(GB(2,3,2)=GB(3,2,3))*(GB
(4,1,4)=GB(3,2,3)))THEN 2090
1660 IF ((GB(1,4,1)=GB(2,3,1
))*(GB(2,3,1)=GB(3,2,1))*(GB
(3,2,1)=GB(4,1,1)))THEN 2090
1670 IF ((GB(1,4,1)=GB(2,4,2
))*(GB(2,4,2)=GB(3,4,3))*(GB
```

```
(3,4,3)=GB(4,4,4)))THEN 2090
1680 IF GB(1,2,1)=45 THEN 17
00
1690 IF ((GB(1,2,1)=GB(2,2,2
))*(GB(2,2,2)=GB(3,2,3))*(GB
(3,2,3)=GB(4,2,4)))THEN 2090
1700 IF GB(1,3,1)=45 THEN 17
30
1710 IF ((GB(1,3,1)=GB(2,3,2
))*(GB(2,3,2)=GB(3,3,3))*(GB
(4,3,4)=GB(3,3,3)))THEN 2090
1720 IF (N=67)+(N=87)THEN 17
40
1730 IF GB(1,1,2)=45 THEN 17
50
1740 IF ((GB(1,1,2)=GB(2,2,2
))*(GB(2,2,2)=GB(3,3,2))*(GB
(3,3,2)=GB(4,4,2)))THEN 2090
1750 IF GB(1,4,2)=45 THEN 17
90
1760 IF ((GB(1,4,2)=GB(2,3,2
))*(GB(2,3,2)=GB(3,2,2))*(GB
(3,2,2)=GB(4,1,2)))THEN 2090
1770 IF GB(1,1,3)=45 THEN 17
90
1780 IF ((GB(1,1,3)=GB(2,2,3
))*(GB(3,3,3)=GB(2,2,3))*(GB
(3,3,3)=GB(4,4,3)))THEN 2090
1790 IF GB(1,4,3)=45 THEN 18
10
1800 IF ((GB(1,4,3)=GB(2,3,3
))*(GB(2,3,3)=GB(3,2,3))*(GB
(4,1,3)=GB(3,2,3)))THEN 2090
1810 IF GB(1,1,4)=45 THEN 18
50
1820 IF ((GB(1,1,4)=GB(2,2,4
))*(GB(2,2,4)=GB(3,3,4))*(GB
(3,3,4)=GB(4,4,4)))THEN 2090
1830 IF ((GB(1,1,4)=GB(2,2,3
))*(GB(2,2,3)=GB(3,3,2))*(GB
(4,4,1)=GB(3,3,2)))THEN 2090
1840 IF ((GB(1,1,4)=GB(2,1,3
))*(GB(2,1,3)=GB(3,1,2))*(GB
(3,1,2)=GB(4,1,1)))THEN 2090
1850 IF GB(1,4,4)=45 THEN 18
90
1860 IF ((GB(1,4,4)=GB(2,3,4
))*(GB(2,3,4)=GB(3,2,4))*(GB
(3,2,4)=GB(4,1,4)))THEN 2090
1870 IF ((GB(1,4,4)=GB(2,3,3
))*(GB(2,3,3)=GB(3,2,2))*(GB
```

```
(3,2,2)=GB(4,1,1)))THEN 2090
1880 IF ((GB(1,4,4)=GB(2,4,3
))*(GB(2,4,3)=GB(3,4,2))*(GB
(3,4,2)=GB(4,4,1)))THEN 2090
1890 IF GB(1,2,4)=45 THEN 19
10
1900 IF ((GB(1,2,4)=GB(2,2,3
))*(GB(2,2,3)=GB(3,2,2))*(GB
(3,2,2)=GB(4,2,1)))THEN 2090
1910 IF GB(1,3,4)=45 THEN 19
30
1920 IF ((GB(1,3,4)=GB(2,3,3
))*(GB(2,3,3)=GB(3,3,2))*(GB
(3,3,2)=GB(4,3,1)))THEN 2090
1930 NEXT P
1940 GOTO 1140
1950 CALL KEY(0,N,T)
1960 IF T<>1 THEN 1950
1970 IF N=82 THEN 2000
1980 IF N=67 THEN 2000
1990 IF (N<49)+(N>52)THEN 19
50
2000 CALL HCHAR(22,11,32,9)
2010 RETURN
2020 FOR I=1 TO 20
2030 CALL HCHAR(18,6+I,ASC(S
EG$("YOU CAN'T MOVE THERE!",
I,1)))
2040 NEXT I
2050 FOR DELAY=1 TO 200
2060 NEXT DELAY
2070 CALL HCHAR(18,7,32,20)
2080 GOTO 1160
2090 REM   WE HAVE A WINNER!

2100 FOR I=1 TO 17
2110 CALL HCHAR(18,6+I,ASC(S
EG$("WE HAVE A WINNER!",I,1)
))
2120 NEXT I
2130 FOR DELAY=1 TO 200
2140 NEXT DELAY
2150 RETURN
2160 REM
2170 REM   END
2180 REM
2190 IF N=67 THEN 2220
2200 PRINT TAB(9);"PLAYER #"
;P;" HAS DONE IT!   GOOD JOB
!"
2210 GOTO 2230
```

```
2220 PRINT "A CAT'S GAME!  T
HERE IS NO  WINNER!"
2230 PRINT
2240 INPUT "PRESS ENTER TO C
LEAR SCREEN:":ANS$
2250 CALL CLEAR
2260 RETURN
```



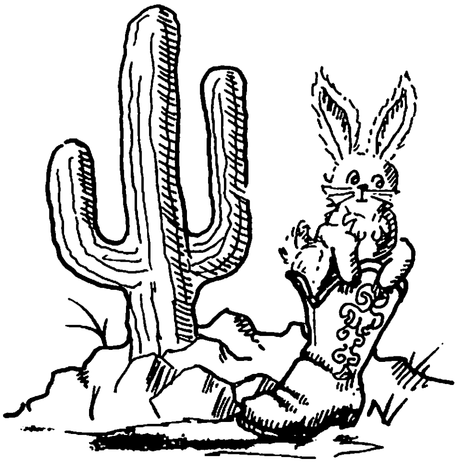
In this popular word game, you try to surmise the "secret word" by guessing the individual letters contained therein. Failure to guess the entire word will cause the completed figure to be "hung."

Lines 800-890 draw the gallows. To verify this, put a STOP command at line 795, run the program, and then relocate the STOP command at 895. On the next run, the gallows will be drawn before the computer STOPs.

The CALL COLOR statements contain three distinct parts. For example, CALL COLOR(6,3,5) can be broken down this way: the first number represents the Character Set (in this case, #6 - characters 72-79); the second number (3) sets the foreground color to medium green (color 3); the last number sets the background color to dark blue (5). In summary, this line reads: draw characters from Character Set #6 (H-O) in medium green on a dark blue background. When the foreground color and the background color match, then the character is, in effect, invisible.

```
10 REM ********************
20 REM ***              ***
30 REM ***    HANGMAN   ***
40 REM ***              ***
50 REM ********************
60 GOSUB 110
70 GOSUB 390
80 GOSUB 590
90 GOSUB 2340
100 END
110 REM
120 REM ***INSTRUCTIONS***
130 REM
140 GOSUB 350
150 PRINT "IN THIS GAME YOU
ARE GIVEN  THE FORMAT OF A W
ORD.  YOU"
160 PRINT "TRY TO SPELL OUT
THE MYSTERYWORD BY GUESSING
ONE LETTER AT A TIME."
170 PRINT
180 PRINT "IF THE LETTER IS
IN THE     WORD, THEN ALL OC
CURRENCES"
190 PRINT "OF THAT LETTER, I
N THEIR    CORRECT POSITION,
 WILL BE"
200 PRINT "EXPOSED.  IF YOU
GUESS A    LETTER NOT FOUND
IN THE     WORD, THEN ANOTHE
R BODY"
210 PRINT "PART WILL BE ADDE
D TO THE   MAN IN THE GALLOW
S."
220 FOR I=1 TO 5
230 PRINT
240 NEXT I
250 GOSUB 330
260 GOSUB 350
270 PRINT "WHEN THE FIGURE I
S COMPLETE (HEAD, NECK, TRUN
K, ARMS,   AND LEGS), HE IS
HUNG - AND YOU LOSE."
280 PRINT
290 PRINT "PLEASE MAKE YOUR
GUESSES    USING UPPER CASE
ONLY."
300 FOR I=1 TO 13
310 PRINT
320 NEXT I
330 INPUT "PRESS ENTER WHEN
```

```
READY TO   CONTINUE: ":ANS$
340 RETURN
350 CALL CLEAR
360 PRINT TAB(8);"*** HANGMA
N ***"
370 PRINT
380 RETURN
390 REM
400 REM ***SETUP***
410 REM
420 CALL CLEAR
430 RANDOMIZE
440 DIM G$(12),P1(25),P2(25)
450 P$="    YOUR GUESS :
    "
460 FOR I=1 TO 25
470 P1(I)=ASC(SEG$(P$,I,1))
480 NEXT I
490 P$="CHOOSE A DIFFERENT L
ETTER"
500 FOR I=1 TO 25
510 P2(I)=ASC(SEG$(P$,I,1))
520 NEXT I
530 CALL COLOR(3,2,2)
540 CALL COLOR(9,14,14)
550 CALL COLOR(10,16,16)
560 CALL COLOR(11,5,5)
570 CALL COLOR(12,13,13)
580 RETURN
590 REM
600 REM ***PLAY***
610 REM
620 CALL CLEAR
630 RESTORE
640 FOR I=1 TO RND*32+1
650 READ WRD$
660 NEXT I
670 DATA PENCIL,COMPUTER,ELE
PHANT,NOTEBOOK,TOGETHER,COUR
AGE
680 DATA COOKIES,PLEASURE,TE
ACHER,REGULAR,ROUTINE,POSTER
,CEILING
690 DATA FORESTRY,PAVEMENT,B
ARRIER,GRAVITY,REGULAR,IRIDE
SCENT
700 DATA HANGMAN,EVERGREEN,F
OOTBALL,SCOUNDREL,MARRIAGE,T
HOUGHTFUL,BARBECUE
710 DATA BOOKKEEPER,PRINTER,
PROFILE,YESTERDAY,MIRROR,MON
```

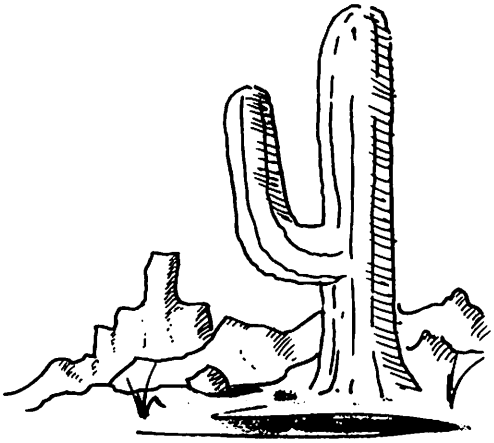


```
OPOLY
720 PG$=""
730 C=0
740 HM=0
750 AA=0
760 Q=LEN(WRD$)
770 FOR I=1 TO Q
780 G$(I)=SEG$(WRD$,I,1)
790 NEXT I
800 FOR X=6 TO 22
810 CALL VCHAR(X,9,52)
820 NEXT X
830 FOR X=10 TO 12
840 CALL HCHAR(18-X,X,52)
850 NEXT X
860 FOR X=9 TO 19
870 CALL HCHAR(5,X,52)
880 NEXT X
890 CALL HCHAR(6,19,52)
900 P$="GUESSES :"
910 FOR I=1 TO 9
920 CALL HCHAR(24,I+3,ASC(SE
G$(P$,I,1)))
930 NEXT I
940 WL=1
950 FOR S=INT(18.5-Q/2)TO IN
T(17.5+Q/2)
960 CALL HCHAR(21,S,95)
970 NEXT S
980 FOR A=1 TO 25
990 CALL HCHAR(23,A+3,P1(A))
1000 NEXT A
1010 CALL KEY(0,G,T)
1020 IF T=0 THEN 1010
1030 REM
1040 REM  YOUR GUESS
1050 REM
1060 FOR I=1 TO LEN(PG$)
1070 IF CHR$(G)=SEG$(PG$,I,1
)THEN 1890
1080 NEXT I
1090 DF=0
1100 S=INT(17.5-Q/2)
1110 FOR I=1 TO Q
1120 S=S+1
1130 DF=DF+1
1140 IF G=ASC(G$(I))THEN 195
0
1150 NEXT I
1160 IF DF=Q THEN 2020
1170 IF C=Q THEN 2110
```

```
1180 PG$=PG$&CHR$(G)
1190 GOTO 980
1200 REM
1210 REM  HEAD
1220 REM
1230 FOR X=7 TO 9
1240 FOR Z=18 TO 20
1250 CALL HCHAR(X,Z,97)
1260 NEXT Z
1270 NEXT X
1280 GOTO 980
1290 REM
1300 REM  NECK
1310 REM
1320 CALL HCHAR(10,19,108)
1330 GOTO 980
1340 REM
1350 REM  TRUNK
1360 REM
1370 FOR X=11 TO 14
1380 FOR Y=18 TO 20
1390 CALL HCHAR(X,Y,116)
1400 NEXT Y
1410 NEXT X
1420 GOTO 980
1430 REM
1440 REM  LEFT ARM
1450 REM
1460 CALL HCHAR(11,21,52)
1470 CALL HCHAR(11,22,52)
1480 FOR X=12 TO 13
1490 CALL HCHAR(X,22,52)
1500 NEXT X
1510 GOTO 980
1520 REM
1530 REM  RIGHT ARM
1540 REM
1550 CALL HCHAR(11,16,52)
1560 CALL HCHAR(11,17,52)
1570 CALL HCHAR(12,16,52)
1580 CALL HCHAR(13,16,52)
1590 GOTO 980
1600 REM
1610 REM  LEFT LEG
1620 REM
1630 FOR X=15 TO 18
1640 CALL VCHAR(X,20,122)
1650 NEXT X
1660 GOTO 980
1670 REM
1680 REM  RIGHT LEG
```

```
1690 REM
1700 FOR X=15 TO 18
1710 CALL VCHAR(X,18,122)
1720 NEXT X
1730 CALL SOUND(600,165,2)
1740 CALL SOUND(1200,110,2)
1750 FOR I=1 TO 600
1760 NEXT I
1770 CALL CLEAR
1780 PRINT TAB(4);"THE WORD
WAS - ";WRD$
1790 FOR I=1 TO 4
1800 PRINT
1810 NEXT I
1820 PRINT TAB(4);"BETTER LU
CK NEXT TIME!"
1830 FOR I=1 TO 10
1840 PRINT
1850 NEXT I
1860 FOR I=1 TO 500
1870 NEXT I
1880 GOTO 2230
1890 FOR I=1 TO 25
1900 CALL HCHAR(23,I+3,P2(I)
)
1910 NEXT I
1920 FOR I=1 TO 200
1930 NEXT I
1940 GOTO 980
1950 REM
1960 REM  GOOD GUESS
1970 REM
1980 CALL HCHAR(21,S,G)
1990 C=C+1
2000 DF=0
2010 GOTO 1150
2020 REM
2030 REM  BAD GUESS
2040 REM
2050 HM=HM+1
2060 AA=AA+2
2070 CALL HCHAR(24,AA+12,G)
2080 CALL HCHAR(24,AA+13,44)
2090 PG$=PG$&CHR$(G)
2100 ON HM GOTO 1200,1290,13
40,1430,1520,1600,1670
2110 REM
2120 REM   YOU GOT IT!
2130 REM
2140 CALL SOUND(300,440,2)
2150 CALL SOUND(600,660,2)
```

```
2160 FOR I=1 TO 200
2170 NEXT I
2180 CALL CLEAR
2190 P$="CONGRATULATIONS!!!"
2200 FOR I=1 TO 18
2210 CALL HCHAR(8,I+7,ASC(SE
G$(P$,I,1)))
2220 NEXT I
2230 FOR I=1 TO 100
2240 NEXT I
2250 CALL CLEAR
2260 P$="DO YOU WANT TO PLAY
 AGAIN?"
2270 FOR I=1 TO 26
2280 CALL HCHAR(8,I+3,ASC(SE
G$(P$,I,1)))
2290 NEXT I
2300 PRINT TAB(14);
2310 INPUT "Y/N? ":ANS$
2320 IF SEG$(ANS$,1,1)="Y" T
HEN 590
2330 RETURN
2340 REM
2350 REM ***END***
2360 REM
2370 CALL CLEAR
2380 PRINT TAB(4);"HOPE YOU
HAD FUN!"
2390 PRINT
2400 PRINT TAB(12);"BYE BYE"
2410 FOR I=1 TO 8
2420 PRINT
2430 NEXT I
2440 RETURN
```



ITCHE

In this game you build a maze for the maze-loving worm. Upon instruction (G for Go), the worm will attempt to solve the maze. Though not a game per se, it is an ingenious utilization of graphics.

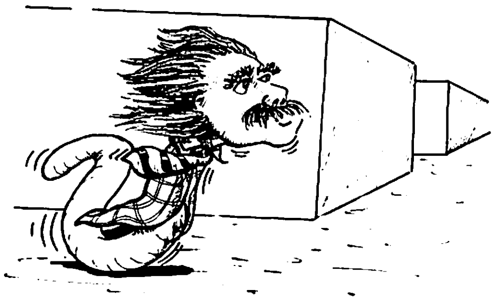
Line 490 sets the screen color to gray (color 15). For an in-depth explanation of CALL SCREEN, see MUBBLE CHASE. There are many CALL HCHAR (and VCHAR) statements in this program. You often want to know which Character Set (and which color) is being represented in a particular statement. As with converting Fahrenheit to Celsius and back again, there is a conversion formula. The conversion equation is:

`INTEGER VALUE OF ((CHARACTER ASCII VALUE - 24) / 8)`

Using this equation, see if you can get the character set represented by each of the following character ASCII values.

ASCII VALUE	EQUATION	CHARACTER SET
35	INT((35-24)/8)	1
81	INT((81-24)/8)	7
144	INT((144-24)/8)	15
151	INT((151-24)/8)	15
96	INT((96-24)/8)	9
116	INT((116-24)/8)	11

```
10 REM ********************
20 REM ***              ***
30 REM ***     ITCHE    ***
40 REM ***              ***
50 REM ********************
60 GOSUB 110
70 GOSUB 470
80 GOSUB 1560
90 GOSUB 2080
100 END
110 REM
120 REM ***INSTRUCTIONS***
130 REM
140 GOSUB 430
150 PRINT "THIS IS THE GAME
OF ITCHE.  ITCHE IS A SPECIA
L WORM WHO LIKES TO SOLVE MA
ZES. YOU"
160 PRINT "ARE TO CREATE A M
AZE FOR THEITCHE TO SOLVE."
170 PRINT
180 PRINT "BY USING THE FOLL
OWING      INSTRUCTIONS, YOU
 WILL BUILDTHE MAZE WALLS. U
PON COMMAND"
190 PRINT "(G), THE ITCHE WI
LL WIND ITSWAY THROUGH THE M
AZE."
200 FOR I=1 TO 9
210 PRINT
220 NEXT I
230 GOSUB 410
240 GOSUB 430
250 PRINT "THE FOLLOWING COM
MANDS WILL AID YOU:"
260 PRINT
270 PRINT "G)O, TO HAVE THE
ITCHE WORM FIND THE EXIT TO
YOUR MAZE."
280 PRINT
290 PRINT "Q)UIT, TO END THE
 GAME."
300 PRINT
310 PRINT "C)LEAR, TO START
THE MAZE   OVER."
320 PRINT
330 PRINT "P)LOT, M)OVE, AND
 E)RASE    FOLLOWED BY DIREC
TION KEYS:"
340 PRINT
350 PRINT TAB(13);"A=UP"
```



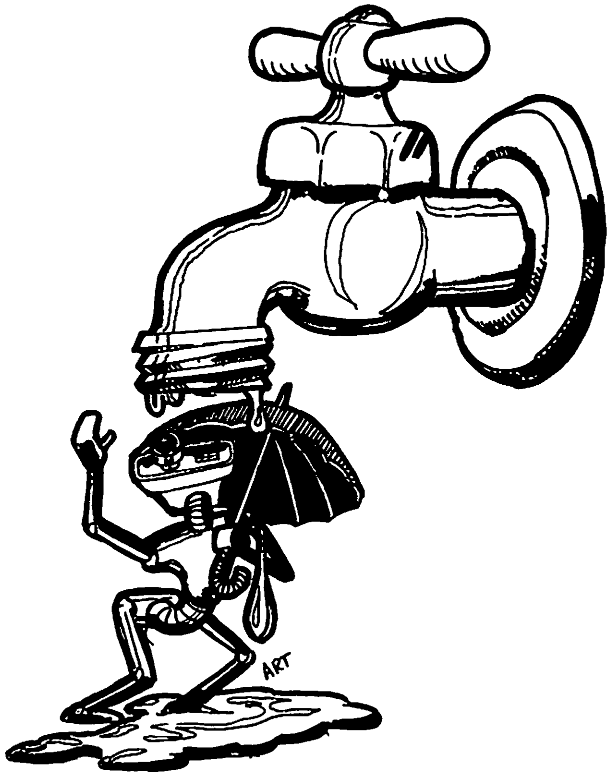
```
360 PRINT "    K=LEFT
 L=RIGHT"
370 PRINT TAB(12);"Z=DOWN"
380 FOR I=1 TO 3
390 PRINT
400 NEXT I
410 INPUT "PRESS ENTER WHEN
READY TO   CONTINUE: ":ANS$
420 RETURN
430 CALL CLEAR
440 PRINT TAB(9);"*** ITCHE
***"
450 PRINT
460 RETURN
470 REM
480 REM ***SETUP***
490 CALL SCREEN(15)
500 CALL CLEAR
510 X2=1
520 Y2=1
530 CALL CHAR(120,"FFFFFFFFF
FFFFFFF")
540 CALL CHAR(128,"FFFFFFFFF
FFFFFFF")
550 CALL CHAR(136,"FFFFFFFFF
FFFFFFF")
560 CALL CHAR(144,"FFFFFFFFF
FFFFFFF")
570 CALL CHAR(152,"FFFFFFFFF
FFFFFFF")
580 CALL COLOR(12,15,15)
590 CALL COLOR(13,5,15)
600 CALL COLOR(14,13,15)
610 CALL COLOR(15,11,15)
620 CALL COLOR(16,7,15)
630 CALL HCHAR(1,3,136,29)
640 CALL HCHAR(22,3,136,29)
650 CALL VCHAR(3,3,136,19)
660 CALL VCHAR(3,31,136,19)
670 IX=2
680 IY=5
690 CALL HCHAR(IX,IY,144)
700 CLR=120
710 FOR I=1 TO 27
720 CALL HCHAR(23,I+3,ASC(SE
G$("P)LOT M)OVE E)RASE C)LR
G)O",I,1)))
730 NEXT I
740 FOR I=1 TO 25
750 CALL HCHAR(24,I+4,ASC(SE
G$("A=UP Z=DN K=LF L=RT Q)UI
```

```
T",I,1)))
760 NEXT I
770 CALL KEY(3,ANS,STAT)
780 IF STAT=0 THEN 770
790 ANS$=CHR$(ANS)
800 IF ANS$="C" THEN 490
810 IF ANS$="P" THEN 880
820 IF ANS$="M" THEN 910
830 IF ANS$="E" THEN 950
840 IF ANS$="Q" THEN 980
850 IF ANS$="G" THEN 1540
860 CALL SOUND(200,1760,5)
870 GOTO 770
880 REM ***PLOT ROUTINE***
890 CLR=128
900 GOTO 1110
910 REM ***MOVE ROUTINE***
920 TMP=200
930 C2=120
940 GOTO 1110
950 REM ***ERASE ROUTINE***
960 CLR=120
970 GOTO 1110
980 REM ***QUIT ROUTINE***
990 CALL HCHAR(23,1,32,32)
1000 CALL HCHAR(24,1,32,32)
1010 FOR I=1 TO 24
1020 CALL HCHAR(23,I+4,ASC(S
EG$("DO YOU WISH TO QUIT Y/N
:",I,1)))
1030 NEXT I
1040 CALL KEY(3,ANS,STAT)
1050 IF STAT=0 THEN 1040
1060 ANS$=CHR$(ANS)
1070 IF ANS$="Y" THEN 1540
1080 IF ANS$<>"N" THEN 1040
1090 CALL HCHAR(23,1,32,32)
1100 GOTO 710
1110 REM ***DIRECTIONS***
1120 CALL KEY(3,ANS,STAT)
1130 IF STAT=0 THEN 1120
1140 X2=IX
1150 Y2=IY
1160 ANS$=CHR$(ANS)
1170 IF ANS$="A" THEN 1230
1180 IF ANS$="Z" THEN 1280
1190 IF ANS$="K" THEN 1330
1200 IF ANS$="L" THEN 1380
1210 TMP=0
1220 GOTO 800
1230 REM ***UP***
```

```
1240 X2=IX-1
1250 IF X2>1 THEN 1430
1260 X2=2
1270 GOTO 1420
1280 REM ***DOWN***
1290 X2=IX+1
1300 IF X2<22 THEN 1430
1310 X2=21
1320 GOTO 1420
1330 REM ***LEFT***
1340 Y2=IY-1
1350 IF Y2>3 THEN 1430
1360 Y2=4
1370 GOTO 1420
1380 REM ***RIGHT***
1390 Y2=IY+1
1400 IF Y2<31 THEN 1430
1410 Y2=30
1420 CALL SOUND(200,1760,5)
1430 IF TMP<>200 THEN 1450
1440 CLR=C2
1450 CALL GCHAR(IX,IY,CK)
1460 AUX=CK
1470 CALL HCHAR(IX,IY,CLR)
1480 CALL GCHAR(X2,Y2,CK)
1490 C2=CK
1500 CALL HCHAR(X2,Y2,AUX)
1510 IX=X2
1520 IY=Y2
1530 GOTO 1110
1540 REM ***GO ROUTINE***
1550 RETURN
1560 REM
1570 REM ***PLAY***
1580 REM
1590 IF ANS$="Y" THEN 2070
1600 CALL HCHAR(23,1,32,32)
1610 CALL HCHAR(24,1,32,32)
1620 FOR I=1 TO 19
1630 CALL HCHAR(23,I+6,ASC(S
EG$("<ITCHE NOW SOLVING>",I,
1)))
1640 NEXT I
1650 X2=2
1660 Y2=4
1670 CALL HCHAR(X2,Y2,152)
1680 IF TMP<>200 THEN 1700
1690 CLR=C2
1700 CALL HCHAR(IX,IY,CLR)
1710 X3=1
1720 Y3=0
```

```
1730 DI=2
1740 IF (X2+X3=2)*(Y2+Y3=3)T
HEN 2070
1750 IF (X2+X3=2)*(Y2+Y3=31)
THEN 2070
1760 CALL GCHAR(X2+X3,Y2+Y3,
CK1)
1770 IF (X2+X3>1)*(X2+X3<22)
*(Y2+Y3>3)*(Y2+Y3<31)*((CK1=
32)+(CK1=120))THEN 1970
1780 DI=DI-1
1790 IF DI>=1 THEN 1810
1800 DI=4
1810 IF DI<>1 THEN 1850
1820 X3=0
1830 Y3=1
1840 GOTO 1740
1850 IF DI<>2 THEN 1890
1860 X3=1
1870 Y3=0
1880 GOTO 1740
1890 IF DI<>3 THEN 1930
1900 X3=0
1910 Y3=-1
1920 GOTO 1740
1930 IF DI<>4 THEN 1970
1940 X3=-1
1950 Y3=0
1960 GOTO 1740
1970 CALL HCHAR(X2,Y2,120)
1980 X2=X2+X3
1990 Y2=Y2+Y3
2000 CALL HCHAR(X2,Y2,152)
2010 CL=CL+1
2020 DI=DI+1
2030 IF DI<=4 THEN 2050
2040 DI=1
2050 CALL SOUND(100,1760,10)
2060 GOTO 1810
2070 RETURN
2080 REM
2090 REM ***END***
2100 CALL HCHAR(X2,Y2,120)
2110 X2=X2+X3
2120 Y2=Y2+Y3
2130 FOR I=1 TO 10
2140 CALL HCHAR(X2,Y2,152)
2150 CALL HCHAR(X2,Y2,120)
2160 CALL SOUND(100,1760,10)
2170 NEXT I
2180 IF ANS$="Y" THEN 2190 E
```

```
LSE 2230
2190 CALL HCHAR(23,1,32,32)
2200 CALL HCHAR(24,1,32,32)
2210 PRINT "ITCHE SAYS YOU S
POILED HIS  FUN"
2220 GOTO 2290
2230 IF Y2<>31 THEN 2270
2240 PRINT "ITCHE HAS SOLVED
 THE MAZE"
2250 PRINT "HE DID IT IN ";C
L;" CLICKS"
2260 GOTO 2290
2270 PRINT "ITCHE CANNOT SOL
VE YOUR MAZE"
2280 PRINT "HE IS STUCK AT T
HE BEGINNING"
2290 RETURN
```



Leaky Faucet

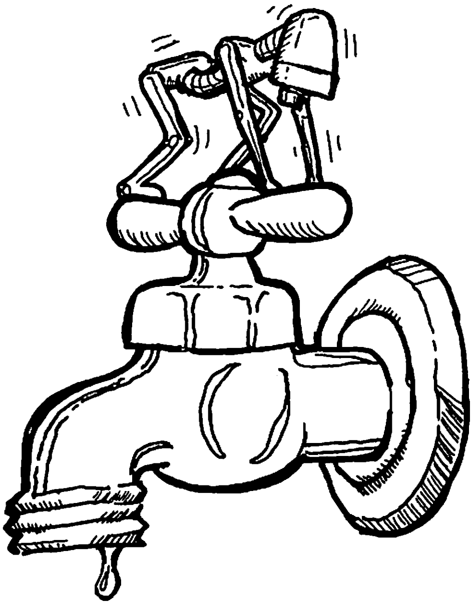
This is not a game. Rather, it is an excellent parade of graphics. The idea is to create a colorful simulation of a dripping faucet.

Lines 470-540 draw the faucet. Using the conversion formula explained in Itche, see if you can identify the faucet's colors without running the program. Line 440 gives the CoLoR (CLR) a RaNDom value between 88 and 152. This is done to give the beginning configuration a varying number of squares of each color. The numbers which are generated (88, 96, 104, 112, 120, 128, 136, 144 and 152) are used in line 450. These numbers correspond to an ASCII value in one of nine different Character Sets. Experiment with line 440 to see how it affects the program. Likewise, change or omit any line whose function is unclear.

```
10 REM ********************
20 REM ***              ***
30 REM *** LEAKY FAUCET ***
40 REM ***              ***
50 REM ********************
60 GOSUB 100
70 GOSUB 260
80 GOSUB 570
90 END
100 REM
110 REM ***INSTRUCTIONS***
120 REM
130 GOSUB 220
140 PRINT "THIS IS NOT REALL
Y A GAME,  BUT AN EXTREMELY
ENTER-      TAINING GRAPHICS
DEMO."
150 PRINT
160 PRINT "WE HOPE YOU ENJOY
 IT!!!"
170 FOR I=1 TO 15
180 PRINT
190 NEXT I
200 INPUT "PRESS ENTER WHEN
READY TO   CONTINUE: ":ANS$
210 RETURN
220 CALL CLEAR
230 PRINT TAB(6);"*** LEAKY
FAUCET ***"
240 PRINT
250 RETURN
260 REM
270 REM ***SETUP***
280 REM
290 CALL CLEAR
300 RANDOMIZE
310 FOR I=8 TO 16
320 CALL COLOR(I,I,I)
330 NEXT I
340 CALL COLOR(3,5,5)
350 CALL COLOR(2,2,2)
360 PRINT "LEFT <== <S>   <D
> ==> RIGHT"
370 PRINT TAB(7);"PRESS <Q>
TO QUIT";
380 CALL HCHAR(22,1,40,32)
390 CALL VCHAR(6,1,40,16)
400 CALL VCHAR(6,32,40,16)
410 FOR I=1 TO 150
420 ROW=INT(RND*14+8)
430 COLUMN=INT(RND*30+2)
```

```
440 CLR=INT(RND*9+8)*8+24
450 CALL HCHAR(ROW,COLUMN,CL
R)
460 NEXT I
470 REM  DRAW THE FAUCET
480 CALL HCHAR(1,5,40,5)
490 CALL HCHAR(2,7,40)
500 CALL HCHAR(3,1,104,4)
510 CALL HCHAR(3,5,120,3)
520 CALL HCHAR(4,1,104,4)
530 CALL HCHAR(4,5,120,3)
540 CALL HCHAR(5,7,40)
550 I=7
560 RETURN
570 REM
580 REM ***PLAY***
590 REM
600 SY=7
610 SX=6
620 SY=I
630 GOSUB 980
640 IF ANS$="Q" THEN 1310
650 CALL HCHAR(SX,I,48)
660 IF SX+1>21 THEN 730
670 CALL GCHAR(SX+1,SY,CK1)
680 IF CK1<>32 THEN 730
690 GOSUB 960
700 SX=SX+1
710 GOSUB 940
720 GOTO 660
730 LR=INT(RND*2)*2-1
740 CALL GCHAR(SX,SY+LR,CK2)
750 CALL GCHAR(SX,SY-LR,CK3)
760 IF (CK2<>32)*(CK3<>32)TH
EN 610
770 IF CK2<>32 THEN 860
780 GOSUB 960
790 SY=SY+LR
800 GOSUB 940
810 CALL GCHAR(SX+1,SY,CK1)
820 IF CK1=32 THEN 690
830 CALL GCHAR(SX,SY+LR,CK2)
840 IF CK2=32 THEN 780
850 GOTO 610
860 GOSUB 960
870 SY=SY-LR
880 GOSUB 940
890 CALL GCHAR(SX+1,SY,CK1)
900 IF CK1=32 THEN 690
910 CALL GCHAR(SX,SY-LR,CK3)
920 IF CK3=32 THEN 860
```

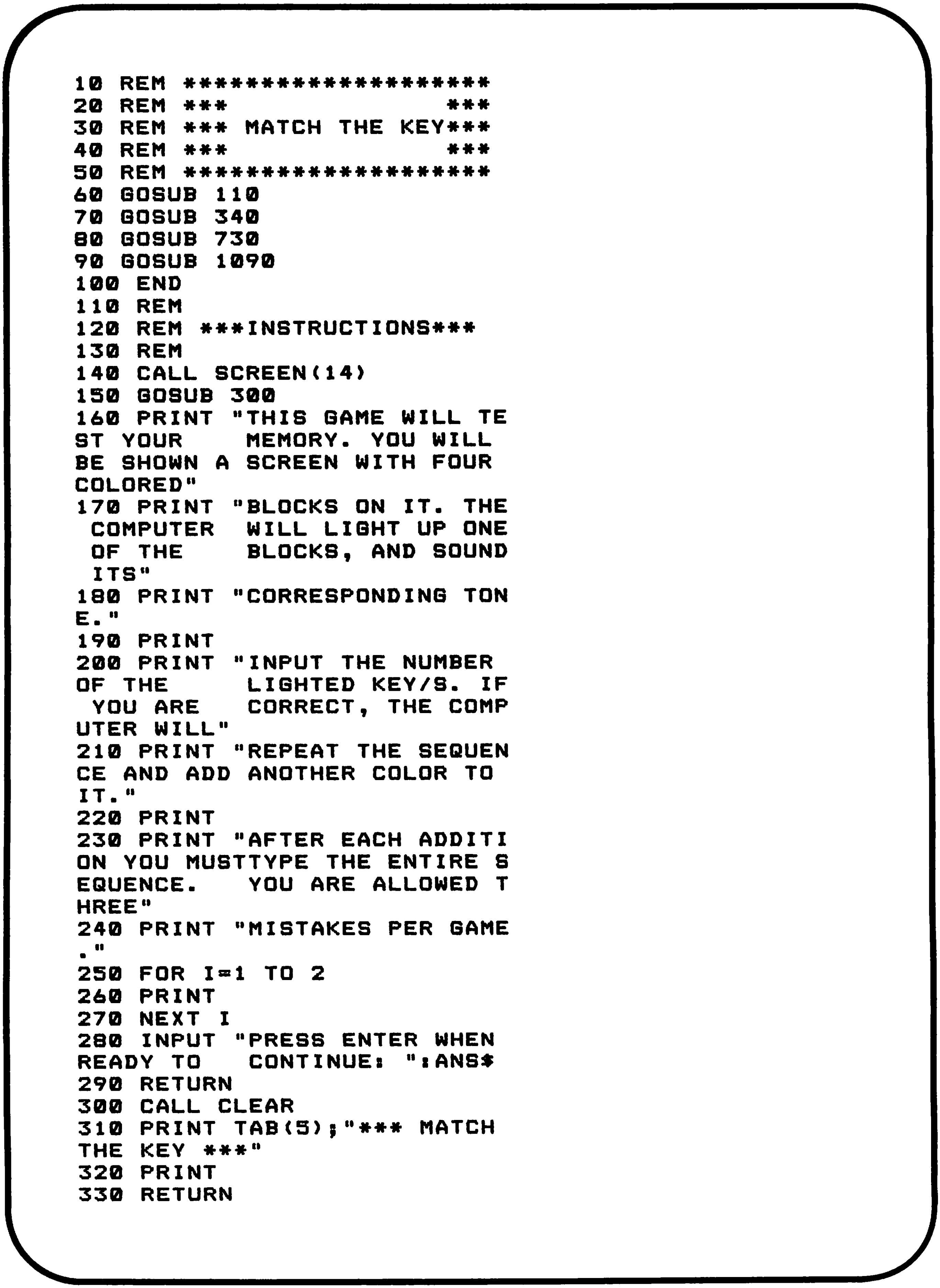
```
930 GOTO 610
940 CALL HCHAR(SX,SY,48)
950 RETURN
960 CALL HCHAR(SX,SY,32)
970 RETURN
980 REM  MOVE FAUCET
990 CALL KEY(3,ANS,STAT)
1000 SY=I
1010 IF STAT=0 THEN 1070
1020 ANS$=CHR$(ANS)
1030 IF (ANS$="S")+(ANS$="D"
)+(ANS$="Q")THEN 1040 ELSE 1
070
1040 IF ANS$="Q" THEN 1070
1050 IF ANS$="S" THEN 1080
1060 IF I=29 THEN 1310 ELSE
1200
1070 RETURN
1080 REM  ***LEFT***
1090 IF I=7 THEN 1310
1100 CALL HCHAR(1,I+2,32)
1110 CALL HCHAR(2,I,32)
1120 CALL HCHAR(5,I,32)
1130 CALL VCHAR(3,I,32,2)
1140 I=I-1
1150 CALL HCHAR(1,I-2,40)
1160 CALL HCHAR(2,I,40)
1170 CALL HCHAR(5,I,40)
1180 CALL VCHAR(3,I-2,120,2)
1190 GOTO 980
1200 REM  ***RIGHT***
1210 CALL HCHAR(1,I-2,32)
1220 CALL HCHAR(2,I,32)
1230 CALL HCHAR(5,I,32)
1240 I=I+1
1250 CALL VCHAR(3,I,120,2)
1260 CALL HCHAR(5,I,40)
1270 CALL HCHAR(2,I,40)
1280 CALL HCHAR(1,I+2,40)
1290 CALL VCHAR(3,I-3,104,2)
1300 GOTO 980
1310 RETURN
```



Match the Key

If you have ever played or seen *Simon*, then you will recognize this game. The object is to type the numeric equivalent of a series of tones. There are a number of mnemonic devices which simplify recall — the trick is to develop your own technique. A very good player can repeat a sequence of twenty tones, and an expert can usually repeat a chain of thirty.

The graphics in this program are interesting, so let's take a look. Lines 570-610 draw the original four boxes. To prove this, type in: 611 STOP. Run the program. Now type in: 569 STOP. Run the program again. You will note that the second run stopped before the boxes were drawn. This method of verifying line functions is a sound technique. Use it anywhere; use it often.

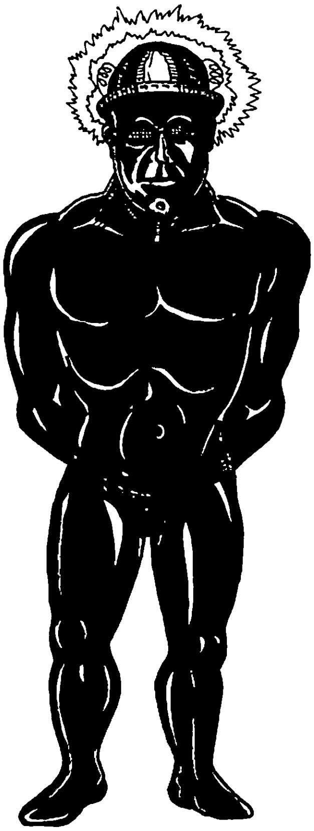


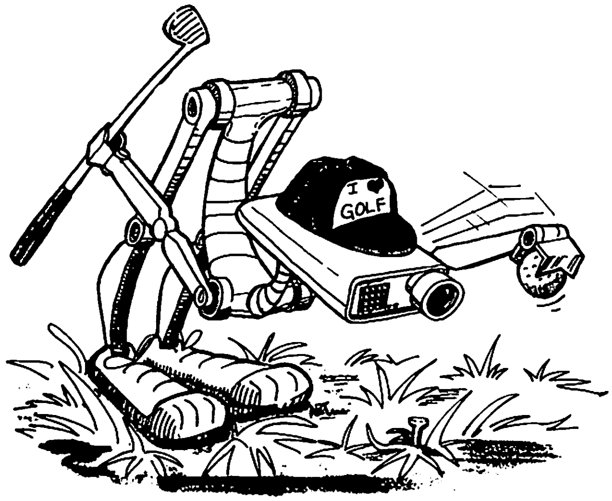
```
10 REM ********************
20 REM ***              ***
30 REM *** MATCH THE KEY***
40 REM ***              ***
50 REM ********************
60 GOSUB 110
70 GOSUB 340
80 GOSUB 730
90 GOSUB 1090
100 END
110 REM
120 REM ***INSTRUCTIONS***
130 REM
140 CALL SCREEN(14)
150 GOSUB 300
160 PRINT "THIS GAME WILL TE
ST YOUR    MEMORY. YOU WILL
BE SHOWN A SCREEN WITH FOUR
COLORED"
170 PRINT "BLOCKS ON IT. THE
 COMPUTER  WILL LIGHT UP ONE
 OF THE    BLOCKS, AND SOUND
 ITS"
180 PRINT "CORRESPONDING TON
E."
190 PRINT
200 PRINT "INPUT THE NUMBER
OF THE     LIGHTED KEY/S. IF
 YOU ARE   CORRECT, THE COMP
UTER WILL"
210 PRINT "REPEAT THE SEQUEN
CE AND ADD ANOTHER COLOR TO
IT."
220 PRINT
230 PRINT "AFTER EACH ADDITI
ON YOU MUSTTYPE THE ENTIRE S
EQUENCE.   YOU ARE ALLOWED T
HREE"
240 PRINT "MISTAKES PER GAME
."
250 FOR I=1 TO 2
260 PRINT
270 NEXT I
280 INPUT "PRESS ENTER WHEN
READY TO   CONTINUE: ":ANS$
290 RETURN
300 CALL CLEAR
310 PRINT TAB(5);"*** MATCH
THE KEY ***"
320 PRINT
330 RETURN
```

```
340 REM
350 REM ***SETUP***
360 REM
370 CALL CLEAR
380 CALL SCREEN(1)
390 RANDOMIZE
400 DIM SEQ(50),MC(4)
410 FOR I=1 TO 4
420 READ MC(I)
430 NEXT I
440 DATA 440,740,1175,1760
450 FOR I=1 TO 50
460 SEQ(I)=INT((RND)*4)+1
470 NEXT I
480 CALL CHAR(128,"FFFFFFFF
FFFFFFF")
490 CALL CHAR(136,"FFFFFFFF
FFFFFFF")
500 CALL COLOR(13,14,1)
510 CALL COLOR(14,16,1)
520 CALL COLOR(3,15,1)
530 CALL COLOR(4,15,1)
540 CALL COLOR(5,15,1)
550 CALL COLOR(6,15,1)
560 CALL COLOR(7,15,1)
570 FOR I=1 TO 4
580 FOR J=1 TO 28 STEP 7
590 CALL HCHAR(12+I,4+J,128,
4)
600 NEXT J
610 NEXT I
620 I=0
630 FOR J=1 TO 26 STEP 7
640 CALL HCHAR(19,5+J,49+I)
650 I=I+1
660 NEXT J
670 FOR I=1 TO 7
680 CALL HCHAR(22,5+I,ASC(SE
G$("MISSES:",I,1)))
690 CALL HCHAR(22,19+I,ASC(S
EG$("SCORE: ",I,1)))
700 NEXT I
710 GOSUB 1280
720 RETURN
730 REM
740 REM ***PLAY***
750 REM
760 FOR Z=1 TO 50
770 SC=(Z-1)-MS
780 GOSUB 1280
790 FOR J=1 TO Z
```

```
800 FOR L=1 TO 4
810 CALL HCHAR(12+L,SEQ(J)*7
-2,136,4)
820 NEXT L
830 CALL SOUND(100,MC(SEQ(J)
),2)
840 FOR K=1 TO 4
850 CALL HCHAR(12+K,SEQ(J)*7
-2,128,4)
860 NEXT K
870 NEXT J
880 FOR J=1 TO Z
890 CALL KEY(3,ANS,STAT)
900 IF STAT=0 THEN 890
910 IF (ANS<49)+(ANS>52)THEN
 890
920 FOR I=1 TO 4
930 CALL HCHAR(12+I,(ANS-48)
*7-2,136,4)
940 NEXT I
950 CALL SOUND(100,MC(ANS-48
),2)
960 FOR I=1 TO 4
970 CALL HCHAR(12+I,(ANS-48)
*7-2,128,4)
980 NEXT I
990 IF ANS-48=SEQ(J)THEN 104
0
1000 CALL SOUND(500,220,2)
1010 MS=MS+1
1020 IF MS<3 THEN 1050
1030 RETURN
1040 NEXT J
1050 FOR PA=1 TO 250
1060 NEXT PA
1070 NEXT Z
1080 RETURN
1090 REM
1100 REM ***END***
1110 REM
1120 CALL CLEAR
1130 GOTO 1230
1140 PRINT "<<<I'M SORRY YOU
 LOST...>>>"
1150 PRINT
1160 GOTO 1260
1170 PRINT "<<<I'M SORRY YOU
 LOST...AND YOU GOT THEM ALL
 WRONG>>>"
1180 PRINT
1190 GOTO 1260
```

```
1200 PRINT "YOU GOT ALL 50 O
F THEM!!!"
1210 PRINT
1220 GOTO 1260
1230 IF (MS=3)*(SC<1)THEN 11
70
1240 IF MS=3 THEN 1140
1250 GOTO 1200
1260 PRINT "YOUR SCORE WAS "
;SC;" OUT OF 50"
1270 RETURN
1280 REM
1290 REM ***TABULATIONS***
1300 REM
1310 X=INT(SC/10)
1320 TMP=X*10
1330 Y=SC-TMP
1340 XX=INT(MS/10)
1350 TMP=XX*10
1360 YY=MS-TMP
1370 CALL HCHAR(22,14,ASC(ST
R$(XX)))
1380 CALL HCHAR(22,15,ASC(ST
R$(YY)))
1390 CALL HCHAR(22,28,ASC(ST
R$(X)))
1400 CALL HCHAR(22,29,ASC(ST
R$(Y)))
1410 RETURN
```





Miniature GOLF

This game is a simulation of miniature golf. There are hazards, obstacles, and unplayable lies, just like in the real thing. The direction and speed of each shot is determined by your input. On each hole, par (the standard number of strokes needed to get the ball in the hole) is determined by the difficulty, shape, and number of hazards.

List lines 520-680. Line 530 sets the background color to green. This is done via the CALL SCREEN command. The screen can be set to any of the 15 colors (the 16th "color" is transparent). Copy and run the following program to help clarify this command:

```
 10 CALL CLEAR
 20 FOR I=1 TO 7
 30 CALL HCHAR(12,12+I,ASC(SEG$("COLOR #",I,1)))
 40 NEXT I
 50 FOR I=2 TO 16
 60 CALL SCREEN (I)
 70 IF I>9 THEN 100
 80 CALL HCHAR (12,21,I+48)
 90 GOTO 120
100 CALL HCHAR(12,20,49)
110 CALL HCHAR(12,21,I+38)
120 FOR D=1 TO 200
130 NEXT D
140 NEXT I
150 END
```

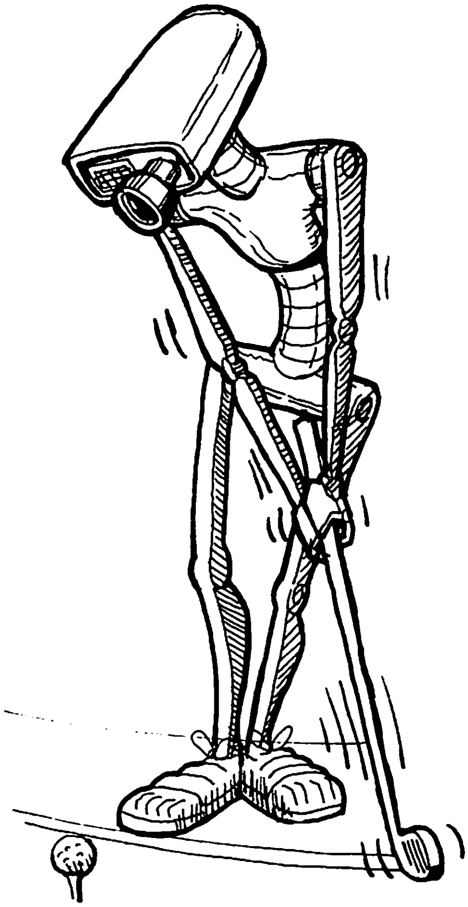
The 9-hole course is designed one hole at a time, i.e., hole 1 begins at line 910, hole 2 begins at line 1010, and so on. Look at the loop between lines 640-680. By determining which Character Set is specified as red, and calculating which characters are contained within that Character Set, you can determine which line(s) draw the perimeter of the hole. The same process will help you find the line that draws the single character which represents the hole.

```
10 REM ********************
20 REM ***              ***
30 REM ***MINIATURE GOLF***
40 REM ***              ***
50 REM ********************
60 GOSUB 110
70 GOSUB 490
80 GOSUB 2350
90 GOSUB 4530
100 END
110 REM
120 REM ***INSTRUCTIONS***
130 REM
140 CALL SCREEN(14)
150 GOSUB 450
160 PRINT "WELCOME!! WE HOPE
 YOU ENJOY YOUR MINIATURE GO
LF GAME."
170 PRINT
180 PRINT "TO PUTT THE BALL,
 YOU HAVE  TO ENTER WHICH DI
RECTION YOUWANT TO AIM IT.
THE EIGHT"
190 PRINT "DIRECTIONS ARE SH
OWN BELOW."
200 PRINT "YOUR BALL IS ASSU
MED TO BE  AT * :"
210 PRINT TAB(13);"2 1 8"
220 PRINT TAB(13);"3 * 7"
230 PRINT TAB(13);"4 5 6"
240 PRINT
250 PRINT
260 PRINT "THEN YOU MUST ENT
ER HOW HARDTO HIT THE BALL.
 THE SPEED SHOULD BE A NUMBE
R BETWEEN"
270 PRINT "0.0 AND 5.0"
280 PRINT
290 GOSUB 420
300 GOSUB 450
310 PRINT "THERE ARE HAZARDS
:"
```

```
320 PRINT
330 PRINT "BLOCKS-PURPLE, YO
U MUST GO  AROUND THESE"
340 PRINT
350 PRINT "SAND-YELLOW, YOU
CANNOT PASSTHROUGH, PENALTY=
1 STROKE."
360 PRINT
370 PRINT "WATER-BLUE, LIKE
SAND,       WILL SLOW AND STO
P BALL,     PENALTY=1 STROKE.
"
380 PRINT
390 PRINT "UNEVEN-LIGHT BLUE
, WILL      CHANGE DIRECTION
OF BALL     RANDOMLY, NO PENA
LTY."
400 PRINT
410 PRINT "IF THE BALL IS HI
T TOO HARD,IT WILL GO PAST T
HE HOLE ANDKEEP ROLLING."
420 PRINT
430 INPUT "PRESS ENTER WHEN
READY TO    CONTINUE: ":ANS$
440 RETURN
450 CALL CLEAR
460 PRINT TAB(4);"*** MINIAT
URE GOLF ***"
470 PRINT
480 RETURN
490 REM
500 REM ***SETUP***
510 REM
520 CALL CLEAR
530 CALL SCREEN(3)
540 RANDOMIZE
550 EF$="3C7EFFFFFFFF7E3C"
560 FF$="FFFFFFFFFFFFFFFF"
570 CALL CHAR(104,FF$)
580 CALL CHAR(112,FF$)
590 CALL CHAR(120,FF$)
600 CALL CHAR(128,FF$)
610 CALL CHAR(136,EF$)
620 CALL CHAR(144,EF$)
630 CALL CHAR(152,FF$)
640 FOR I=9 TO 16
650 READ J
660 CALL COLOR(I,J,13)
670 NEXT I
680 DATA 9,3,5,14,11,16,9,8
690 DIM HA(9,5)
```

```
700 FOR I=1 TO 9
710 FOR J=1 TO 5
720 READ HA(I,J)
730 NEXT J
740 NEXT I
750 DATA 0,0,0,0,2
760 DATA 1,0,0,0,3
770 DATA 0,1,0,0,3
780 DATA 1,1,0,0,3
790 DATA 0,0,0,1,3
800 DATA 0,0,1,1,3
810 DATA 1,0,1,0,3
820 DATA 0,1,1,0,4
830 DATA 0,0,1,1,3
840 DIM DI(8,2)
850 FOR I=1 TO 8
860 READ DI(I,1)
870 READ DI(I,2)
880 NEXT I
890 DATA -1,0,-1,-1,0,-1,1,-
1,1,0,1,1,0,1,-1,1
900 RETURN
910 REM  HOLE 1
920 CALL HCHAR(2,8,120,20)
930 CALL HCHAR(20,8,120,20)
940 CALL VCHAR(2,8,120,18)
950 CALL VCHAR(2,27,120,18)
960 CALL HCHAR(5,17,144)
970 BY=15
980 BX=INT(RND*5)+15
990 CALL HCHAR(BX,BY,136)
1000 RETURN
1010 REM  HOLE 2
1020 CALL HCHAR(2,6,120,24)
1030 CALL HCHAR(20,6,120,14)
1040 CALL HCHAR(10,20,120,10
)
1050 CALL VCHAR(2,6,120,18)
1060 CALL VCHAR(2,29,120,8)
1070 CALL VCHAR(10,20,120,11
)
1080 CALL VCHAR(10,19,112,7)
1090 CALL VCHAR(11,18,112,4)
1100 CALL VCHAR(12,17,112,2)
1110 CALL VCHAR(12,16,112)
1120 CALL HCHAR(6,25,144)
1130 BY=10
1140 GOTO 980
1150 REM  HOLE 3
1160 CALL HCHAR(2,20,120,10)
1170 CALL HCHAR(9,4,120,17)
```

```
1180 CALL HCHAR(20,4,120,17)
1190 CALL HCHAR(16,20,120,11
)
1200 CALL VCHAR(9,4,120,11)
1210 CALL VCHAR(2,20,120,7)
1220 CALL VCHAR(2,30,120,15)
1230 CALL VCHAR(16,20,120,4)
1240 FOR I=1 TO 6
1250 CALL HCHAR(9+I,30-I,128
,I)
1260 NEXT I
1270 CALL HCHAR(5,25,144)
1280 BY=10
1290 GOTO 980
1300 REM  HOLE 4
1310 CALL HCHAR(2,4,120,27)
1320 CALL HCHAR(20,4,120,11)
1330 CALL HCHAR(10,14,120,8)
1340 CALL HCHAR(14,22,120,9)
1350 CALL VCHAR(2,4,120,18)
1360 CALL VCHAR(2,30,120,13)
1370 CALL VCHAR(10,14,120,11
)
1380 CALL VCHAR(10,22,120,5)
1390 CALL HCHAR(3,14,128,10)
1400 CALL HCHAR(4,15,128,7)
1410 CALL HCHAR(7,18,112)
1420 CALL HCHAR(8,15,112,6)
1430 CALL HCHAR(9,14,112,8)
1440 CALL HCHAR(11,26,144)
1450 BY=10
1460 GOTO 980
1470 REM  HOLE 5
1480 CALL HCHAR(2,5,120,23)
1490 CALL HCHAR(20,5,120,23)
1500 CALL HCHAR(8,8,120,18)
1510 CALL VCHAR(2,5,120,19)
1520 CALL VCHAR(2,28,120,19)
1530 CALL VCHAR(5,8,120,3)
1540 CALL VCHAR(5,25,120,3)
1550 CALL HCHAR(6,16,144)
1560 BY=17
1570 GOTO 980
1580 REM  HOLE 6
1590 CALL HCHAR(2,4,120,27)
1600 CALL HCHAR(20,4,120,16)
1610 CALL HCHAR(12,19,120,12
)
1620 CALL VCHAR(2,4,120,18)
1630 CALL VCHAR(2,30,120,11)
1640 CALL VCHAR(13,19,120,8)
```



```
1650 CALL HCHAR(5,21,120,6)
1660 CALL HCHAR(10,21,120,6)
1670 CALL VCHAR(5,21,120,6)
1680 FOR I=14 TO 18
1690 CALL VCHAR(9,I,152,7)
1700 NEXT I
1710 CALL VCHAR(10,19,152,2)
1720 CALL VCHAR(11,20,152)
1730 CALL HCHAR(8,27,144)
1740 BY=10
1750 GOTO 980
1760 REM  HOLE 7
1770 CALL HCHAR(2,4,120,12)
1780 CALL HCHAR(12,4,120,17)
1790 CALL HCHAR(6,15,120,16)
1800 CALL HCHAR(20,20,120,11
)
1810 CALL VCHAR(2,4,120,10)
1820 CALL VCHAR(2,15,120,4)
1830 CALL VCHAR(12,20,120,8)
1840 CALL VCHAR(6,30,120,14)
1850 FOR I=1 TO 6
1860 CALL HCHAR(5+I,5,112,I+
1)
1870 NEXT I
1880 CALL HCHAR(7,22,152,8)
1890 CALL HCHAR(8,22,152,8)
1900 FOR I=1 TO 8
1910 CALL HCHAR(8+I,27,152,3
)
1920 NEXT I
1930 CALL HCHAR(4,9,144)
1940 BY=24
1950 GOTO 980
1960 REM  HOLE 8
1970 CALL HCHAR(2,4,120,27)
1980 CALL HCHAR(15,4,120,10)
1990 CALL HCHAR(11,13,120,8)
2000 CALL HCHAR(20,21,120,10
)
2010 CALL VCHAR(2,4,120,14)
2020 CALL VCHAR(11,13,120,4)
2030 CALL VCHAR(11,21,120,10
)
2040 CALL VCHAR(2,30,120,18)
2050 CALL VCHAR(5,7,128,4)
2060 CALL VCHAR(6,8,128,5)
2070 CALL VCHAR(5,9,128,6)
2080 CALL VCHAR(5,10,128,4)
2090 CALL VCHAR(5,11,128,3)
2100 FOR I=22 TO 29
```

```
2110 READ J
2120 CALL VCHAR(3,I,152,J)
2130 NEXT I
2140 DATA 3,4,5,6,9,10,13,14
2150 CALL VCHAR(3,21,152)
2160 CALL VCHAR(14,27,152)
2170 CALL VCHAR(13,8,144)
2180 BY=25
2190 GOTO 980
2200 REM  HOLE 9
2210 CALL HCHAR(2,4,120,27)
2220 CALL HCHAR(11,4,120,16)
2230 CALL HCHAR(20,20,120,10
)
2240 CALL VCHAR(2,4,120,10)
2250 CALL VCHAR(2,30,120,19)
2260 CALL VCHAR(11,20,120,10
)
2270 CALL VCHAR(5,13,120,4)
2280 CALL HCHAR(5,8,120,5)
2290 FOR I=1 TO 9
2300 CALL HCHAR(2+I,20+I,152
,10-I)
2310 NEXT I
2320 CALL HCHAR(8,9,144)
2330 BY=25
2340 GOTO 980
2350 REM
2360 REM ***PLAY***
2370 REM
2380 FOR HO=1 TO 9
2390 CALL CLEAR
2400 BC=104
2410 ON HO GOSUB 910,1010,11
50,1300,1470,1580,1760,1960,
2200
2420 FOR I=1 TO 11
2430 CALL HCHAR(21,2+I,ASC(S
EG$("HOLE NUMBER",I,1)))
2440 NEXT I
2450 CALL HCHAR(21,15,ASC(ST
R$(HO)))
2460 FOR I=1 TO 3
2470 CALL HCHAR(22,5+I,ASC(S
EG$("PAR",I,1)))
2480 NEXT I
2490 CALL HCHAR(22,10,ASC(ST
R$(HA(HO,5))))
2500 FOR I=1 TO 6
2510 CALL HCHAR(23,2+I,ASC(S
EG$("TRAPS:",I,1)))
```

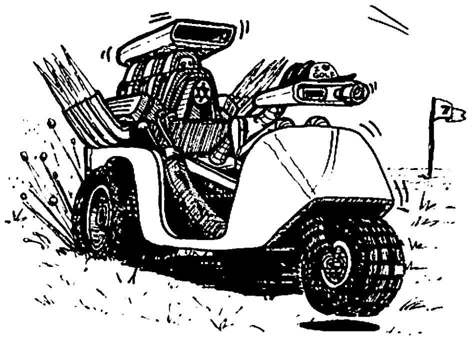
```
2520 NEXT I
2530 AMT=2
2540 I=1
2550 IF HA(HO,I)=0 THEN 2580
2560 AMT=AMT+7
2570 ON I GOSUB 4210,4260,42
80,4300
2580 I=I+1
2590 IF I<5 THEN 2550
2600 A$="DIRECTION (1-8):"
2610 GOSUB 2740
2620 CALL KEY(3,KEY,STAT)
2630 IF STAT=0 THEN 2620
2640 CALL HCHAR(24,19,KEY)
2650 FOR I=1 TO 100
2660 NEXT I
2670 KEY=KEY-48
2680 IF (KEY<1)+(KEY>8)+(KEY
<>INT(KEY))THEN 2690 ELSE 28
20
2690 CALL HCHAR(24,3,32,18)
2700 A$="THE DIRECTION IS FR
OM 1 TO 8"
2710 GOSUB 2740
2720 GOSUB 2780
2730 GOTO 2600
2740 FOR I=1 TO LEN(A$)
2750 CALL HCHAR(24,2+I,ASC(S
EG$(A$,I,1)))
2760 NEXT I
2770 RETURN
2780 FOR I=1 TO 500
2790 NEXT I
2800 CALL HCHAR(24,1,32,32)
2810 RETURN
2820 DIR=KEY
2830 CALL HCHAR(24,2,32,18)
2840 A$="SPEED (0-5):"
2850 GOSUB 2740
2860 GOSUB 4320
2870 IF (KEY<0)+(KEY>5)THEN
2880 ELSE 2930
2880 CALL HCHAR(24,2,32,18)
2890 A$="THE SPEED IS FROM 0
.0 TO 5.0"
2900 GOSUB 2740
2910 GOSUB 2780
2920 GOTO 2840
2930 CALL HCHAR(24,2,32,18)
2940 SP=KEY
2950 UF=0
```

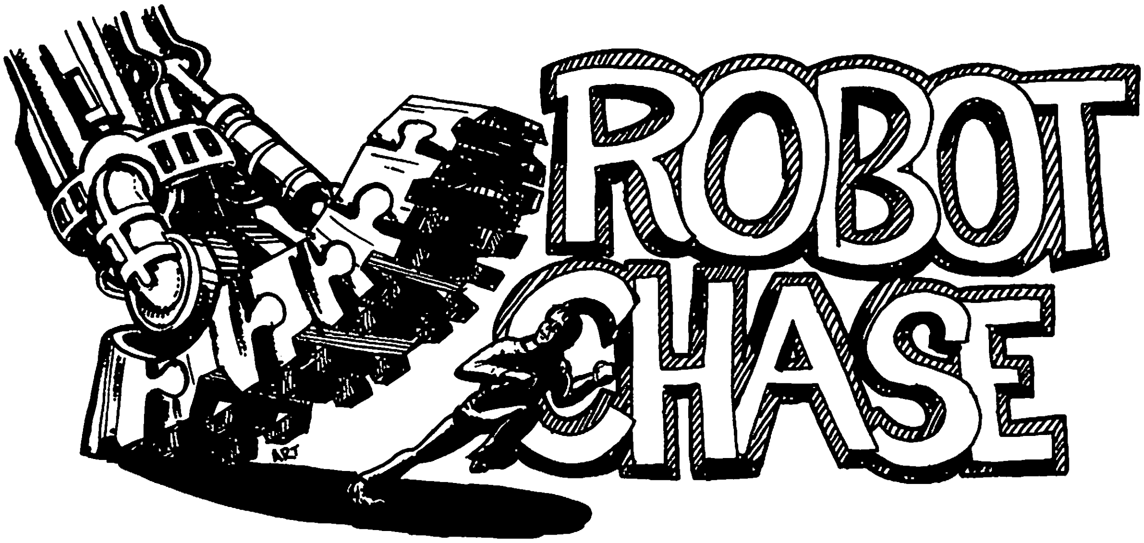
```
2960 REM  CLEAR UNEVEN FLAG

2970 TF=0
2980 REM  CLEAR TRAP FLAG

2990 X2=BX+DI(DIR,1)
3000 Y2=BY+DI(DIR,2)
3010 CALL GCHAR(X2,Y2,CK)
3020 IF (CK=32)+(CK=104)THEN
 3030 ELSE 3060
3030 GOSUB 3740
3040 BC=104
3050 GOTO 4120
3060 IF CK=120 THEN 3110
3070 IF CK=112 THEN 3670
3080 IF CK=128 THEN 3800
3090 IF CK=152 THEN 3830
3100 IF CK=144 THEN 3940
3110 IF (DIR=1)+(DIR=3)THEN
3120 ELSE 3140
3120 DIR=DIR+4
3130 GOTO 2990
3140 IF (DIR=7)+(DIR=5)THEN
3150 ELSE 3170
3150 DIR=DIR-4
3160 GOTO 2990
3170 CALL GCHAR(X2+1,Y2,CK1)
3180 CALL GCHAR(X2,Y2+1,CK)
3190 CALL GCHAR(X2,Y2-1,CK3)
3200 CALL GCHAR(X2-1,Y2,CK2)
3210 ON DIR/2 GOTO 3220,3340
,3450,3560
3220 IF (CK=120)*(CK1=120)TH
EN 3240 ELSE 3260
3230 IF TF<=0 THEN 4150
3240 DIR=6
3250 GOTO 2990
3260 IF CK<>120 THEN 3290
3270 DIR=4
3280 GOTO 2990
3290 IF CK1<>120 THEN 3320
3300 DIR=8
3310 GOTO 2990
3320 DIR=6
3330 GOTO 2990
3340 IF (CK=120)*(CK2=120)TH
EN 3350 ELSE 3370
3350 DIR=8
3360 GOTO 2990
3370 IF CK<>120 THEN 3400
3380 DIR=2
```

```
3390 GOTO 2990
3400 IF CK2<>120 THEN 3430
3410 DIR=6
3420 GOTO 2990
3430 DIR=8
3440 GOTO 2990
3450 IF (CK3=120)*(CK2=120)T
HEN 3460 ELSE 3480
3460 DIR=2
3470 GOTO 2990
3480 IF CK3<>120 THEN 3510
3490 DIR=8
3500 GOTO 2990
3510 IF CK2<>120 THEN 3540
3520 DIR=4
3530 GOTO 2990
3540 DIR=2
3550 GOTO 2990
3560 IF (CK3=120)*(CK1=120)T
HEN 3570 ELSE 3590
3570 DIR=4
3580 GOTO 2990
3590 IF CK3<>120 THEN 3620
3600 DIR=6
3610 GOTO 2990
3620 IF CK1<>120 THEN 3650
3630 DIR=2
3640 GOTO 2990
3650 DIR=4
3660 GOTO 2990
3670 GOSUB 3700
3680 BC=112
3690 GOTO 4120
3700 IF TF THEN 3740
3710 TF=3
3720 SC=SC+1
3730 REM  TRAP FLAG
3740 CLR=BC
3750 CALL HCHAR(BX,BY,CLR)
3760 CALL HCHAR(X2,Y2,136)
3770 BX=X2
3780 BY=Y2
3790 RETURN
3800 GOSUB 3700
3810 BC=128
3820 GOTO 4120
3830 IF UF THEN 3910
3840 UF=1
3850 REM  UNEVEN FLAG
3860 DIR=DIR+INT(RND*2)*2-1
3870 IF DIR THEN 3890
```

```
3880 DIR=8
3890 IF DIR<>9 THEN 3910
3900 DIR=1
3910 GOSUB 3740
3920 BC=152
3930 GOTO 4120
3940 CALL HCHAR(BX,BY,104)
3950 CALL HCHAR(X2,Y2,136)
3960 CALL HCHAR(X2,Y2,144)
3970 DIR=DIR+INT(RND*2)*2-1
3980 IF DIR THEN 4000
3990 DIR=8
4000 SP=SP-.4
4010 IF SP<=0 THEN 4050
4020 X2=X2+DI(DIR,1)
4030 Y2=Y2+DI(DIR,2)
4040 GOTO 3010
4050 IF DIR<>9 THEN 4070
4060 DIR=1
4070 FOR I=1 TO 20
4080 CALL SOUND(50,1760,2)
4090 NEXT I
4100 SC=SC+1
4110 GOTO 4190
4120 CALL SOUND(50,1760,2)
4130 TF=TF-1
4140 IF TF=0 THEN 4170
4150 SP=SP-.2
4160 IF SP>0 THEN 2990
4170 SC=SC+1
4180 GOTO 2420
4190 NEXT HO
4200 RETURN
4210 H$="WATER"
4220 FOR I=1 TO LEN(H$)
4230 CALL HCHAR(23,AMT+I,ASC
(SEG$(H$,I,1)))
4240 NEXT I
4250 RETURN
4260 H$="SAND"
4270 GOTO 4220
4280 H$="UNEVEN"
4290 GOTO 4220
4300 H$="BLOCKS"
4310 GOTO 4220
4320 REM  SPEED
4330 CALL KEY(3,L,STAT)
4340 IF (STAT=0)+(STAT=-1)TH
EN 4330
4350 IF L<>46 THEN 4380
4360 L=0
```

```
4370 GOTO 4430
4380 CALL HCHAR(24,16,L)
4390 L=L-48
4400 CALL KEY(3,M,STAT1)
4410 IF (STAT1=0)+(STAT1=-1)
THEN 4400
4420 IF M=13 THEN 4490
4430 CALL HCHAR(24,17,46)
4440 CALL KEY(3,R,STAT2)
4450 IF (STAT2=0)+(STAT2=-1)
THEN 4440
4460 CALL HCHAR(24,18,R)
4470 R=R-48
4480 R=R*.1
4490 KEY=L+R
4500 FOR I=1 TO 100
4510 NEXT I
4520 RETURN
4530 REM
4540 REM ***END***
4550 REM
4560 GOSUB 450
4570 FOR I=1 TO 8
4580 PRINT
4590 CALL SOUND(200,I*110,2)
4600 NEXT I
4610 PRINT "THE GAME IS OVER
!!!"
4620 PRINT
4630 PRINT "ON THE PAR 27 CO
URSE, YOU   SHOT A ROUND OF
";SC
4640 PRINT
4650 PRINT "THAT IS AN AVERA
GE OF";SC/9;"SHOTS PER HOLE.
"
4660 PRINT
4670 PRINT "HOPE YOU ENJOYED
 THE GAME!!!"
4680 RETURN
```





ROBOT CHASE

In this game you try to avoid being captured by the killer robots, but escaping their relentless pursuit is most difficult. Because the pursuit is entirely predetermined, it might be a good idea to chart your course before venturing forth.

Look at lines 940 through 1030. These lines draw the original game setup. To test this out, erase lines within the loop to see the effect of their omissions. In keeping with good programming practice, the variable names are representative. RO is symbolic of the RObots; YX and YY are Y)our X and Y)our Y coordinates.

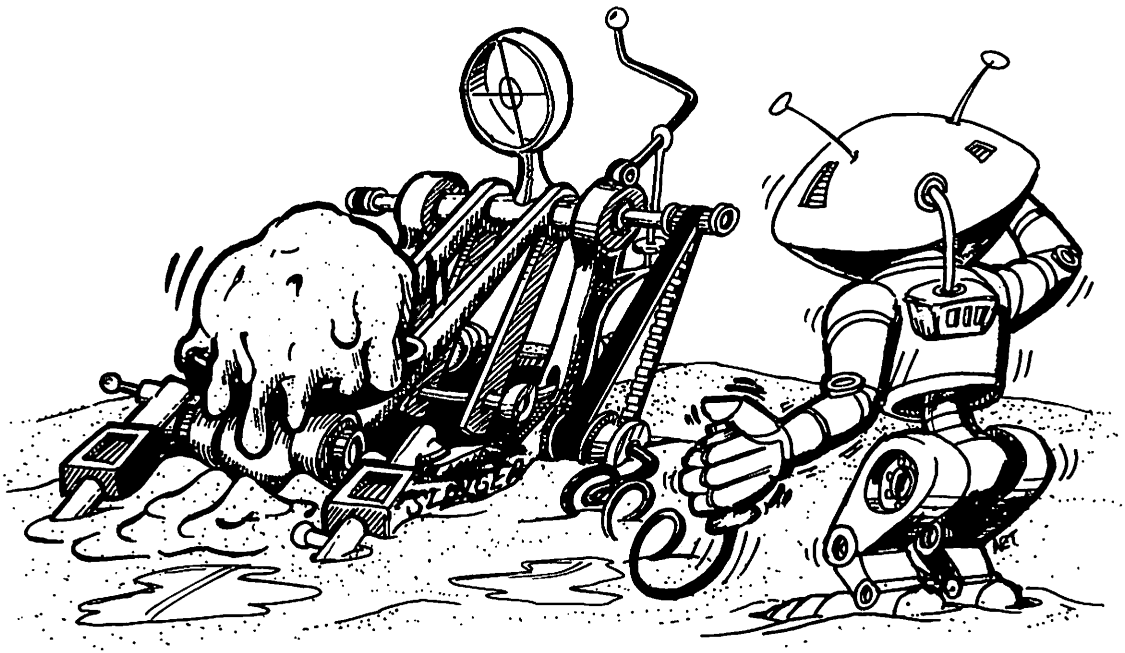
```
10 REM *******************
20 REM ***             ***
30 REM *** ROBOT CHASE ***
40 REM ***             ***
50 REM *******************
60 GOSUB 110
70 GOSUB 480
80 GOSUB 1090
90 GOSUB 1620
100 END
110 REM
120 REM ***INSTRUCTIONS***
130 REM
140 GOSUB 440
150 PRINT "IN ROBOT CHASE, Y
OU ARE AN  EXPLORER WHO HAS
LANDED HIS SPACESHIP ON A HO
STILE      PLANET."
160 PRINT
170 PRINT "SEVERAL PROTECTOR
 ROBOTS ARETRYING TO CAPTURE
 YOU. IF   YOU CAN REACH A B
ASE. YOU"
180 PRINT "WILL BE SAFE BEHI
ND ITS     PROTECTIVE FORCE
FIELD."
190 FOR I=1 TO 10
200 PRINT
210 NEXT I
220 GOSUB 420
230 GOSUB 440
240 PRINT "HERE'S HOW THINGS
 WORK:"
250 PRINT
260 PRINT "BLUE  - AN EXPLOS
IVE FENCE          (BAD!)"
270 PRINT
280 PRINT "WHITE - YOU"
290 PRINT
300 PRINT "GREEN - ATTACKING
 ROBOT             (BAD!)"
310 PRINT
320 PRINT "RED   - PROTECTIV
E BASE             (GOOD!)"
330 PRINT
340 PRINT " 3 2 1"
350 PRINT "  \!/  THIS IS YO
UR CHOICE"
360 PRINT " 4-+-8 OF MOVEMEN
T THROUGH"
370 PRINT "  /!\  THE MAZE"
380 PRINT " 5 6 7"
```

```
390 FOR I=1 TO 2
400 PRINT
410 NEXT I
420 INPUT "PRESS ENTER WHEN
READY TO   CONTINUE: ":ANS$
430 RETURN
440 CALL CLEAR
450 PRINT TAB(7);"*** ROBOT
CHASE ***"
460 PRINT
470 RETURN
480 REM
490 REM ***SETUP***
500 CALL CLEAR
510 CALL SCREEN(15)
520 RANDOMIZE
530 DIM DI(8,2),FI(21,11),RO
(9,2)
540 FF$="FFFFFFFFFFFFFFFF"
550 CALL CHAR(120,FF$)
560 CALL CHAR(128,FF$)
570 CALL CHAR(136,FF$)
580 CALL CHAR(144,FF$)
590 CALL CHAR(152,FF$)
600 CALL COLOR(12,15,15)
610 CALL COLOR(13,5,15)
620 CALL COLOR(14,7,15)
630 CALL COLOR(15,13,15)
640 CALL COLOR(16,16,15)
650 DEF R(X)=INT((RND)*X)+1
660 NR=4+R(5)
670 NO=INT((NR-4)/2)+1
680 FOR I=1 TO NR
690 RX=R(20)
700 RY=R(10)
710 IF FI(RX,RY)THEN 690
720 FI(RX,RY)=144
730 RO(I,1)=RX
740 RO(I,2)=RY
750 NEXT I
760 FOR I=1 TO NO
770 OX=R(20)
780 OY=R(10)
790 IF FI(OX,OY)THEN 770
800 FI(OX,OY)=136
810 NEXT I
820 YX=R(20)
830 YY=R(10)
840 IF FI(YX,YY)THEN 820
850 FI(YX,YY)=152
860 FOR I=0 TO 21
```

```
870 FI(I,0)=128
880 FI(I,11)=128
890 NEXT I
900 FOR I=0 TO 11
910 FI(0,I)=128
920 FI(21,I)=128
930 NEXT I
940 CALL HCHAR(3,7,128,22)
950 CALL VCHAR(4,7,128,10)
960 CALL HCHAR(14,7,128,22)
970 CALL VCHAR(4,28,128,10)
980 FOR I=1 TO 20
990 FOR J=1 TO 10
1000 IF FI(I,J)=0 THEN 1020
1010 CALL HCHAR(J+3,I+7,FI(I
,J))
1020 NEXT J
1030 NEXT I
1040 FOR I=1 TO 8
1050 READ DI(I,1),DI(I,2)
1060 NEXT I
1070 DATA 1,-1,0,-1,-1,-1,-1
,0,-1,1,0,1,1,1,1,0
1080 RETURN
1090 REM
1100 REM ***PLAY***
1110 REM
1120 GOSUB 1760
1130 CALL HCHAR(23,28,128)
1140 CALL SOUND(80,880,0)
1150 CALL KEY(3,N,STAT)
1160 IF (STAT=0)+(STAT=-1)TH
EN 1150
1170 ANS=N-48
1180 IF (ANS<1)+(ANS>8)THEN
1130
1190 CALL HCHAR(23,28,N)
1200 X2=YX+DI(ANS,1)
1210 Y2=YY+DI(ANS,2)
1220 CALL HCHAR(YY+3,YX+7,12
0)
1230 CALL HCHAR(Y2+3,X2+7,15
2)
1240 IF FI(X2,Y2)<>128 THEN
1280
1250 WL=0
1260 RETURN
1270 REM ***FENCE***
1280 IF FI(X2,Y2)<>144 THEN
1320
1290 WL=0
```

```
1300 RETURN
1310 REM ***ROBOT***
1320 IF FI(X2,Y2)<>136 THEN
1360
1330 WL=1
1340 RETURN
1350 REM ***BASE***
1360 FI(YX,YY)=120
1370 YX=X2
1380 YY=Y2
1390 FI(YX,YY)=152
1400 FOR I=1 TO NR
1410 IF R(4)<>1 THEN 1450
1420 X2=R(3)-2
1430 Y2=R(3)-2
1440 GOTO 1470
1450 X2=SGN(YX-RO(I,1))
1460 Y2=SGN(YY-RO(I,2))
1470 X2=X2+RO(I,1)
1480 Y2=Y2+RO(I,2)
1490 IF (FI(X2,Y2)=128)+(FI(
X2,Y2)=144)+(FI(X2,Y2)=136)T
HEN 1410
1500 CALL HCHAR(RO(I,2)+3,RO
(I,1)+7,120)
1510 CALL HCHAR(Y2+3,X2+7,14
4)
1520 IF FI(X2,Y2)<>152 THEN
1560
1530 WL=0
1540 RETURN
1550 REM ***HUMAN***
1560 FI(RO(I,1),RO(I,2))=120
1570 RO(I,1)=X2
1580 RO(I,2)=Y2
1590 FI(RO(I,1),RO(I,2))=144
1600 NEXT I
1610 GOTO 1130
1620 REM
1630 REM ***END***
1640 REM
1650 CALL CLEAR
1660 PRINT "THE GAME IS OVER
!!!!"
1670 PRINT
1680 IF WL=0 THEN 1710
1690 PRINT "YOU'VE BEATEN TH
E KILLER    ROBOTS (YEAH!!)"
1700 GOTO 1720
1710 PRINT "THE KILLER ROBOT
S GOT YOU   (SORRY..)"
```

```
1720 FOR K=1 TO 10
1730 CALL SOUND(100,1760,2)
1740 NEXT K
1750 RETURN
1760 REM
1770 REM ***DIRECTIONS***
1780 REM
1790 CALL HCHAR(20,14,32,7)
1800 CALL HCHAR(21,14,32,7)
1810 CALL HCHAR(22,14,32,7)
1820 FOR I=1 TO 5
1830 CALL HCHAR(20,I+14,ASC(
SEG$("3 2 1",I,1)))
1840 CALL HCHAR(21,I+14,ASC(
SEG$("4 + 8",I,1)))
1850 CALL HCHAR(22,I+14,ASC(
SEG$("5 6 7",I,1)))
1860 NEXT I
1870 FOR I=1 TO 20
1880 CALL HCHAR(23,I+7,ASC(S
EG$("WHICH DIRECTION ===>",I
,1)))
1890 NEXT I
1900 RETURN
```



Schmoo

The whimsical name and the humorous objective of the game serve to camouflage an excellent thinking-man's game. It is hard to imagine how a person could play and not come away with a better understanding of the X,Y coordinate system. The object is to splat a mudball on the mud-loving Schmoo. The elevation at which you aim the mudball slinger determines how far the mudball will travel. The angle at which you should shoot is determined by the coordinates where the Schmoo is located. The following is a list of coordinates and the angles they represent.

X	Y	ANGLE
0	12588	0
17866	17866	45
30910	0	90
5888	-5888	135
0	-9400	180
-25727	-25727	225
-18992	0	270
-34101	34101	315

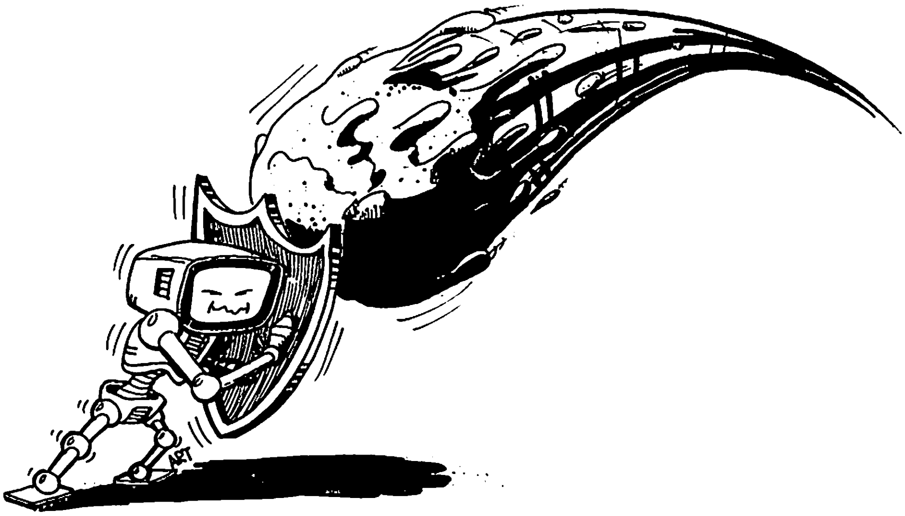
This chart should help you to understand how various coordinates relate to angles.

```
10 REM ********************
20 REM ***              ***
30 REM ***    SCHMOO    ***
40 REM ***              ***
50 REM ********************
60 GOSUB 110
70 GOSUB 520
80 GOSUB 620
90 GOSUB 1050
100 END
110 REM
120 REM ***INSTRUCTIONS***
130 REM
140 CALL SCREEN(15)
150 GOSUB 480
160 PRINT "THIS IS THE GAME
OF SCHMOO. IN IT YOU THROW M
UD AROUND  IN HOPES OF HITTI
NG THE MUD-LOVING SCHMOO."
170 PRINT
180 PRINT "YOU ARE SITUATED
IN THE      CENTER OF AN X,Y
COORDINATE SYSTEM; AT POSITI
ON 0,0. THE"
190 PRINT "SCHMOO WILL BE LO
CATED       SOMEWHERE ON THE
SAME PLANE.HIS COORDINATES A
RE GIVEN TO"
200 PRINT "YOU BEFORE EACH T
URN."
210 FOR I=1 TO 8
220 PRINT
230 NEXT I
240 GOSUB 460
250 GOSUB 480
260 PRINT "YOU HAVE YOUR TRU
STY AUTO-  MATIC MUDBALL SLI
NGER WHICH YOU USE TO TOSS M
UDBALLS AT"
270 PRINT "THE SCHMOO. YOU E
NTER THE   ELEVATION AND THE
 ANGLE AT  WHICH YOU WISH TO
 FIRE THE"
280 PRINT "MUDBALL. AFTER EA
CH SHOT YOUARE GIVEN THE COO
RDINATES   WHERE THE MUDBALL
 LANDED."
290 PRINT
300 PRINT "FOR EXAMPLE, IF T
HE SCHMOO'SCOORDINATES ARE (
-500,1000) THEN THE SCHMOO I
S ABOUT"
```

```
310 PRINT "FIVE HUNDRED FEET
 TO YOUR   LEFT AND ABOUT A
THOUSAND   FEET IN FRONT OF
YOU."
320 FOR I=1 TO 4
330 PRINT
340 NEXT I
350 GOSUB 460
360 GOSUB 480
370 PRINT "THE ELEVATION FOR
 THE SHOT  WOULD BE ABOUT 89
.3 DEGREES WHILE THE ANGLE W
HERE THE"
380 PRINT "SCHMOO CAN BE FOU
ND IS ABOUT333.4 DEGREES."
390 PRINT
400 PRINT "THE MUDBALLS ARE
BIG ENOUGH TO MUDDY THE SCHM
OO AS LONG AS THEY LAND WITH
IN 100 FEETOF HIM."
410 PRINT
420 PRINT "NOW THAT YOU KNOW
 HOW TO    MAKE THE SCHMOO H
APPY, GO   GET HIM. GOOD LUC
K!!!."
430 FOR I=1 TO 6
440 PRINT
450 NEXT I
460 INPUT "PRESS ENTER WHEN
READY TO   CONTINUE: ":ANS$
470 RETURN
480 CALL CLEAR
490 PRINT TAB(9);"*** SCHMOO
 ***"
500 PRINT
510 RETURN
520 REM
530 REM ***SETUP***
540 REM
550 RANDOMIZE
560 S1=INT(RND*2)*2-1
570 S2=INT(RND*2)*2-1
580 SX=(INT(RND*26000)+5000)
*S1
590 SY=(INT(RND*26000)+5000)
*S2
600 CN=3.1415926357989/180
610 RETURN
620 REM
630 REM ***PLAY***
640 REM
```

```
650 GOSUB 480
660 FOR I=1 TO 3
670 PRINT
680 NEXT I
690 CALL SOUND(80,440,0)
700 PRINT "THE SCHMOO IS AT
              COORDINATES:"
710 PRINT "(";SX;",";SY;")"
720 FOR I=1 TO 6
730 PRINT
740 NEXT I
750 PRINT "WHAT ELEVATION FO
R THE"
760 INPUT "MUDBALL SLINGER(1
-90)?:":EL
770 PRINT
780 IF (EL>=1)*(EL<=90)THEN
810
790 PRINT "THE ELEVATION RAN
GES FROM    1 TO 90..."
800 GOTO 750
810 IF EL<>90 THEN 850
820 PRINT "THAT WOULD SHOOT
THE MUD    STRAIGHT UP... AN
D IT WOULD COME DOWN ON YOU!
!"
830 PRINT
840 GOTO 750
850 PRINT "WHAT ANGLE OF DIR
ECTION      FOR THE MUDBALL S
LINGER?"
860 INPUT "(0-360):":AN
870 IF (AN>=0)+(AN<=360)THEN
 910
880 PRINT "THE ANGLES RANGE
FROM        0 TO 360."
890 PRINT
900 GOTO 850
910 DM=ABS(INT(93000*SIN(EL*
CN)*COS(EL*CN)))
920 XM=DM*SIN(AN*CN)
930 YM=DM*COS(AN*CN)
940 DS=SQR((SX-XM)^2+(SY-YM)
^2)
950 PRINT
960 PRINT "THE MUD SPLATTERE
D AT        COORDINATES:"
970 PRINT "(";INT(XM);",";IN
T(YM);")"
980 PRINT
990 TRY=TRY+1
```

```
1000 IF DS>100 THEN 700
1010 PRINT
1020 CALL SOUND(800,262,0,33
0,0,524,0)
1030 PRINT "THAT'S GOOD ENOU
GH TO        SPLAT THE SCHMOO
!!!"
1040 RETURN
1050 REM
1060 REM ***END***
1070 REM
1080 PRINT
1090 PRINT
1100 PRINT "YOU SPLATTERED T
HE SCHMOO IN ";TRY;" TRIES."
1110 PRINT
1120 PRINT
1130 PRINT "THANKS FOR PLAYI
NG SCHMOO!!!"
1140 FOR I=1 TO 5
1150 PRINT
1160 NEXT I
1170 RETURN
```



Grab a friend and get ready for some heated competition. The object of this game is to wall-in your opponent and to prevent him from moving. On each turn, you enter the coordinates of an adjacent square into which you will move, plus, you enter the coordinates of a square you wish to be henceforth uninhabitable. The first player unable to move loses the game.

The graphics are fairly easy to explain. The exceptions, lines 480-510, are hexadecimal print formatting commands. Don't concern yourself with them. The CALL COLOR statements are often the key to unlocking the secret of CALL HCHAR (or VCHAR). Look at line 660. In English, the instruction reads: print at row 10 column 10 character 136 (a special character for user purposes). From the paucity of information given, you can determine the function of this line! How? The tip-off comes from 136. Character 136 is part of Character Set #14 (check your manual). Line 530 sets all members of Character Set #14 to have foreground color 7 (dark red) on a background of color 3 (medium green). Type in: 665 STOP. Run the program. Can you spot a red on green figure? How about a solid dark red square? Line 490 eliminated any background from character 136 (which, granted, you wouldn't know); therefore, the solid red square was drawn by line 660. You are on your way to becoming a computer sleuth. Use this same technique to pinpoint the function of 670. Hint: Check lines 500 and 540.

```
10 REM ********************
20 REM ***              ***
30 REM ***   STRANDED   ***
40 REM ***              ***
50 REM ********************
60 GOSUB 110
70 GOSUB 330
80 GOSUB 730
90 GOSUB 1410
100 END
110 REM
120 REM ***INSTRUCTIONS***
130 REM
140 CALL SCREEN(15)
150 GOSUB 290
160 PRINT "THIS IS A GAME FO
R TWO      PLAYERS. BOTH OF
YOU WILL BEPLACED IN AN 8 BY
 8 MATRIX."
170 PRINT "A PLAYER MAY MOVE
 IN ANY OF EIGHT DIRECTIONS.
 ENTER THE COORDINATES OF TH
E TARGET   SQUARE."
180 PRINT
190 PRINT "AFTER YOU MOVE YO
U WILL BE  ASKED THE COORDIN
ATES OF THESECOND SQUARE. TH
IS SQUARE"
200 PRINT "WILL THEN BE BLAC
KED OUT ANDBE UNENTERABLE."
210 PRINT
220 PRINT "THE FIRST PLAYER
UNABLE TO  MOVE LOSES THE GA
ME. THE TOPLEFT OF THE BOARD
 IS (1,1),"
230 PRINT "AND THE TOP RIGHT
 IS (1,8)."
240 FOR I=1 TO 2
250 PRINT
260 NEXT I
270 INPUT "PRESS ENTER WHEN
READY TO   CONTINUE: ":ANS$
280 RETURN
290 CALL CLEAR
300 PRINT TAB(6);"*** STRAND
ED ***"
310 PRINT
320 RETURN
330 REM
340 REM ***SETUP***
350 REM
```

```
360 CALL CLEAR
370 CALL SCREEN(3)
380 DIM DIR(8,2),M(8,8),P(2,
3)
390 DEF CNVX(X)=X+3+(X-1)
400 DEF CNVY(X)=X+9+(X-1)
410 DEF FNP(X)=(X-1)*8+136
420 M(4,1)=1
430 M(5,8)=2
440 P(1,1)=4
450 P(1,2)=1
460 P(2,2)=5
470 P(2,3)=8
480 CALL CHAR(128,"FFFFFFFFF
FFFFFFF")
490 CALL CHAR(136,"FFFFFFFFF
FFFFFFF")
500 CALL CHAR(144,"FFFFFFFFF
FFFFFFF")
510 CALL CHAR(152,"FFFFFFFFF
FFFFFFF")
520 CALL COLOR(13,16,3)
530 CALL COLOR(14,7,3)
540 CALL COLOR(15,6,3)
550 CALL COLOR(16,2,3)
560 FOR I=3 TO 20 STEP 2
570 CALL HCHAR(I,9,128,16)
580 NEXT I
590 FOR I=3 TO 20 STEP 2
600 CALL VCHAR(3,I+6,128,17)
610 NEXT I
620 FOR I=1 TO 8
630 CALL HCHAR(2,I+9+(I-1),4
8+I)
640 CALL HCHAR(I+3+(I-1),8,4
8+I)
650 NEXT I
660 CALL HCHAR(10,10,136)
670 CALL HCHAR(12,24,144)
680 FOR I=1 TO 8
690 READ DIR(I,1),DIR(I,2)
700 NEXT I
710 DATA 1,1,-1,-1,1,-1,-1,1
,1,0,0,1,-1,0,0,-1
720 RETURN
730 REM
740 REM ***PLAY***
750 REM
760 FOR PL=1 TO 2
770 CALL HCHAR(22,20,32)
780 CALL HCHAR(24,15,32,3)
```

```
790 CALL HCHAR(22,10,32,15)
800 CALL HCHAR(22,18,ASC(STR
$(PL)))
810 GOSUB 1470
820 CALL KEY(3,R,STATUS)
830 IF STATUS=0 THEN 820
840 CALL HCHAR(24,15,R)
850 CALL HCHAR(24,16,44)
860 CALL KEY(3,C,STATUS)
870 IF (STATUS=-1)+(STATUS=0
)THEN 860
880 CALL HCHAR(24,17,C)
890 X1=R-48
900 Y1=C-48
910 IF (X1<1)+(X1>8)+(X1<>IN
T(X1))+(Y1<1)+(Y1>8)+(Y1<>IN
T(Y1))THEN 960
920 X=P(PL,PL)
930 Y=P(PL,PL+1)
940 IF SQR(((X1-X)^2)+((Y1-Y
)^2))>=2 THEN 960
950 IF M(X1,Y1)=0 THEN 980
960 GOSUB 1670
970 GOTO 820
980 M(X,Y)=0
990 XX=CNVX(X)
1000 YY=CNVY(Y)
1010 CALL HCHAR(XX,YY,13)
1020 M(X1,Y1)=PL
1030 CO=FNP(PL)
1040 P(PL,PL)=X1
1050 P(PL,PL+1)=Y1
1060 X=X1
1070 Y=Y1
1080 XX=CNVX(X)
1090 YY=CNVY(Y)
1100 CALL HCHAR(XX,YY,CO)
1110 GOSUB 1570
1120 CALL KEY(3,R,STATUS)
1130 IF STATUS=0 THEN 1120
1140 CALL HCHAR(24,15,R)
1150 CALL HCHAR(24,16,44)
1160 CALL KEY(3,C,STATUS)
1170 IF (STATUS=0)+(STATUS=-
1)THEN 1160
1180 CALL HCHAR(24,17,C)
1190 X2=R-48
1200 Y2=C-48
1210 IF (X2<1)+(X2>8)+(Y2<1)
+(Y2>8)+(X2<>INT(X2))+(Y2<>I
NT(Y2))THEN 1230
```

```
1220 IF M(X2,Y2)=0 THEN 1250
1230 GOSUB 1670
1240 GOTO 1120
1250 M(X2,Y2)=3
1260 X=X2
1270 Y=Y2
1280 XX=CNVX(X)
1290 YY=CNVY(Y)
1300 CALL HCHAR(XX,YY,152)
1310 K=3-PL
1320 FOR J=1 TO 8
1330 X4=DIR(J,1)+P(K,K)
1340 Y4=DIR(J,2)+P(K,K+1)
1350 IF (X4<1)+(X4>8)+(Y4<1)
+(Y4>8)THEN 1370
1360 IF M(X4,Y4)=0 THEN 1390
1370 NEXT J
1380 RETURN
1390 NEXT PL
1400 GOTO 760
1410 REM
1420 REM  ***END***
1430 REM
1440 PRINT "THE GAME IS OVER
....."
1450 PRINT "PLAYER NUMBER ";
PL;"IS THE WINNER!!!"
1460 RETURN
1470 REM
1480 REM ***INSTR***
1490 REM
1500 FOR I=1 TO 7
1510 CALL HCHAR(22,I+10,ASC(
SEG$("PLAYER ",I,1)))
1520 NEXT I
1530 FOR I=1 TO 21
1540 CALL HCHAR(23,I+4,ASC(S
EG$("MOVE TO (ROW,COLUMN):",
I,1)))
1550 NEXT I
1560 RETURN
1570 REM
1580 REM ***BLOCK***
1590 REM
1600 CALL HCHAR(24,15,32,3)
1610 CALL HCHAR(22,10,32,13)
1620 CALL HCHAR(23,5,32,25)
1630 FOR I=1 TO 9
1640 CALL HCHAR(22,I+10,ASC(
SEG$("BLOCK AT:",I,1)))
1650 NEXT I
```

```
1660 RETURN
1670 REM
1680 REM ***BAD MOVE***
1690 REM
1700 CALL SOUND(1000,110,2)
1710 CALL HCHAR(24,15,32,3)
1720 RETURN
```

This game requires good timing. A small, moving target is the object of your marksman talent. The angle of the launch pad will determine when it is time to shoot.

All of the graphics for the gameboard are drawn in the 280 subroutine. Line 330 sets the SCREEN color to deep blue. Lines 440 through 470 draw the black gameboard perimeter. This can be verified by first adding line 439 STOP, running the program, then replacing 439 with 475 GOTO 475. When you run the program after the second change, the program will hit the indefinite loop (at 475) after drawing the black border. Lines 610 through 650 draw the launching pad in a symmetrical configuration, at a RaNDom location, utilizing a five dot design. You can verify this by employing the same technique as used above. To escape from an indefinite loop, press FCTN-4 (CLEAR), then eliminate the line.

Line 750 draws the target. Line 710 draws over the target in deep blue. This "blacks out" the target so that each time line 750 is encountered, the target appears to have moved. To better understand this, change line 710 to read: CALL VCHAR(X3,Y3,126,2). What you are doing is changing 152 (an ASCII code in Character Set #16) to 126 (an ASCII code in Character Set #12). Line 430 sets all characters in Character Set #16 to be printed in color 6 (deep blue) on a background of color 6. Line 390 sets all characters in Character Set #12 to be printed in color 11 (dark yellow) on a color 6 background. The reason why the target is solid yellow is because of line 340 — which is beyond the scope of this book.

```
10 REM ********************
20 REM ***              ***
30 REM ***    TARGET    ***
40 REM ***              ***
50 REM ********************
60 GOSUB 90
70 GOSUB 280
80 END
90 REM
100 REM ***INSTRUCTIONS***
110 REM
120 GOSUB 240
130 PRINT "IN THIS GAME YOU
TRY TO HIT A MOVING TARGET.
BY PRESSINGANY KEY, A SMALL
BALL WILL"
140 PRINT "FIRE FROM THE PAD
DLE ON THE SCREEN."
150 PRINT
160 PRINT "YOU MUST TIME THE
 RELEASE   SUCH THAT THE SMA
LL ONE HITSTHE LARGER ONE."
170 PRINT
180 PRINT "THE DIRECTION AND
 DISTANCE  WILL VARY WITH EA
CH TARGET."
190 FOR I=1 TO 8
200 PRINT
210 NEXT I
220 INPUT "PRESS ENTER WHEN
READY TO   CONTINUE: ":ANS$
230 RETURN
240 CALL CLEAR
250 PRINT TAB(9);"*** TARGET
 ***"
260 PRINT
270 RETURN
280 REM
290 REM ***PLAY***
300 REM
310 RANDOMIZE
320 CALL CLEAR
330 CALL SCREEN(6)
340 CALL CHAR(126,"FFFFFFFFF
FFFFFFF")
350 CALL CHAR(128,"FFFFFFFFF
FFFFFFF")
360 CALL CHAR(136,"3C7EFFFFF
FFF7E3C")
370 CALL CHAR(144,"FFFFFFFFF
FFFFFFF")
```

```
380 CALL CHAR(152,"FFFFFFFFF
FFFFFFF")
390 CALL COLOR(12,11,6)
400 CALL COLOR(13,2,6)
410 CALL COLOR(14,16,6)
420 CALL COLOR(15,9,6)
430 CALL COLOR(16,6,6)
440 CALL HCHAR(1,2,128,31)
450 CALL VCHAR(1,2,128,23)
460 CALL VCHAR(1,32,128,23)
470 CALL HCHAR(23,2,128,31)
480 FOR I=1 TO 10
490 CALL HCHAR(24,2+I,ASC(SE
G$("SCORE:    ",I,1)))
500 CALL HCHAR(24,12+I,ASC(S
EG$("SHOTS:    ",I,1)))
510 CALL HCHAR(24,22+I,ASC(S
EG$("PCENT:   %",I,1)))
520 NEXT I
530 IF SH=0 THEN 550
540 PC=INT(SC/SH*100)
550 PM=INT(RND*2)*2-1
560 XP=INT(RND*5)+10
570 YP=INT(RND*5)+10
580 SP=INT(RND*2)*PM
590 FLAG=0
600 GOSUB 1200
610 CALL HCHAR(XP+2*SP,YP-2,
144)
620 CALL HCHAR(XP+SP,YP-1,14
4)
630 CALL HCHAR(XP,YP,144)
640 CALL HCHAR(XP-SP,YP+1,14
4)
650 CALL HCHAR(XP-2*SP,YP+2,
144)
660 X2=INT(RND*10)+17
670 Y2=2
680 S2=INT(RND*2)+1
690 X3=X2
700 Y3=Y2
710 CALL VCHAR(Y3,X3,152,2)
720 CALL GCHAR(Y2,X2,CK)
730 CALL GCHAR(Y2+1,X2,CK1)
740 IF (CK=136)+(CK1=136)THE
N 1120
750 CALL VCHAR(Y2,X2,126,2)
760 X3=X2
770 Y3=Y2
780 IF FLAG THEN 910
790 CALL KEY(3,KEY,STAT)
```

```
800 IF STAT=0 THEN 1150
810 FLAG=1
820 B1=YP+3
830 B2=XP
840 B3=B1
850 B4=B2
860 SH=SH+1
870 PC=INT(SC/SH*100)
880 GOSUB 1200
890 CALL GCHAR(B4,B3,CK)
900 IF (B2<>B4)*(CK<>136)THE
N 1120
910 CALL GCHAR(B4+1,B3,CK)
920 IF CK=126 THEN 1120
930 CALL HCHAR(B4,B3,152)
940 CALL GCHAR(B2,B1,CK)
950 IF CK=126 THEN 1120
960 CALL HCHAR(B2,B1,136)
970 B3=B1
980 B4=B2
990 B1=B1+1
1000 B2=B2+SP
1010 IF (B2<2)+(B2>22)THEN 1
020 ELSE 1050
1020 B2=B4
1030 SP=-SP
1040 GOTO 1150
1050 IF B1<=31 THEN 1110
1060 CALL HCHAR(B4,B3,152)
1070 FLAG=0
1080 SP=ABS(SP)*PM
1090 GOTO 1150
1100 CALL GCHAR(B2+1,B1,CK)
1110 IF CK<>126 THEN 1150
1120 SC=SC+1
1130 CALL CLEAR
1140 GOTO 440
1150 Y2=Y2+1
1160 IF Y2<=21 THEN 1180
1170 Y2=2
1180 GOTO 710
1190 RETURN
1200 REM
1210 REM ***SCORE***
1220 REM
1230 X=INT(SC/10)
1240 TMP=X*10
1250 Y=SC-TMP
1260 CALL HCHAR(24,9,ASC(STR
$(X)))
1270 CALL HCHAR(24,10,ASC(ST
R$(Y)))
```

```
1280 REM
1290 REM ***SHOTS***
1300 REM
1310 X=INT(SH/10)
1320 TMP=X*10
1330 Y=SH-TMP
1340 CALL HCHAR(24,19,ASC(ST
R$(X)))
1350 CALL HCHAR(24,20,ASC(ST
R$(Y)))
1360 REM
1370 REM ***PERCENT***
1380 REM
1390 IF PC>99 THEN 1460
1400 X=INT(PC/10)
1410 TMP=X*10
1420 Y=PC-TMP
1430 CALL HCHAR(24,30,ASC(ST
R$(X)))
1440 CALL HCHAR(24,31,ASC(ST
R$(Y)))
1450 GOTO 1490
1460 CALL HCHAR(24,29,49)
1470 CALL HCHAR(24,30,48)
1480 CALL HCHAR(24,31,48)
1490 RETURN
```

In this game, the object is to avoid the pursuit of the horrible Twinky and to escape from the danger filled labyrinth. There are a plethora of obstacles which impede your escape. There are twenty squares which relocate you somewhere else in the maze. There are twenty impenetrable squares. In addition, there is one square which contains a highly sensitive exploding device. If you move onto this space, the ensuing blast will end your perilous journey . . . and your life. You are given a zap gun to repel the Twinky, but do not get complacent. The Twinky recovers from the blast quickly, and renews the pursuit.

```
10 REM ********************
20 REM ***              ***
30 REM ***    TWINKY    ***
40 REM ***              ***
50 REM ********************
60 GOSUB 100
70 GOSUB 550
80 GOSUB 1000
90 END
100 REM
110 REM ***INSTRUCTIONS***
120 REM
130 GOSUB 510
140 PRINT "THIS IS THE GAME
OF TWINKY. IN IT YOU PRETEND
 TO BE A   SPACE EXPLORER WH
O HAS"
150 PRINT "LANDED ON A HOSTI
LE PLANET."
160 PRINT
170 PRINT "CAPTURED BY THE U
NFRIENDLY  NATIVES, YOU ARE
TOSSED INTOA LARGE PRISON AL
ONG WITH A"
180 PRINT "FEROCIOUS TWINKY."
190 PRINT
200 PRINT "A TWINKY IS A HOR
RIBLE      CREATURE THAT WIL
L CATCH YOUAND ABSORB YOUR B
ODY INTO"
210 PRINT "HIS IF HE GETS CL
OSER THAN  TWO UNITS AWAY FR
OM YOU."
220 FOR I=1 TO 5
230 PRINT
240 NEXT I
250 GOSUB 490
260 GOSUB 510
270 PRINT "IN THE INTEREST O
F FAIR PLAYYOU ARE GIVEN A Z
AP GUN THATWILL TEMPORARILY
CHASE THE  TWINKY AWAY."
280 PRINT
290 PRINT "ALSO, IF YOU CAN
MAKE IT TO THE SPECIAL OBJEC
TIVE SQUAREBEFORE BEING ABSO
RBED, YOU"
300 PRINT "WILL BE SET FREE."
310 PRINT
320 PRINT "AFTER YOU MOVE, Y
OU WILL BE INFORMED OF YOUR
```

```
DISTANCE   FROM THE OBJECTIV
E SQUARE AS"
330 PRINT "WELL AS FROM THE
TWINKY."
340 FOR I=1 TO 6
350 PRINT
360 NEXT I
370 GOSUB 490
380 GOSUB 510
390 PRINT "THERE ARE TWENTY
RELOCATION SQUARES. THESE SQ
UARES SEND YOU TO SOME OTHER
 SECTION OFTHE MAZE."
400 PRINT
410 PRINT "THERE ARE TWENTY
IMPREGNABLESQUARES WHICH YOU
 CANNOT    ENTER."
420 PRINT
430 PRINT "THERE IS ONE SUPE
R DEADLY   AUTOMATIC KILL SQ
UARE WHICH ENDS YOUR ORDEAL
QUICKLY ANDPAINLESSLY."
440 PRINT
450 PRINT "THAT'S IT...TRY T
O ENJOY    IT!!!"
460 FOR I=1 TO 4
470 PRINT
480 NEXT I
490 INPUT "PRESS ENTER WHEN
READY TO   CONTINUE: ":ANS$
500 RETURN
510 CALL CLEAR
520 PRINT TAB(10);"***TWINKY
***"
530 PRINT
540 RETURN
550 REM
560 REM ***SETUP***
570 REM
580 CALL CLEAR
590 RANDOMIZE
600 DIM MA(15,15)
610 DEF FNR(X)=INT(RND*X)+1
620 DEF FNA(X)=.001*INT(X*10
00+.5)
630 FOR I=1 TO 20
640 X=FNR(15)
650 Y=FNR(15)
660 IF MA(X,Y)THEN 640
670 MA(X,Y)=1
680 NEXT I
```

```
690 REM  BLOCKED
700 FOR I=1 TO 20
710 X=FNR(15)
720 Y=FNR(15)
730 IF MA(X,Y)THEN 710
740 MA(X,Y)=2
750 NEXT I
760 REM  RELOCATION
770 X=FNR(15)
780 Y=FNR(15)
790 IF MA(X,Y)THEN 770
800 MA(X,Y)=3
810 REM  SUPER KILL
820 XO=FNR(15)
830 YO=FNR(15)
840 IF MA(XO,YO)THEN 830
850 MA(XO,YO)=4
860 REM  OBJECTIVE
870 XT=FNR(15)
880 YT=FNR(15)
890 IF MA(XT,YT)THEN 870
900 MA(XT,YT)=5
910 REM  TWINKY
920 XP=FNR(15)
930 YP=FNR(15)
940 IF MA(XP,YP)THEN 920
950 MA(XP,YP)=6
960 REM  PLAYER
970 ST=0
980 SP=0
990 RETURN
1000 REM
1010 REM ***PLAY***
1020 REM
1030 GOSUB 510
1040 GOSUB 1060
1050 GOTO 1130
1060 DT=FNA(SQR((XT-XP)^2+(Y
T-YP)^2))
1070 DO=FNA(SQR((XP-XO)^2+(Y
P-YO)^2))
1080 PRINT "THE TWINKY IS";D
T;"UNITS AWAY"
1090 PRINT
1100 PRINT "THE OBJECTIVE IS
";DO;"UNITS AWAY"
1110 PRINT
1120 RETURN
1130 IF DT>=2 THEN 1160
1140 GOSUB 2530
1150 RETURN
```

```
1160 INPUT "MOVE OR SHOOT...
TYPE IN     (M) OR (S):";ANS
$
1170 PRINT
1180 IF ANS$="S" THEN 1910
1190 IF ANS$<>"M" THEN 1160
1200 GOSUB 1220
1210 GOTO 1410
1220 PRINT "FORWARD, BACKWAR
D"
1230 INPUT "RIGHT OR LEFT...
TYPE IN     (F),(B),(R) OR (
L):";ANS$
1240 PRINT
1250 IF ANS$="B" THEN 1320
1260 IF ANS$="R" THEN 1350
1270 IF ANS$="L" THEN 1380
1280 IF ANS$<>"F" THEN 1220
1290 X=0
1300 Y=-1
1310 GOTO 1400
1320 X=0
1330 Y=1
1340 GOTO 1400
1350 X=1
1360 Y=0
1370 GOTO 1400
1380 X=-1
1390 Y=0
1400 RETURN
1410 X=X+XP
1420 Y=Y+YP
1430 IF X<1 THEN 1470
1440 IF X>15 THEN 1470
1450 IF Y<1 THEN 1470
1460 IF Y<=15 THEN 1490
1470 PRINT "THAT WOULD TAKE
YOU OUT OF  THE MAZE...MOVE
NOT ALLOWED."
1480 GOTO 2230
1490 IF MA(X,Y)<>1 THEN 1530
1500 PRINT "THAT SPACE IS BL
OCKED..."
1510 PRINT "MOVE NOT ALLOWED
."
1520 GOTO 2230
1530 IF MA(X,Y)<>2 THEN 1590
1540 PRINT "YOU'VE BEEN RELO
CATED"
1550 X=FNR(15)
1560 Y=FNR(15)
```

```
1570 IF MA(X,Y)=1 THEN 1550
1580 GOTO 1530
1590 IF MA(X,Y)<>3 THEN 1680
1600 PRINT "YOU FOUND THE SU
PER KILL    SQUARE!!!!"
1610 PRINT "MOVE ALLOWED...B
UT"
1620 PRINT "YOU'VE BEEN KILL
ED!!!!"
1630 WL=1
1640 FOR I=1 TO 7
1650 CALL SOUND(100,494,2,24
7,2,494,2)
1660 NEXT I
1670 RETURN
1680 IF MA(X,Y)<>4 THEN 1790
1690 PRINT "YOU FOUND THE OB
JECTIVE!!!!"
1700 PRINT
1710 PRINT "MOVE ALLOWED"
1720 PRINT
1730 PRINT "YOU WIN A TRIP O
FF THIS     PLANET!!!!"
1740 WL=0
1750 FOR I=1 TO 7
1760 CALL SOUND(100,494,2,24
7,2,494,2)
1770 NEXT I
1780 RETURN
1790 IF MA(X,Y)<>5 THEN 1840
1800 PRINT "MOVE ALLOWED."
1810 PRINT
1820 GOSUB 2530
1830 RETURN
1840 PRINT "MOVE ALLOWED."
1850 MA(XP,YP)=SP
1860 XP=X
1870 YP=Y
1880 SP=MA(XP,YP)
1890 MA(XP,YP)=6
1900 GOTO 2230
1910 GOSUB 1220
1920 SX=XP
1930 SY=YP
1940 SX=SX+X
1950 SY=SY+Y
1960 PRINT "ZAP--"
1970 CALL SOUND(50,1760,10)
1980 IF SX<1 THEN 2020
1990 IF SX>15 THEN 2020
2000 IF SY<1 THEN 2020
```

```
2010 IF SY<=15 THEN 2060
2020 PRINT "FIZZLE.. THE SHO
T LEFT THE  SCREEN."
2030 PRINT
2040 PRINT "THE SHOT MISSED
THE TWINKY!!"
2050 GOTO 2230
2060 IF MA(SX,SY)=0 THEN 194
0
2070 IF MA(SX,SY)=2 THEN 194
0
2080 IF MA(SX,SY)=3 THEN 194
0
2090 IF MA(SX,SY)=4 THEN 194
0
2100 IF MA(SX,SY)<>1 THEN 21
50
2110 PRINT "BLAST!!!"
2120 PRINT "THE SHOT HIT A W
ALL."
2130 PRINT "THE SHOT MISSED.
"
2140 GOTO 2230
2150 PRINT "OUCH!!!"
2160 PRINT "THE SHOT HIT THE
 TWINKY."
2170 PRINT "THE TWINKY RETRE
ATS"
2180 MA(SX,SY)=ST
2190 XT=FNR(15)
2200 YT=FNR(15)
2210 ST=MA(XT,YT)
2220 MA(XT,YT)=5
2230 REM  TWINKY MOVE LOGIC

2240 GOSUB 1060
2250 PRINT "THE TWINKY MOVES
..."
2260 FOR I=1 TO 500
2270 NEXT I
2280 PRINT
2290 IF DT>=2 THEN 2320
2300 GOSUB 2530
2310 RETURN
2320 IF XP>XT THEN 2360
2330 X=-1
2340 Y=0
2350 GOTO 2470
2360 IF XP<XT THEN 2400
2370 X=1
2380 Y=0
```

```
2390 GOTO 2470
2400 IF YP>YT THEN 2440
2410 X=0
2420 Y=-1
2430 GOTO 2470
2440 IF YP<YT THEN 2470
2450 X=0
2460 Y=1
2470 MA(XT,YT)=ST
2480 XT=XT+X
2490 YT=YT+Y
2500 ST=MA(XT,YT)
2510 MA(XT,YT)=5
2520 GOTO 1040
2530 PRINT "<<< S C H L O O
R P!!!! >>>"
2540 PRINT "YOU'VE BEEN ABSO
RBED BY THE TWINKY..."
2550 PRINT
2560 PRINT "YOU LOSE."
2570 WL=1
2580 CALL SOUND(100,1760,2,1
661,2,1568,2)
2590 CALL SOUND(100,1480,2,1
397,2,1319,2)
2600 CALL SOUND(100,1245,2,1
175,2,1109,2)
2610 CALL SOUND(100,1047,2,9
88,2,932,2)
2620 CALL SOUND(100,880,2)
2630 CALL SOUND(4000,110,10)
2640 RETURN
```

GAMES TIs PLAY

ALL THE GAMES IN THIS BOOK ON TWO DISKS

!!! Save hours of typing and de-bugging!!!
$24.95 for two disks with this coupon.
Available ONLY to readers of this book.

- * Spend your time playing the games and writing your own versions.
- * Use the sound routines and graphics in your own programs.
- * Games may be copied, listed, and modified.
- * Let your fingers do the playing, not the typing.

Please RUSH me the GAMES TIs PLAY diskettes for $24.95
Please include $1.75 for shipping and handling
California residents add 6.5% tax ($1.62)

NAME ______________________________

ADDRESS ______________________________

CITY ______________ STATE __________ ZIP ________

PHONE (   ) ______________________________

☐ MASTER ☐ VISA ☐ CHECK

CREDIT CARD NUMBER ______________ EXPIRES __________

SIGNATURE (for charge cards) ______________________

Send to: DATAMOST™
8943 Fullbright Avenue
Chatsworth, CA 91311-2750
(818) 709-1202

Other Great Books for the TI-99/4A from DATAMOST:

KIDS AND THE TI-99/4A *by Edward H. Carlson.*
The most popular introduction to computers ever published. Thirty-three step-by-step lessons introduce the reader painlessly to the wonderful world of computers. Each lesson includes introductory notes for the teacher or parent as well as review questions for self-testing at the end of each chapter. Cleverly illustrated with dozens of cartoons. *Not just for Kids* — also suitable for the Adult embarking on computer literacy!
$19.95

COMPUTER PLAYGROUND ON THE TI-99/4A *by M.J. Winter.*
Based on a child's natural interest in words, games and graphics, this book is a collection of BASIC computer activities for the TI-99/4A. Each activity is presented as a "problem" with questions to lead the child to the correct answer. Children will find the problems interesting, fun and most importantly — geared to their level of understanding. They'll run sample programs, learn how to modify them, and complete partially written programs. Everything is kept simple — the child is led step-by-step through the programming process. Illustrated with over 70 color cartoons and pictures to color. It's BASIC programming playtime for children grades 2 through 6.
$9.95

THE ELEMENTARY TI-99/4A *by William B. Sanders.*
Ten chapters lead you step-by-step through the process of hooking up the computer, loading and saving programs, creating graphics, music, and all kinds of handy utilities. Everything is made simple so by the time you're finished, you'll be writing and using programs. Even if you're already programming, this book has lots of helpful information and will satisfy the entire family's desire to participate in the computer revolution! Dozens of delightful cartoons.
$14.95

Available at your local computer dealer or bookstore
OR
Order directly from

DATAMOST™
8943 Fullbright Ave.
Chatsworth, CA 91311-2750
(818) 709-1202

WHAT BETTER WAY TO LEARN PROGRAMMING THAN BY PLAYING GAMES?

The authors have compiled a selection of classic computer games along with many clever new games. Written in TI BASIC, they are formatted in a way which lets you easily adapt any of the routines to your own programs. Each game is explained in simple, straightforward terms — allowing you to modify and customize to your heart's content!

You'll learn principles of text formatting, word games, data statements, and input routines. Many of the games will help you understand how grids are constructed and graphics animated. Helpful techniques are given so you can dissect BASIC programs and learn what makes them tick.

You'll have hours of fun learning to program BASIC in this easy, enjoyable way. And when you're done, you'll have the games to play! Whoever thought that gaming could teach so much about programming and de-bugging?

0 48831 00120

ISBN 0-88190-120-2

DATAMOST™ INC

8943 Fullbright Avenue, Chatsworth, CA 91311-2750
(818) 709-1202