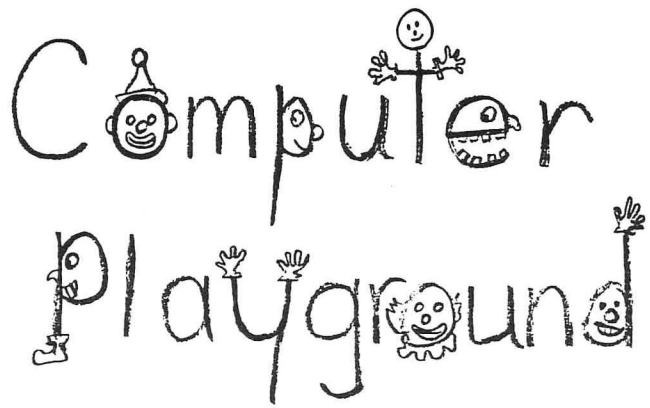
by
M. J.
Winter

Computer Playground™

$9.95

Atari 400/800/1200

DATAMOST


Computer
Playground

Computer Playground

On the Atari

M.J. Winter, Ph.D.
Professor of Mathematics
Michigan State University

Illustrated by
Martin Cannon

DATAMOST™
8943 Fullbright Avenue
Chatsworth, CA 91311-2750
(213) 709-1202

DATAMOST™
ISBN 0-88190-190-3

Acknowledgements

My thanks go to all those who encouraged me to write something for younger children. Special appreciation is due to:

Dave Gordon and his very creative and supportive staff at DATAMOST.

My fellow instructors at the MSU summer computer camps: Ed Carlson, John Forsythe and Mark Lardie. Double thanks to Ed with whom I wrote materials for the camp, and who encouraged this project.

The children in the first classes - especially Amy, Linda, Harriet, Claire, Ida, Elizabeth, Beth, Susie, Leah, Jason, Courtney, Jessica, Erin and Jon.

All the teachers who've responded to my articles and programs, describing the enthusiasm of their students.

My family, who not only live with my three computers in the kitchen, but who liberally gave me suggestions and encouragement.

The 6-year old, now 8, who started this project off when she complained:

I don't want to just play games.
Show me how to make the computer DO something.

THE PROBLEMS

Introduction

This book is for the child who wants to make the computer "do" something. It is based on the perception that such a child is probably not thinking of evaluating arithmetic expressions, but rather of making the computer respond to his/her directions. A child who knows some BASIC will be able to "talk" to the computer and control its actions.

The BASIC taught in this book will enable children to make the computer "do" things that interest them. The method of presentation is that of "conversational BASIC": each idea is taught by the combination of an interesting format and lots of repetition.

The book is intended for the child in grades 2.5–7 whose family has an ATARI computer, or for children who have access to a classroom computer. While the programs presented are specifically designed to work on the ATARI, this is not a book on "*What You Can Do with Your ATARI.*" Rather, it is a book on how to use BASIC with an ATARI. While many of the special ATARI features are included, they are used only as illustrations or applications of BASIC words.

Each activity is presented as a "Problem" in a workbook. Children will find the problems both interesting and geared to their level of understanding. They will feel satisfaction in being able to fill in the blanks in the workbook. As the book is completed, a stock of successful programs is accumulated. Frequently a child will spend long periods of time repeating and varying a single activity.

As a child progresses through the workbook, she/he will learn how to use many features of BASIC. Because the commands have been made meaningful, the children will have established a firm base of knowledge. As they grow older, they will be able to extend their knowledge and apply it to more difficult, and more "mathematical" problems.

Overview

Children are able to mimic before they can create. In some of the problems they are directed to run sample programs, then modify them. In others, the children are to complete partially written programs.

Most young children will need a HELPER available (parent, teacher, older child). The helper should read the COMMENTS AND DESCRIPTIONS section to see what is required, and what explanations are suggested. Older children may need occasional assistance.

The BASIC vocabulary of each group of problems is given below. See the COMMENTS AND DESCRIPTIONS section for more details.

Group	Problems	BASIC Vocabulary
1	1 – 5	PRINT, GOTO, END
2	6 – 7	INPUT, DIM A$(20)
3	9 – 15	FOR . . . NEXT
4	16 – 20	IF . . . THEN
5	21 – 24	POKE, PEEK
6	25 – 28	A$(3), A$(1,3), LEN(A$)
7	29 – 32	arithmetic operations
8	33 – 34	SOUND, STEP -1
9	35 – 40	RND(1)

Comments and Descriptions

Problems 1 - 5

The first problems teach the words PRINT and GOTO, as well as the purpose of line numbers.

1. *SEASON'S GREETINGS.* The program prints an endless number of greetings. To stop it, press the BREAK key. PRINT by itself makes the computer skip a line.

2. *TAKE A NUMBER, PLEASE.* The line numbers attached to each statement determine the order in which the computer goes through the program. Even if the lines are entered in mixed-up order, the computer knows which should come next. Type LIST to see the lines in their correct order.

3. *MISTAKES.* Every time RETURN is pressed, the ATARI checks the line. If it finds a mistake, it prints an error message. However, your mistakes might make sense to the computer. In that case you'll find something wrong when you try to run the program. If the computer prints a Ø (zero) where you expected a word, you probably forgot the quotation marks.

4. *INITIALS.* The example makes a large P from small ones. To design a slanted initial, like A or W, it may be helpful to use a grid.

5. *WHERE NOW?.* A puzzle which emphasizes that the computer will always go to the next higher line number if it isn't told to go somewhere else. END is necessary to stop the program without printing extra letters. A semicolon at the end of a PRINT statement keeps the letters all printed on the same row.

Problems 6 – 7

Most children have seen a computer ask "what's your name?". The INPUT statement makes the computer stop until a word is typed and RETURN pressed. The ATARI must know ahead of time how many letters each word is allowed to have. DIM A$(2Ø) says that the word A$ will have no more than 2Ø letters.

6. *WALLPAPER.* The notation (A) means hold down the CTRL key while typing the A key. The computer prepares a "box" labeled P$. When the pattern is entered, the computer stores it in the box. Every time the program needs the pattern, it goes to the box and looks inside. (Demonstrate with a labeled box and a piece of paper with a pattern on it.)

7. *LETTER TO SANTA.* (Young children will object on the grounds that they're too old for this.) Four words are used in this program. A matching exercise has the child connect the name on each "box" with what it contains.

Problems 8 – 15

The next group is based on the FOR ... NEXT instructions. The first instance of FOR . . . NEXT is to make the computer pause. In other applications it does something each time it counts. Cursor controls are used in graphics applications of loops and delays.

8. *CURSOR CONTROLS.* This problem shows how to make your program clear the screen, move the cursor and print reversed characters. *E* represents the ESC key.

9. *PAUSING FOR BREATH.* The computer prints a line, then pauses while it counts to 5000, then prints another line. Use a stopwatch or watch to time how fast it can count.

10. *ROCKETS.* The cursor controls, graphics characters and the pause are combined to draw rockets shooting up into the blue.

11. *COUNTING.* Watch the computer while it counts. How much longer does it take to count to 5000 when the numbers are printed? Finally, a reaction test checks on your reflexes.

12. *TOUCH YOUR TOES. . . . .* Make the computer repeat an action while it counts. It can print a message, skip lines or draw the sides of a box.

13. *MOVERS.* A double application of FOR ... NEXT. One FOR ... NEXT is used to repeat the printing of the star; the other is used to slow down the printing. Movement is really an illusion. If a star is printed, erased and printed again one space further along, it will appear to move.

14. *THE WORM.* The same as #13, but this time a worm crawls across the screen. To erase the tail, it's necessary to back up six spaces.

15. *MORE COUNTING.* A review of FOR...NEXT. This time the upper limit is an INPUT. It's not necessary to prepare a special box for numbers; the computer does that automatically.

Problems 16 – 20

These problems introduce the idea of testing, i.e., comparing two things. Testing is done with IF something is true THEN do something. The equals sign, =, means "is the same as"; the "<>" means "is not the same as." Testing is done first with words, then with numbers.

16. *TESTING.* Simple and complicated statements can be tested. IF A$="RED" OR A$="BLUE" is an example. THEN is followed by an action, in this case PRINT.

17. *GOLDILOCKS.* Complete the program by testing the temperature of the porridge.

18. *ANIMAL GUESS.* This program uses IF...THEN with branching. Younger children will probably be content with changing line 100 every time. Others may want to make this a two-person game.

19. *LITTLE INDIANS.* A short program combines FOR...NEXT with IF...THEN to print out the counting rhyme. This is more complicated than it appears as the child must supply the upper limit, the test and the final message.

20. *NURSERY RHYME.* The problem is to write a program which prints out a rhyme like that in #19.

Problems 21 – 24

These problems use POKE, PEEK and SETCOLOR. PEEK is used to look at what's in a certain memory location. POKE is used to change what's in the location. SETCOLOR is used to change the color of the screen, the border and the letters. The locations contain information about the color of the screen, the color of the border, the presence or absence of the cursor and the name of the last key touched. All this information is in special codes.

21. *BLINKING LIZARDS.* Make the orphan blink her eyes. To make only the eyes change, we move the cursor and print on top of them. POS.2,0 puts the cursor back in the upper left corner. POKE 752,1 will make the cursor disappear.

22. *SCREEN COLORS.* Experiment with different colors for the screen and border by using SETCOLOR.

23. *PEEKABOO.* While a program is running, the computer constantly checks the keyboard to see what key is being pressed. If it's the BREAK key, the program will stop. Each time a new key is pressed, the code number for that key goes into location 764. To learn the code numbers for the keys, PEEK into 764. Even though we take our hands away, the number in 764 doesn't change until a different key is pressed.

24. *FUNNYBONES. . . .* Use the code numbers found in #23 to make the computer respond differently to certain keys.

Problems 25 – 28

These problems are concerned with words and parts of words. A$(3) and A$(1,3) are defined, as well as LEN(A$).

25. *QUIZ TIME.* While no new BASIC concepts are introduced, this is more complicated than any of the earlier programs. All the answers are stored in the A$ box. Answers are checked; sometimes a wrong answer produces another chance.

26. *PIECES OF WORDS.* An investigation into A$(3), A$(1,3), LEN(A$), etc. Attention is restricted to the first N letters of a word, or all *but* the first N letters.

27. *RHYMING GAME.* The string functions A$(1,1) and LEN(A$) are used to give hints to the guesser. Children will enjoy thinking of tricky rhymes that aren't given away by the two hints.

28. *WORD PUZZLE.* Using the beginnings and endings of a word, print out other words. This should be done with a pencil and then checked.

Problems 29 – 32

Very simple arithmetic operations on variables are introduced in these problems. Experience has shown that manipulating variables generally requires a high level of mental development and the ability to handle abstractions. The operations in these problems are kept as concrete as possible.

29. *DOING MATH.* The operation is doubling. The * is the multiplication sign on the computer. Change the operation to tripling. Have older children change the program to dividing by 2, doubling and adding 3, etc.

30. *MATH QUIZ.* Two values are input. The user is asked to add them; the computer checks the answer. An extension would be to have a wrong answer cause the problem to be printed again. All children can change this to a multiplication quiz.

31. *TIMES TABLES.* First the computer prints multiples of 2. Then labels are added so that the output looks like a multiplication table. Younger children may have difficulty understanding the elaborate PRINT statement, but they will be delighted with the output and will be able to change it.

32. *DAYS AND NIGHTS.* Use the computer to solve the problem of how much we sleep in a year if we sleep eight hours a night. Several variables are used, but their names are highly mnemonic.

Problems 33 – 34

SOUND and more complicated FOR ... NEXT loops are demonstrated in these problems. SOUND is limited to varying the pitch; volume can be adjusted on the TV set. STEP −1 is used to count backwards.

33. *SOUNDING OFF.* After experimenting with single notes, a FOR N = 1Ø TO 4Ø loop produces a downhill sequence. Finally, the keyboard codes are used to generate notes. Try typing various words to see how they sound. Slow typists should increase the 5Ø in line 1Ø.

34. *COUNTDOWN.* Simple demonstrations of how to count from 5 down to 1. Two FOR...NEXT loops will let you count up and then down again.

Problems 35 – 40

The last six problems all require the computer to "think" of a number. This is done by using the random function RND(1). The expression

```
INT(RND(1)*6+1)
```

must be used to obtain an integer from 1 to 6.

Older (10 and up) children may understand the explanation:

`RND(1)`	is a (decimal) number between 0 and 1
`RND(1)*6`	is a (decimal) number between 0 and 6
`RND(1)*6+1`	is a (decimal) number between 1 and 7
`INT`	means take the whole number part, so
`INT(RND(1)*6+1)`	is one of: 1,2,3,4,5,6

35. *THINK OF A NUMBER.* How to use the formula to roll a die. The results of six rolls are recorded.

36. *TWO DICE.* Every time the computer has to think of a number from 1 to 6, the expression INT(RND(1)*6+1) must be used. The lines of this program are used in the next problem, so don't NEW it.

37. *GET OUT OF JAIL.* The classical reason for rolling two dice. The program uses FOR...NEXT for the three rolls, IF...THEN for the test. Remind the child that "=" means "is the same as."

38. *NUMBER GUESS.* The computer thinks of a number and the user tries to guess it. Messages of TOO BIG and TOO SMALL are printed. A counter is added to count how many guesses are used. Compare it with the highway counters: every time a car drives over it, it increases the counter.

39. *ANSWER MACHINE.* Like a crystal ball, the computer can see into the future. After being asked a question, the computer randomly selects one of its supply of answers.

40. *SOUND AND LIGHT SHOW.* The grande finale! When completed, this program will play random notes while flashing random screen colors; the rhythm is also random. A spectacular finish to the problems.

PROBLEM 1 SEASON'S GREETINGS

A. Enter this program — that means after typing each line, you press RETURN — and RUN it.

```
10 PRINT "HAPPY"
20 PRINT "THANKSGIVING"
```

Copy what happened _____

_____

B. Enter two more lines:

```
30 PRINT
40 GOTO 10
```

and RUN the program again. To stop it, press the BREAK key.

What happened this time? _____

_____

_____

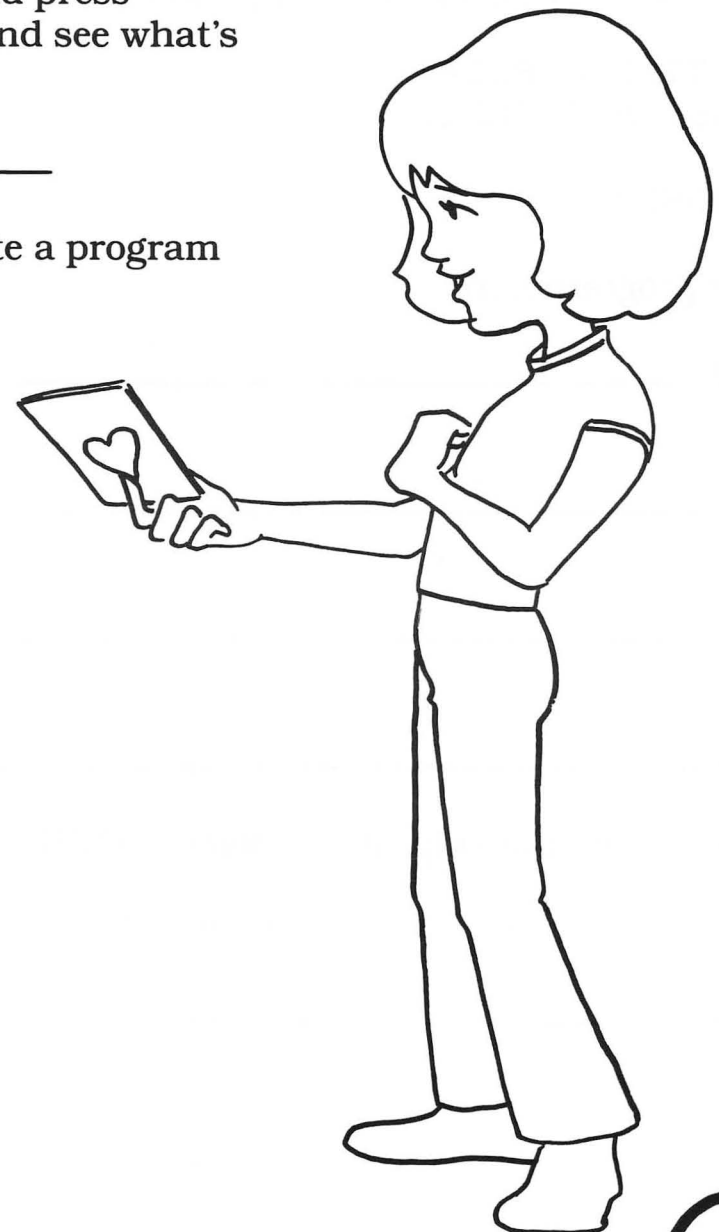
_____

What did line 30 do? (If you're not sure, type 30 and press RETURN to erase it; then run the program again and see what's different.)

_____

C. Enter NEW to erase this program. Now you write a program that prints

```
BE MY
VALENTINE

BE MY
VALENTINE

etc.
```



Copy your program here:

10 _____

20 _____

30 _____

40 _____

D. Enter NEW. Write a program that prints

```
LITTLE BOPEEP
LOST HER SHEEP

BOOHOO
```

Copy your program here:

10 _____

20 _____

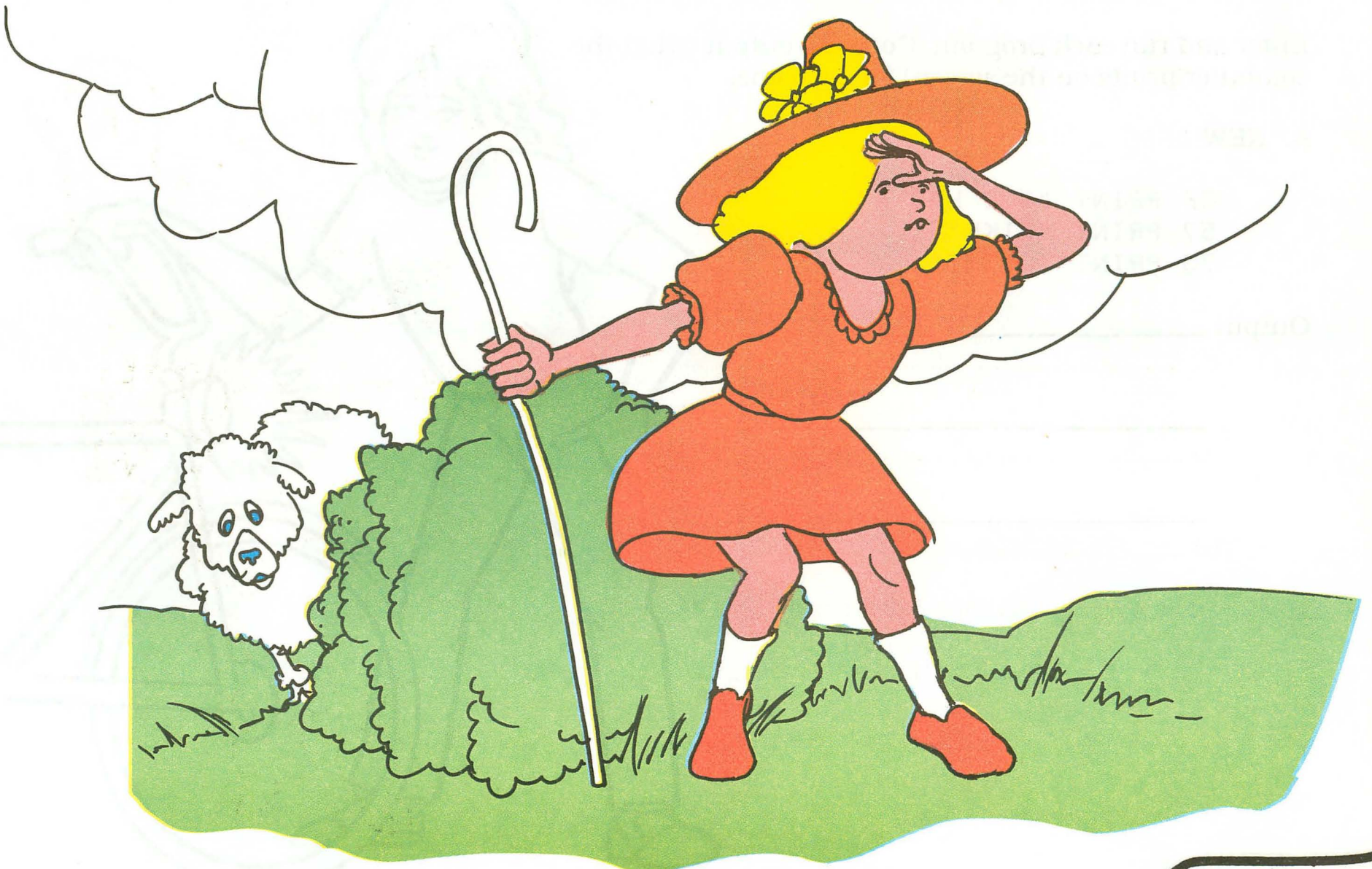
30 _____

40 _____

Be sure your program skips a line before BOOHOO.

Add line 50 so that BOOHOO keeps repeating:

50 _____



PROBLEM 2 TAKE A NUMBER, PLEASE

Enter and run each program. Copy the output (what the computer prints on the screen) for each one.

A. NEW

```
38 PRINT "SUE HAS A"
57 PRINT "BIKE"
73 PRINT "LIGHT"
```

Output _____

_____

_____

B. NEW

```
86 PRINT "SAM HAS A"
93 PRINT "BIKE"
88 PRINT "LIGHT"
```

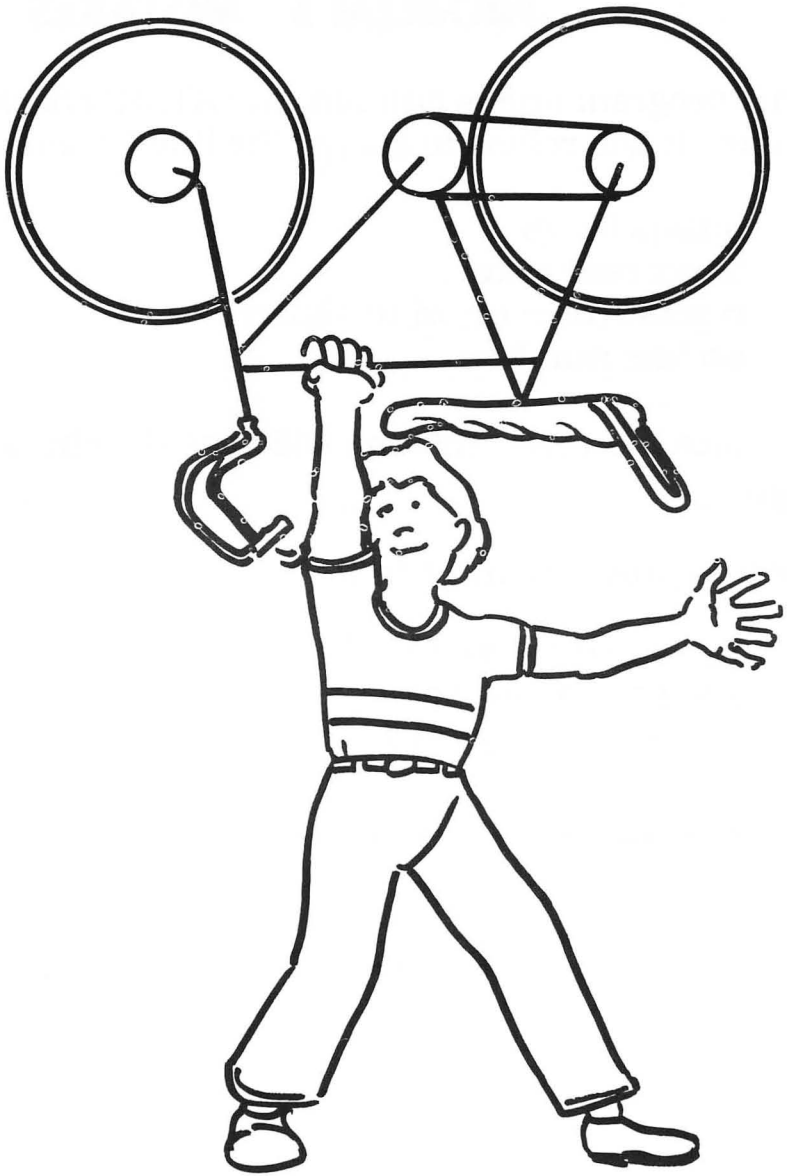
Output _____

_____

_____

Why did the programs print something different?

_____



PROBLEM 3 MISTAKES

When a program line is entered, the ATARI checks it for mistakes. It will refuse to accept the line for lots of reasons:

misspellings
lower case letters
reverse (blue on white) letters
no line number

Sometimes, however, the line will pass the check, but it won't be right.

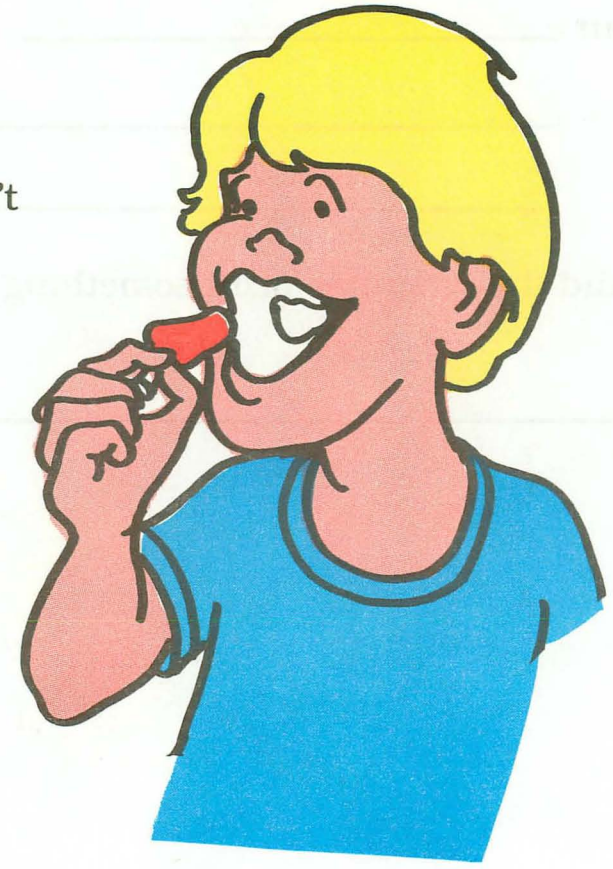
Enter this program and run it:

```
10 PRINT "RED"
20 PRINT "HOT"
30 PRINT PEPPER
```

What was printed? _____

_____

_____



Change line 30 so that the program prints

RED
HOT
PEPPER

Write your new line 30 here

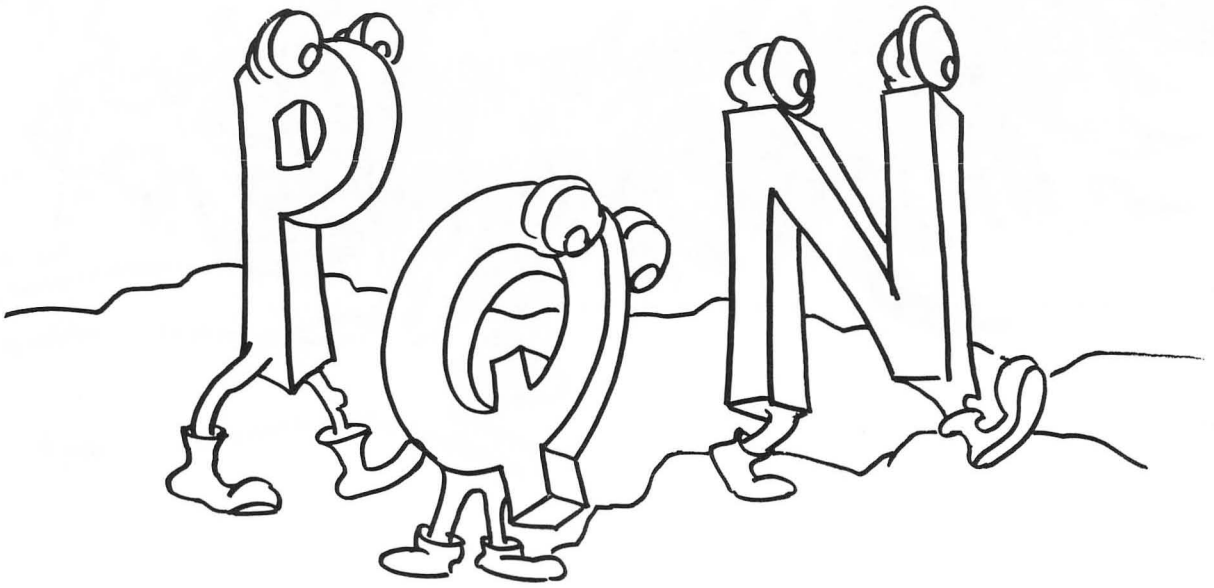
30 _____

PROBLEM 4 INITIALS

Run this one:

```
10 PRINT
20 PRINT
30 PRINT "PPPP"
40 PRINT "P  P"
50 PRINT "PPPP"
60 PRINT "P "
70 PRINT "P "
80 PRINT "P "
```

Write a program like this that uses your initial; it should make a large initial out of several small ones.



Copy your program here: (you might not need all the lines)

10 _____

20 _____

30 _____

40 _____

50 _____

60 _____

70 _____

80 _____

90 _____

PROBLEM 5 WHERE NOW?

Unless you tell it to go somewhere else, a program that just finished one line will go to the next line number.

A. Look at this program. PREDICT what it will do. Then enter and run it to check.

```
10 GOTO 35
15 PRINT "T";
20 GOTO 45
25 PRINT "A";
30 GOTO 15
35 PRINT "R";
40 GOTO 25
45 END
```

Prediction: _____

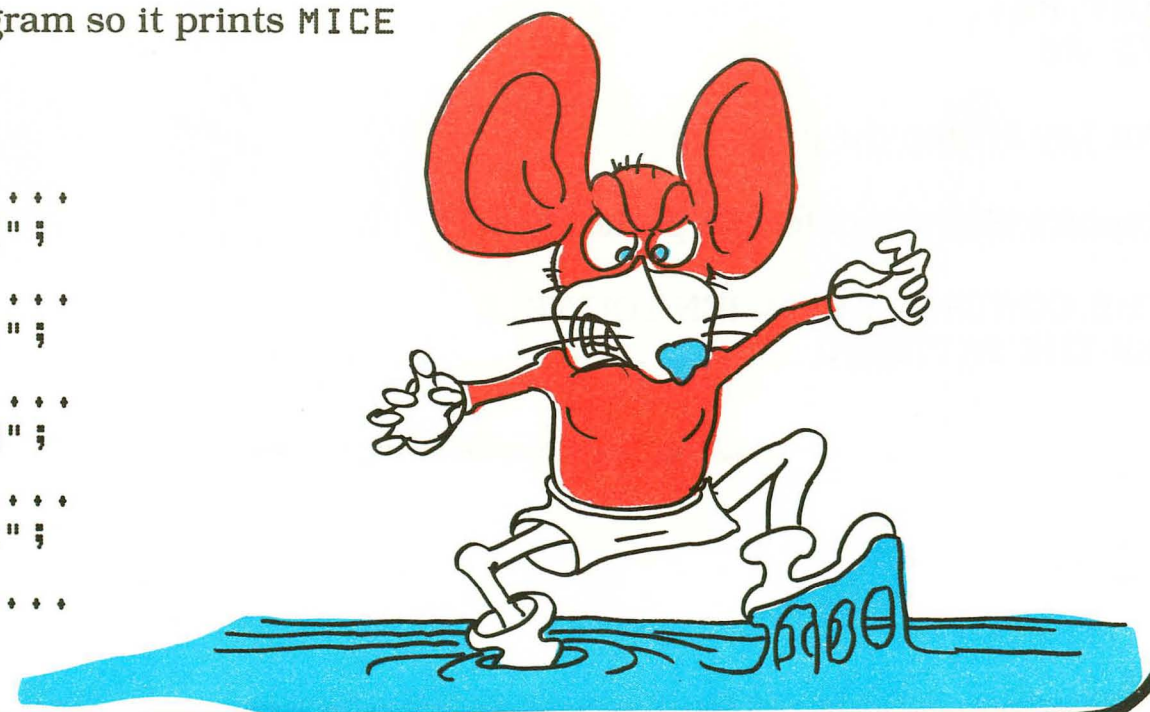
B. Fill in the GOTO statements to make the program print ART.

```
10 GOTO ......
15 PRINT "T";

20 GOTO ......
25 PRINT "A";

30 GOTO ......
35 PRINT "R";

40 GOTO ......
45 END
```

What makes the letters print on the same line? _____

C. Complete this program so it prints MICE

```
10 GOTO ......
20 PRINT "E";

30 GOTO ......
40 PRINT "I";

50 GOTO ......
60 PRINT "M";

70 GOTO ......
80 PRINT "C";

90 GOTO ......
100 END
```



PROBLEM 6 WALLPAPER

This program will design wallpaper. You'll give it a basic pattern, and it will keep repeating it.

Each time it's run, the program will ask for the pattern. When it does, type in a pattern and press RETURN.

The computer will put the pattern in a special place, called P$. Line 10 tells the computer that the pattern won't be more than 20 characters long.

```
10 DIM P$(20)
20 PRINT "WHAT'S THE PATTERN"
30 INPUT P$
40 PRINT P$;
50 GOTO 40
```

Press the BREAK key to stop the program.

We'll use the graphice characters for the patterns.

HOLD DOWN THE CONTROL KEY WHEN YOU TELL THE COMPUTER THE PATTERN!

Try these:

1. (A) (T) (A) (R) (I) The circles mean hold down CTRL.
2. (M) (A) (R) (Y)
3. (B) (A) (S) (I) (C)
4. (H) (E) (N) (R) (Y)
5. (C) (O) (M) (P) (U) (T) (E) (R)

You can use the ATARI key to reverse the pattern.

6. (HJ) ....  (JH)

Press the BREAK key to stop the program

PROBLEM 7 LETTER TO SANTA

In this program, the "user" (that's whoever is running it) will enter four pieces of information. Lines 11, 12, 13 and 14 put names on special places for the four different things.

Read through lines 11 – 90. Then answer the questions at the end.

```
 11 DIM N$(20)
 12 DIM G$(20)
 13 DIM Q$(20)
 14 DIM QG$(20

 20 PRINT "WHAT'S YOUR NAME"
 30 INPUT N$
 40 PRINT "WHAT WOULD YOU LIKE?"
 50 INPUT G$
 60 PRINT "WHO ELSE IS ON YOUR LIST?"
 70 INPUT Q$
 80 PRINT "WHAT WOULD "; Q$; " LIKE?"
 90 INPUT QG$

100 PRINT
110 PRINT "DEAR SANTA"
120 PRINT "PLEASE BRING ME ";_____

130 PRINT _____
```

```
140 PRINT "WOULD LIKE A ";_____
150 PRINT "THANK YOU."
160 PRINT
170 PRINT "LOVE"
180 PRINT_____
```

Draw a line from each "name" to what it stands for.

N$	gift you'd like
G$	other person's name
Q$	other person's gift
QG$	your name

One of the "names" should be written in each set of ____ in the program.

Put in the names and run the program to check it.


Dear Santa,
Gimme Gimme

PROBLEM 8 CURSOR CONTROLS

On the ATARI computers you can move the cursor up, down, right and left by using the cursor control keys.

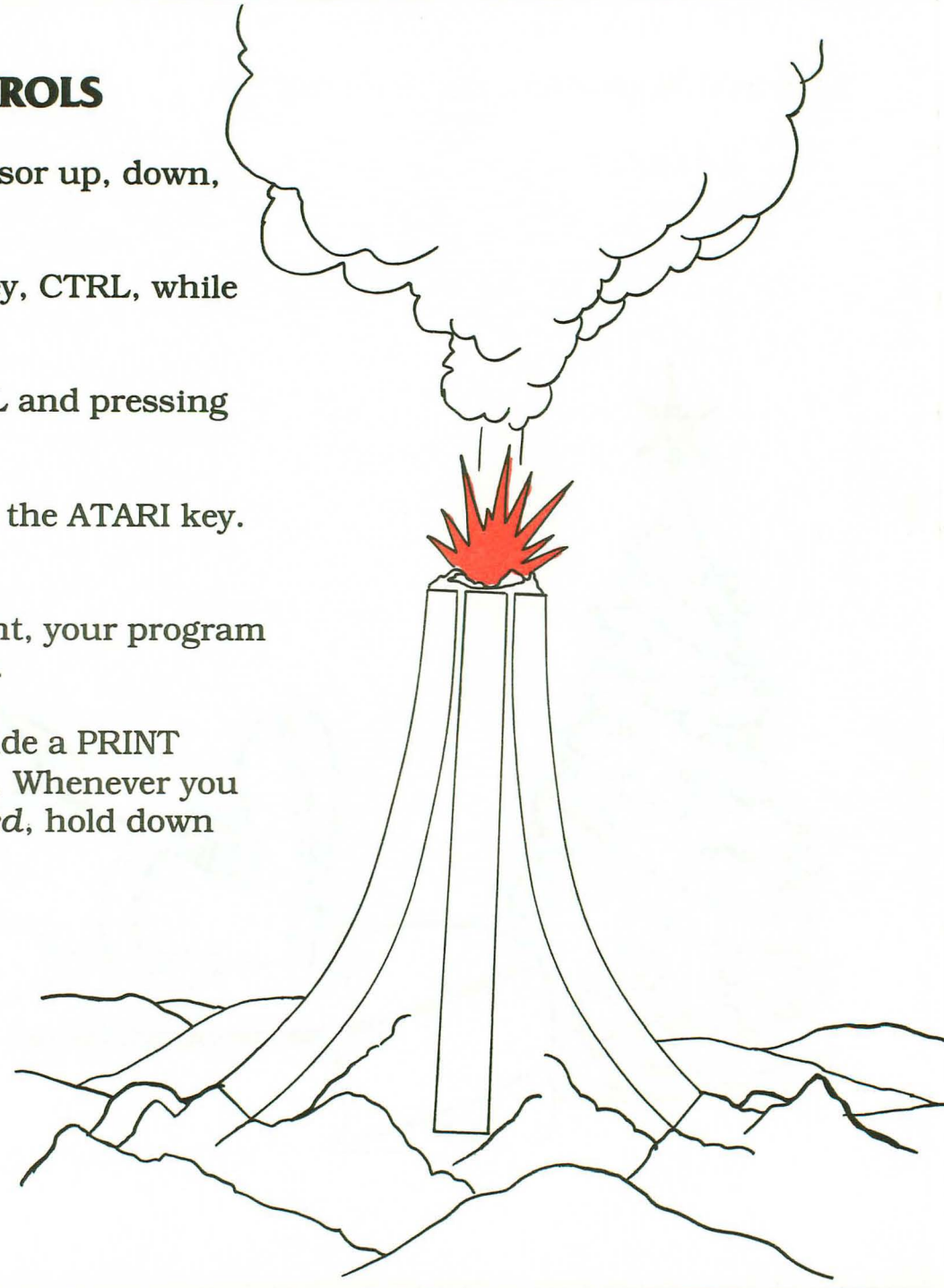
To control the cursor, hold down the control key, CTRL, while pressing the arrow key.

You can clear the screen by holding down CTRL and pressing CLEAR.

You can switch to reversed characters by using the ATARI key. Switch back by pressing it again.

If these commands are inside a PRINT statement, your program will make the cursor move and the screen clear.

To clear the screen or command the cursor inside a PRINT statement, you MUST hit the ESCAPE key first. Whenever you see *E*, hit the ESC key. Whenever a key is *circled*, hold down the CTRL key while you hit it.



Type in this program, but don't run it yet:

```
10 PRINT "E(clear)HELLO"(clear)
20 PRINT "E↓  THERE!"
```

(clear) means hold CTRL and hit CLEAR

On the screen you should see

```
10 PRINT " ↄ HELLO"
20 PRINT " ↓ THERE!"
```

The ATARI key does not need the ESC key.

Add these lines to your program and run it:

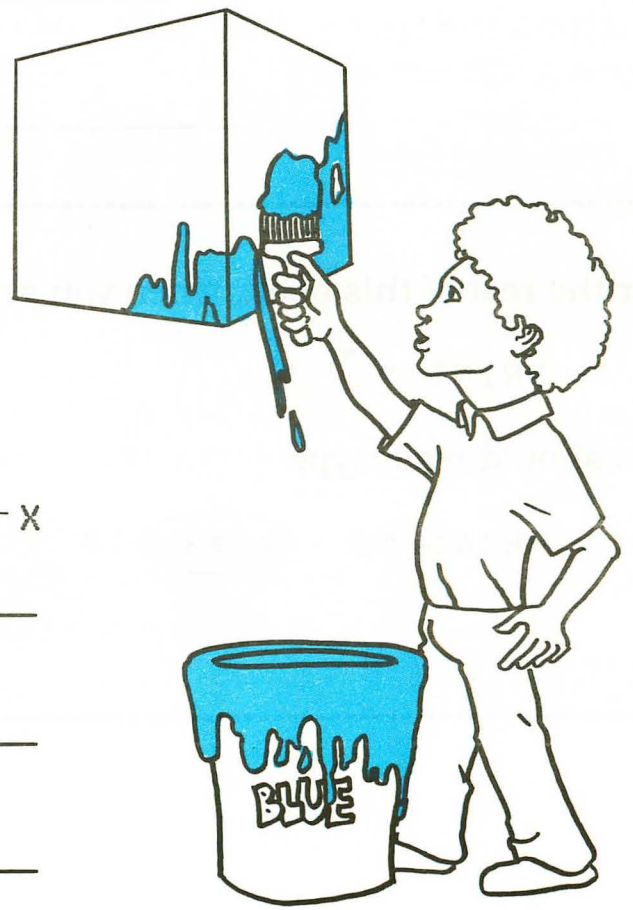
```
30 PRINT "▲HOW ARE▲ YOU?"
40 PRINT "XE↓XE↓XE↓XE↓XE↓←XE↓←X
```

What did line 40 draw? _____

_____

_____

_____



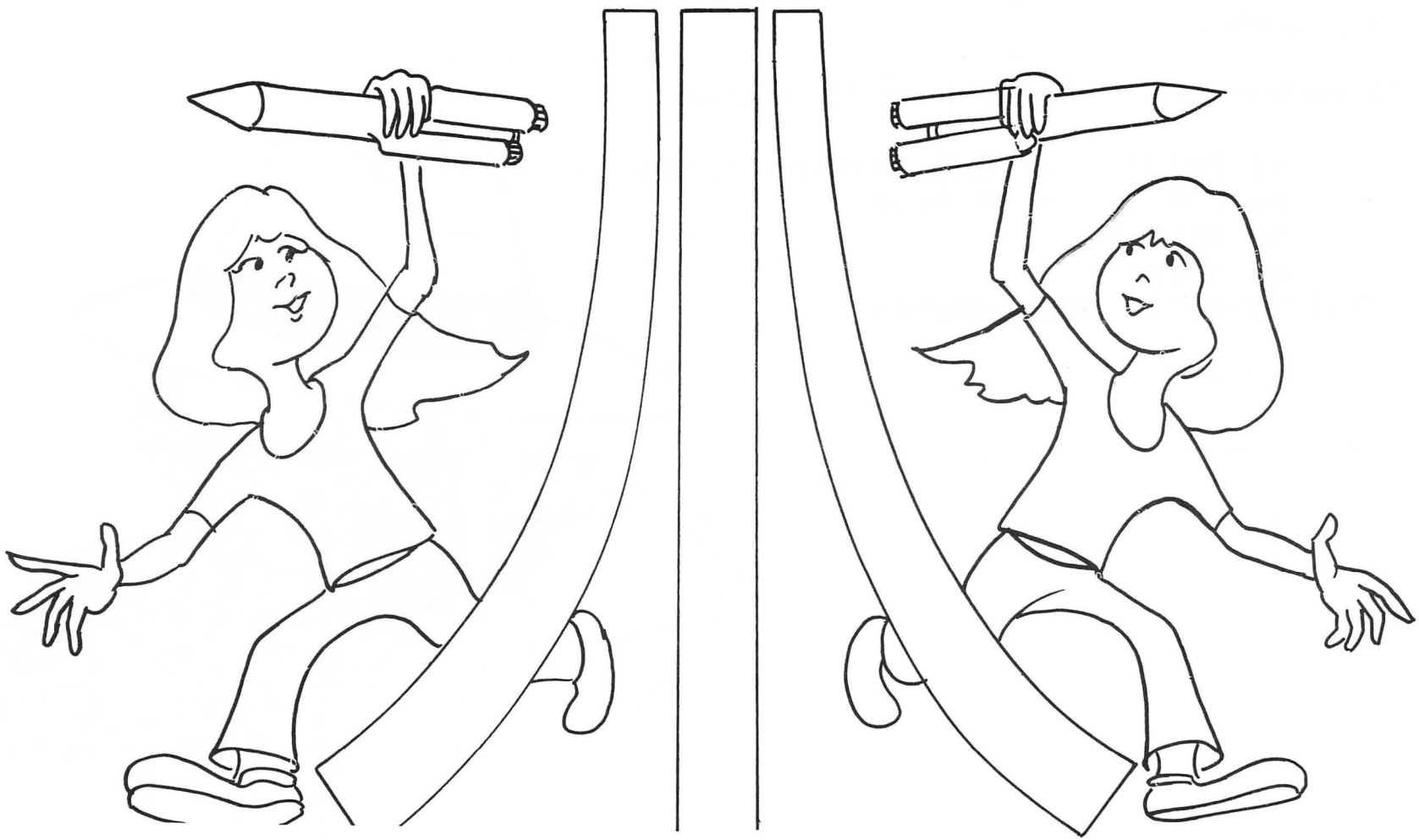
For the rest of this book, when you see

```
PRINT " ↻ "
```

you should really type

```
PRINT "E  (clear)  "
```

where clear means hold CTRL and hit CLEAR.



PROBLEM 9 PAUSING FOR BREATH

To slow down the computer, we can make it stop and count to itself. Since it counts very fast, we'll have to make it count to a high number.

To see how to do it, enter and run this program:

```
10 PRINT " ↻ ONE MINUTE, PLEASE"
20 FOR T = 1 TO 5000
30 NEXT T
40 PRINT
50 PRINT "NOW WHAT?"
```

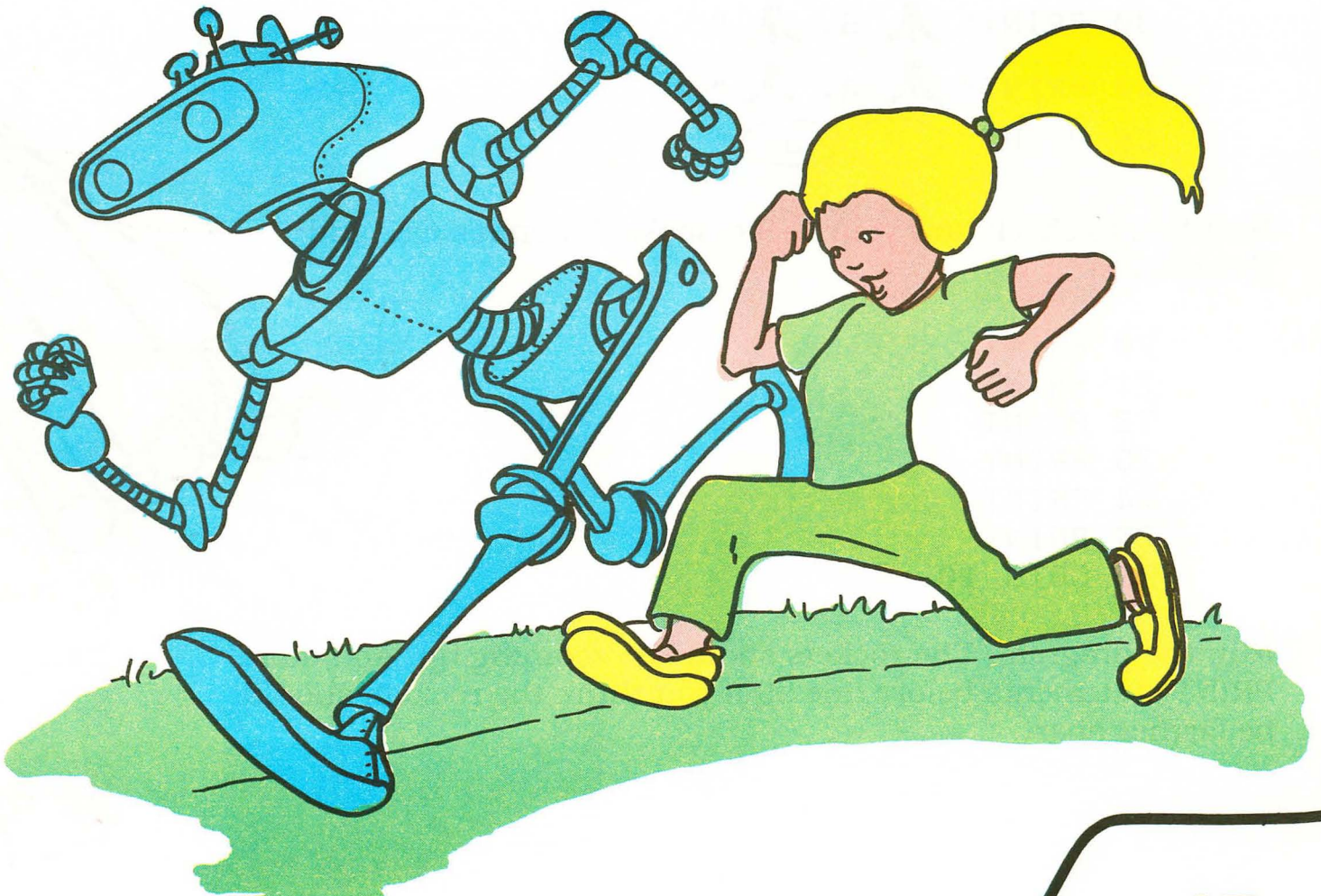
Get a watch or stopwatch and time how long it takes from the time you press RETURN after typing RUN until the NOW WHAT? appears. Then change the 5000 in line 20 to each of the following numbers and time the delay.

5000 _____ sec

10000 _____ sec

20000 _____ sec

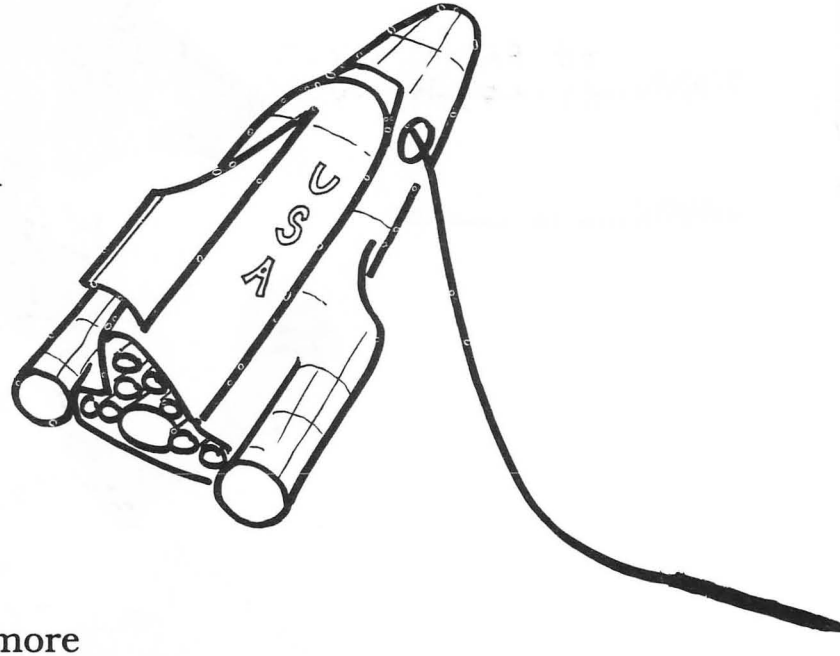
40000 _____ sec



PROBLEM 10 ROCKETS

We're going to send rockets up into the wild blue yonder. The rocket body will be made with reversed characters.

```
10 PRINT" H J "
20 PRINT" ▚ U. ▚ "
30 PRINT" ▚ S. ▚ "
40 PRINT" ▚ A. ▚ "
50 PRINT" H J "
```



Run this to check how your rocket looks. When it's OK, add these lines:

```
70 PRINT
71 PRINT
72 PRINT
73 PRINT
74 PRINT
75 PRINT
99 GOTO 10
```

Run the program. The rockets should be zooming up. Add more PRINT statements before line 99 if you think the rockets should be farther apart.

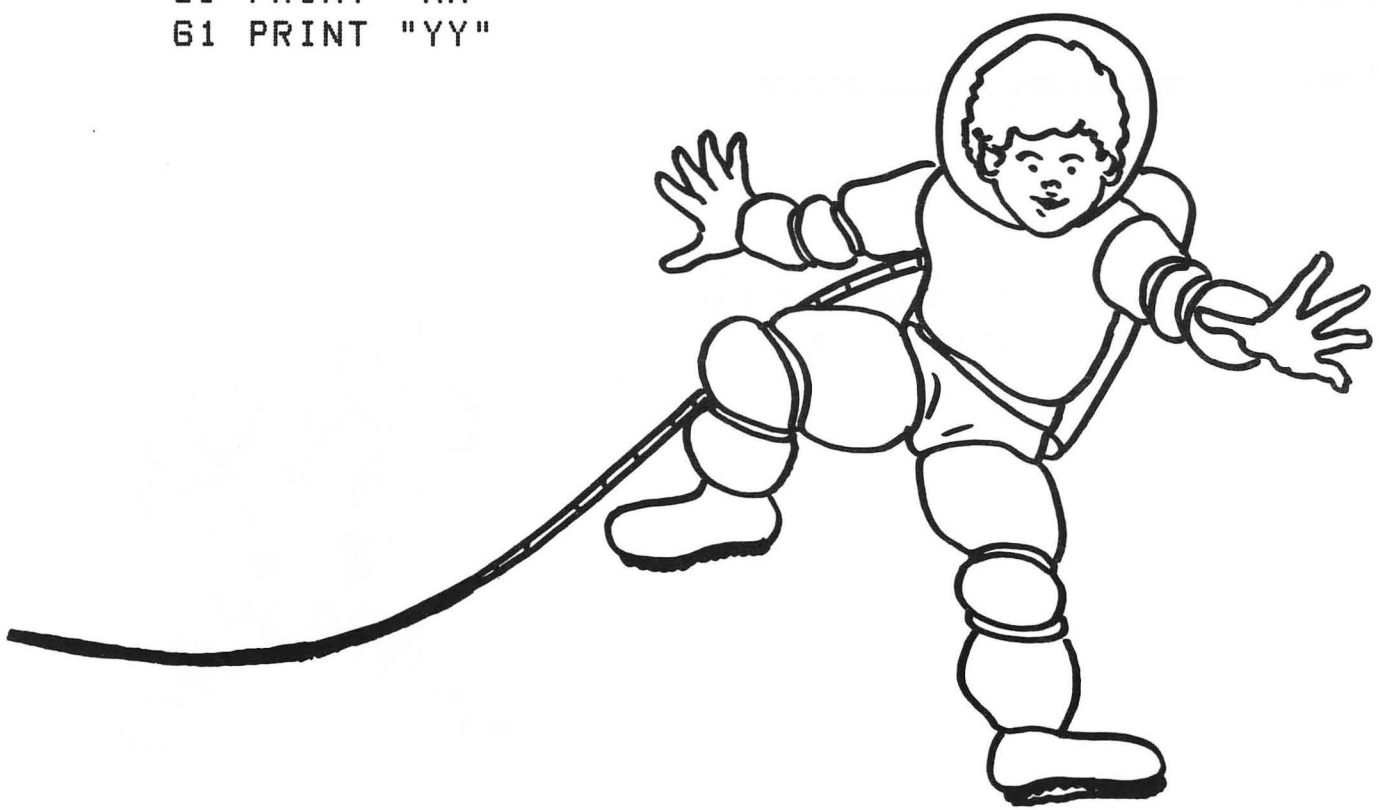
To make the rockets blast off, add a pause at lines 55 and 56:

```
55 FOR T = 1 TO 2000
56 NEXT T
```

Adjust the pause, if you like.

Now for some exhaust at ignition time. After the pause we'll put in some smoke:

```
60 PRINT "XX"
61 PRINT "YY"
```



PROBLEM 11 COUNTING

To watch the computer count, enter this program and run it:

```
10 FOR K = 1 TO 15
20 PRINT K
30 NEXT K
```

Change line 10 so the numbers from 1 to 100 appear:

10 _____

Change line 20 so all 100 numbers are on the screen at one time (like Wallpaper):

20 _____

Get out your watch and time how long it takes the computer to count to 5000 if it has to take time to write the numbers. It takes

_____ sec

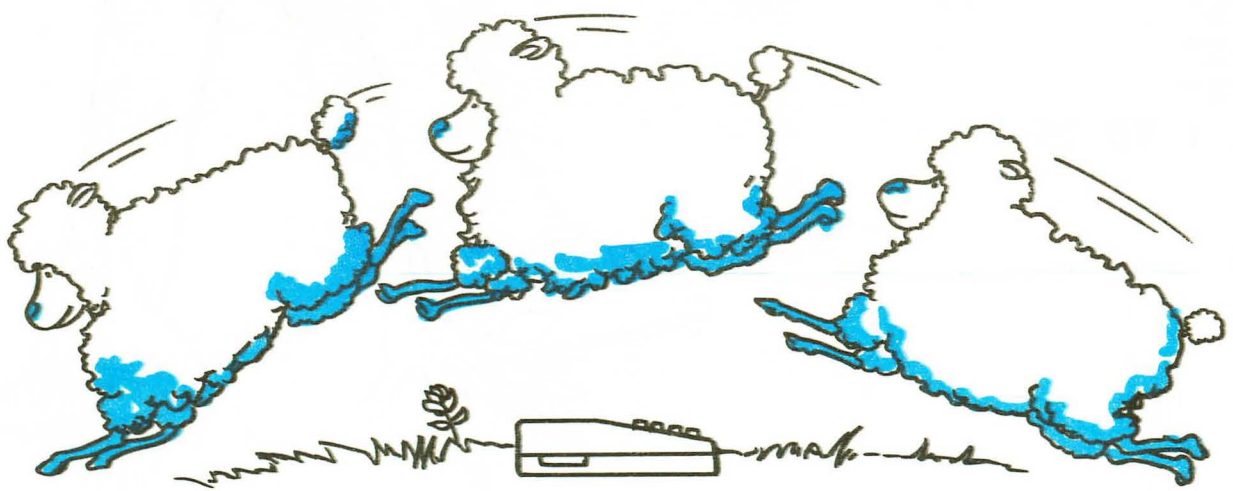
Run the next program three times to test your reflexes.

```
10 PRINT " ) "
20 PRINT "WHEN THE NUMBERS APPEAR"
30 PRINT "PRESS THE BREAK KEY"
40 FOR T = 1 TO 1500
50 NEXT T
60 FOR K = 1 TO 500
70 PRINT K
80 NEXT K
```

What were your reaction numbers? _____

How can you tell if someone presses BREAK before the numbers appear?

_____



PROBLEM 12 TOUCH YOUR TOES TEN TIMES

The FOR...NEXT loop (that's what it's called whenever you use FOR and NEXT to count something) can be used to do something a fixed number of times. Try each of these to see how:

A. It can print a message.

```
10 FOR M = 1 TO 10
20 PRINT "TOO BAD"
30 NEXT M
```

Change line 20 to print something else.

B. It can skip spaces.

```
10 PRINT "A"
20 PRINT "X"
30 PRINT "X"
40 PRINT "X"
50 FOR S = 1 TO 25
60 PRINT
70 NEXT S
80 GOTO 10
```

C. It can draw a box.

```
10 PRINT "*****"
20 FOR S = 1 TO 10
30 PRINT "*...*"
40 NEXT S
50 PRINT "*****"
```

Erase the dots in line 30 so the box is empty.

Write a program that draws a high, wide box.

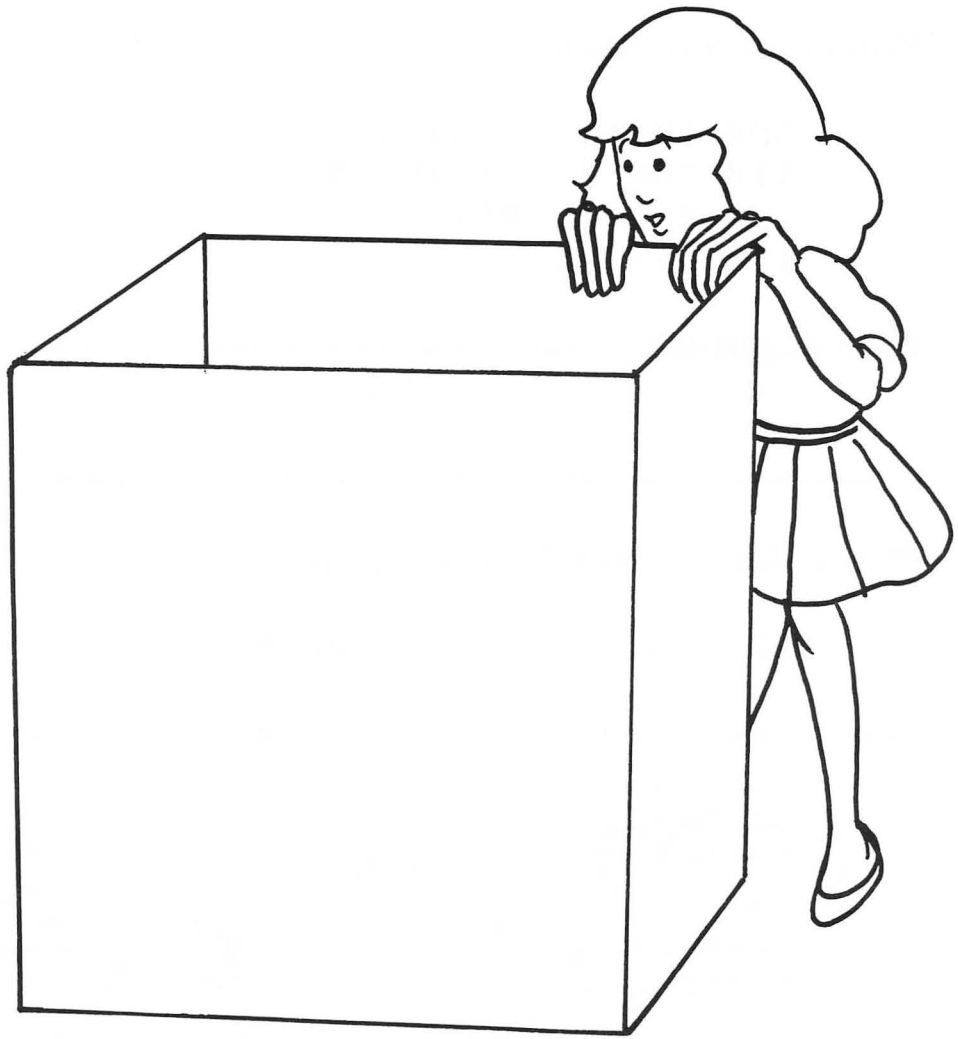
10 _____

20 _____

30 _____

40 _____

50 _____



PROBLEM 13 MOVERS

Enter and run this program.

```
100 PRINT "☽ *";
110 FOR K = 1 TO 35
120 PRINT "*";
150 NEXT K
```

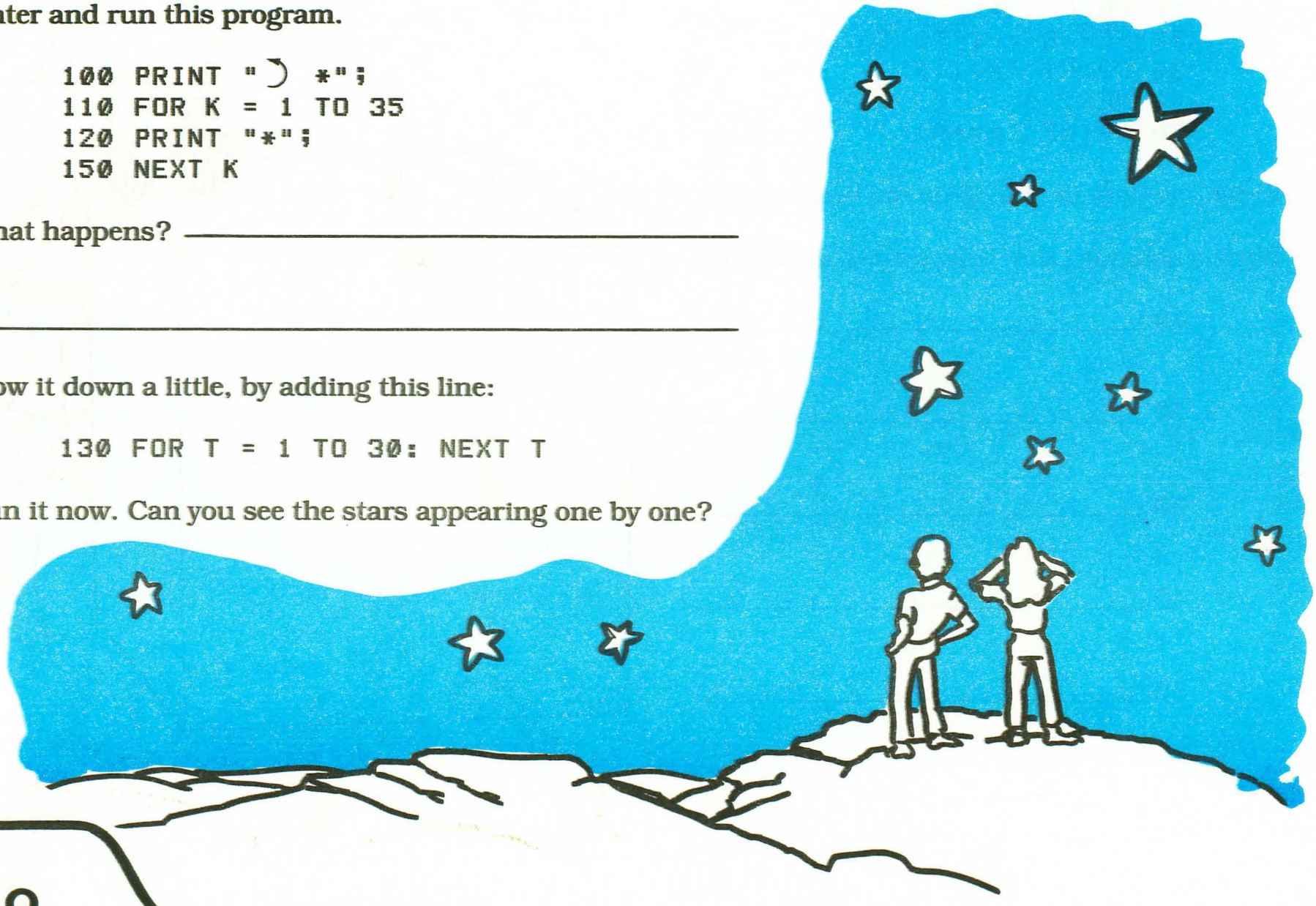
What happens? _____

_____

Slow it down a little, by adding this line:

```
130 FOR T = 1 TO 30: NEXT T
```

Run it now. Can you see the stars appearing one by one?



If we erase each star before we print the next one, it will look as if just one star is moving. Change line 120 so that it erases as it goes:

```
120 PRINT" E ← (SP) *";
```

Run it now. Does the last star stay or disappear? _____

Finally, make the star leave a trail of dots:

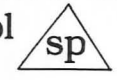
```
120 PRINT"E ← *";
```

PROBLEM 14 THE WORM CROSSES

This time we'll move a worm. The worm will have five body segments and a head.

●●●●●*

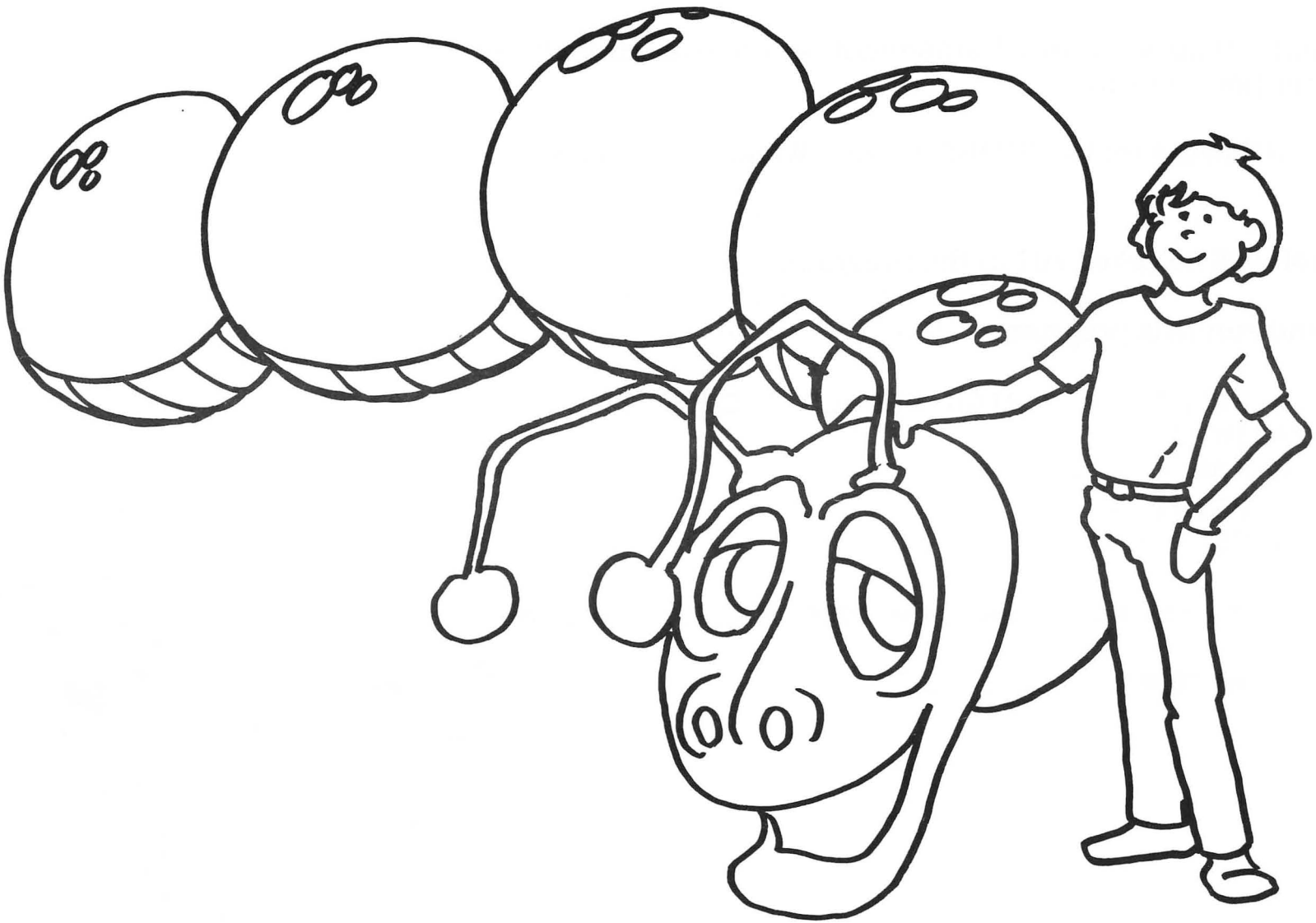
The body segments are made by CTRL and T.

To erase the tip of the tail, we'll need to move back six times and then print a space. The symbol 

means hit a space.

```
1000 PRINT " ↻ T T T T T *";
 110 FOR K = 1 TO 30
 120 PRINT "E ← E ← E ← E ← E ← E←
 sp  T T T T T *";
 130 FOR T = 1 TO 50
 140 NEXT T
 150 NEXT K
```

Adjust the speed; make the worm cross more slowly, then make it cross faster.



PROBLEM 15 MORE COUNTING

This program uses an INPUT statement, so the user can tell the computer how high to count.

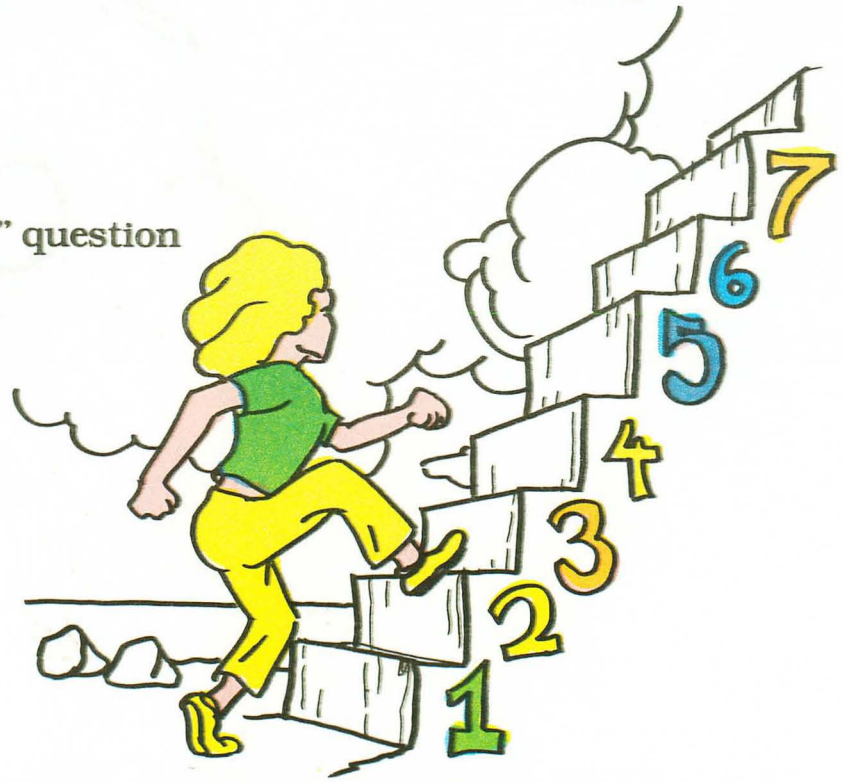
Because it's being told a NUMBER, not a WORD, the $ is not used.

How High will be called HH in the program.

Enter and run this program:

```
10 PRINT "HOW HIGH SHOULD I COUNT?"
20 INPUT HH
30 FOR K = 1 TO HH
40 PRINT K
50 NEXT K
```

What happens when you answer the "HOW HIGH ..." question with the word TEN? _____



(When the computer expects a NUMBER, you must enter a number.)

Add a line to make the program print

```
ALL DONE
```

when it has finished counting. Write the line you added here:

_____

What punctuation mark at the end of line 40 will make the computer print the counting numbers across the

screen? _____

Try it.

PROBLEM 16 TESTING WITH IF...THEN

In BASIC, "=" means "is the same as"
"<>" means "is not the same as"

We can compare numbers and we can compare words.

Enter and run this program:

```
 5 DIM C$(20)
10 PRINT "TELL ME A COLOR"
20 INPUT C$
30 IF C$ = "GREEN" THEN PRINT "GRASS IS
GREEN"
```

What happens if you say RED? _____

Now you add line 40 so that the response to PINK is

```
VERY PREPPY.
```

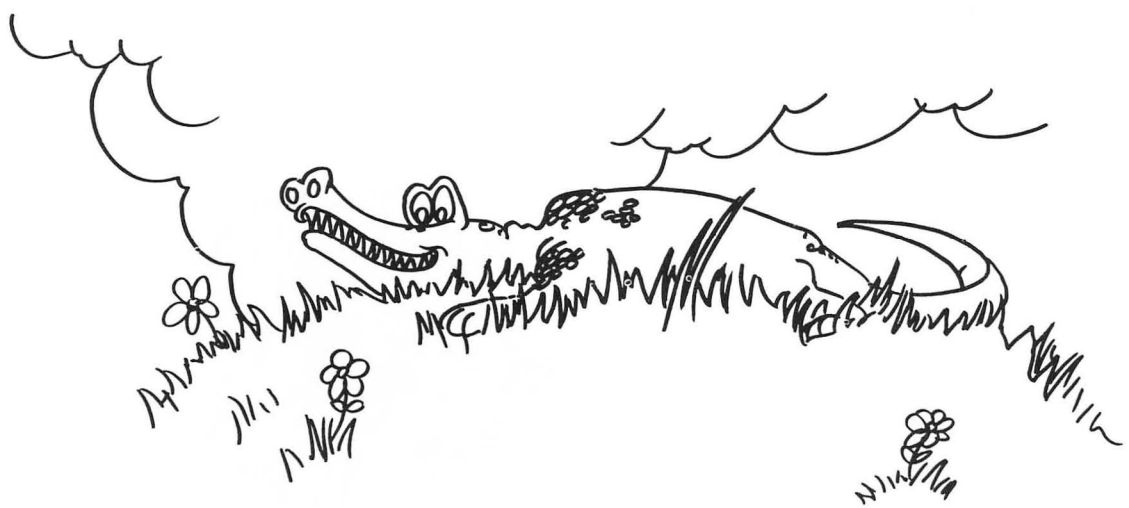
Write line 40 here:

40 _____

Finally, add line 50:

```
50 IF C$ = "RED" OR C$ = "BLUE" OR C$ =
"YELLOW" THEN PRINT "PRIMARY COLOR"
```

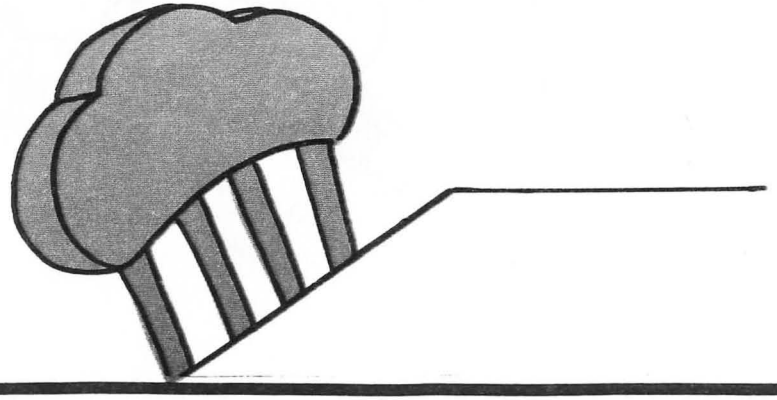
Run the program for all the colors you've used.



PROBLEM 17 GOLDILOCKS

Complete and enter the Goldilocks program:

```
 3 DIM A$ (20)
 5 PRINT " ↶ WHOSE PORRIDGE IS IT?"
10 PRINT "E ↓ TASTE THE PORRIDGE"
15 PRINT "E ↓ ENTER HOT, COLD, OR OK"
20 INPUT A$
30 IF A$ = "HOT" THEN PRINT "PAPA'S"

40 IF A$ = "COLD" THEN PRINT "_________

50 _____________________________________
60 END
```



PROBLEM 18 ANIMAL GUESS

Now we'll use IF .... THEN to play a guessing game. First we'll set the animal to be a HORSE, then we'll add extra lines to make it a two-person game.

Look at the program. What happens after line 140 if the guess is HORSE?

_____

Complete line 170, then enter and run the program.

```
10 DIM AN$(20)
20 DIM G$(20)

100 AN$ = "HORSE"
110 PRINT " ↶ "
120 PRINT "I'M THINKING OF AN ANIMAL"
130 PRINT "E ↓ GUESS"
140 INPUT G$
150 IF G$ <> AN$ THEN PRINT "NO"
160 IF G$ <> AN$ THEN GOTO 130

170 _____________________________________________
```

Once the program is working, add these lines:

```
10 PRINT "THINKER"
20 PRINT "TELL ME AN ANIMAL"
100 INPUT AN$
115 PRINT "GUESSER'S TURN"
```

The thinker types in an animal while the guesser isn't looking.
Then the guesser tries to guess it.

PROBLEM 19 LITTLE INDIANS

We're going to write a program that prints:

```
 1 LITTLE
 2 LITTLE
 3 LITTLE
INDIANS
 4 LITTLE
 5 LITTLE
 6 LITTLE
INDIANS
 7 LITTLE
 8 LITTLE
 9 LITTLE
INDIANS
 10 LITTLE
NAVAJO KIDS
```

Underneath which numbers is the word INDIANS? _____

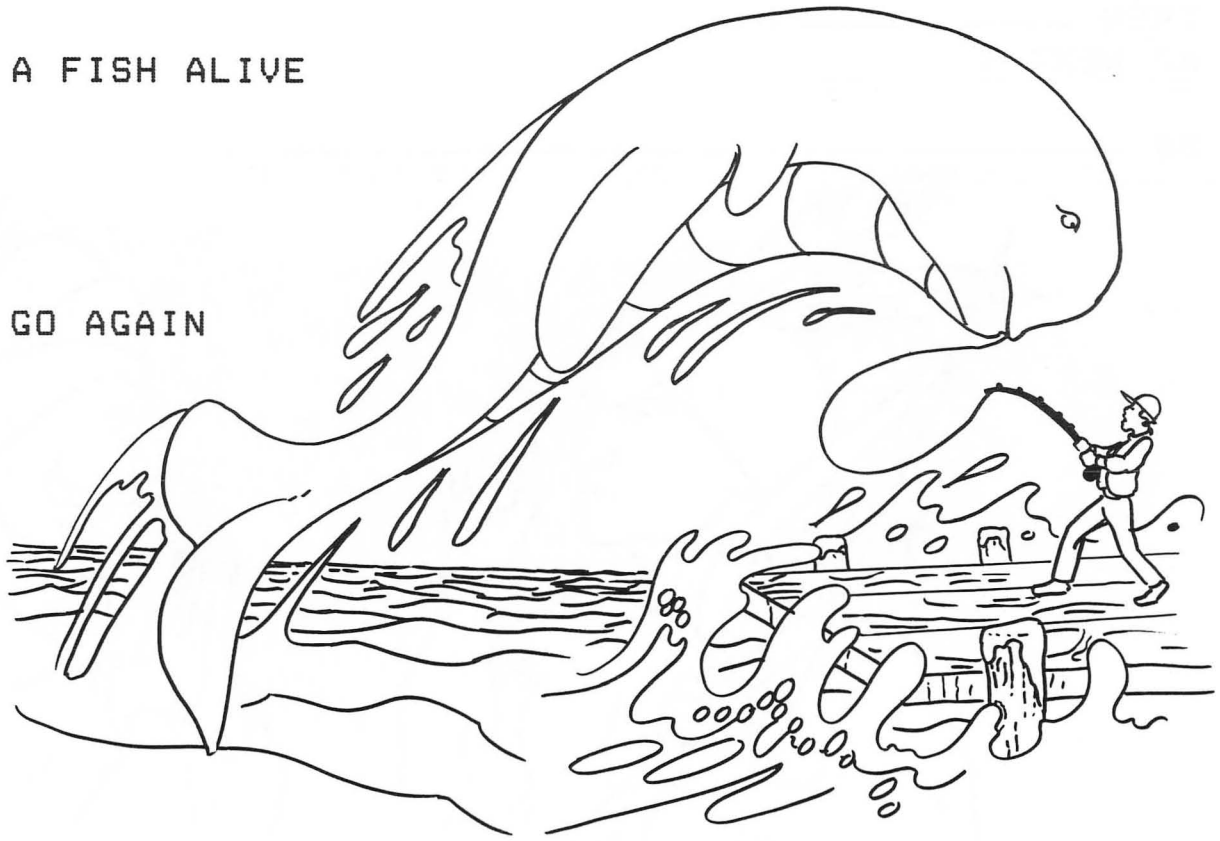
Complete this program and run it.

```
10 FOR K = 1 TO _____
20 PRINT K; " LITTLE"

30 IF K = _____ OR K = _____ OR K = _____

THEN _____
40 NEXT K

50 _____
```

PROBLEM 20 NURSERY RHYME

Now you write a counting rhyme program. Your program should print one of the rhymes below.

```
 1
 2
 3
 4
 5
ONCE I CAUGHT A FISH ALIVE
 6
 7
 8
 9
 10
THEN I LET IT GO AGAIN
```



or this one

```
 1
 2
 3
 4
MARY AT THE COTTAGE DOOR
 5
 6
 7
 8
EATING CHERRIES OFF A PLATE
```

Use this space to record your program. You may not need all the lines.

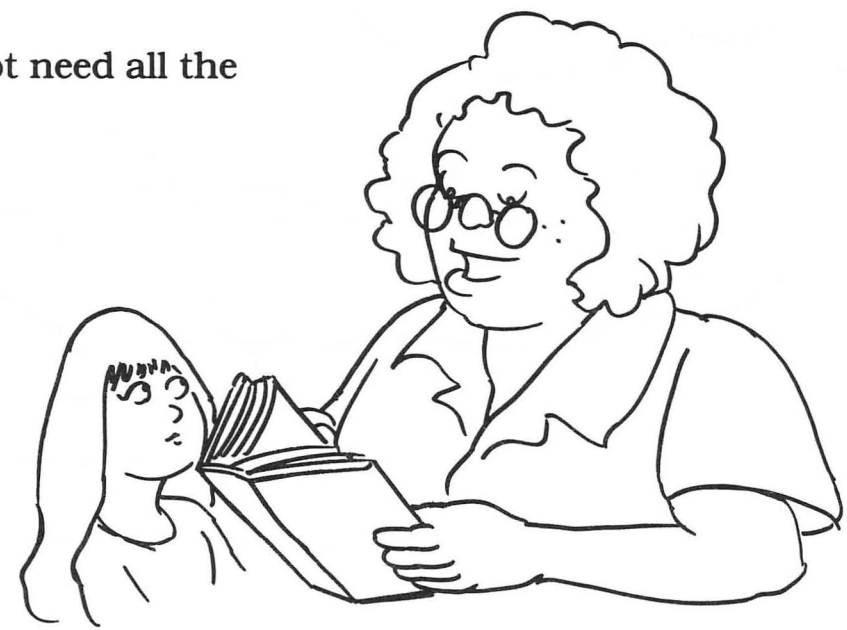
10 FOR _____

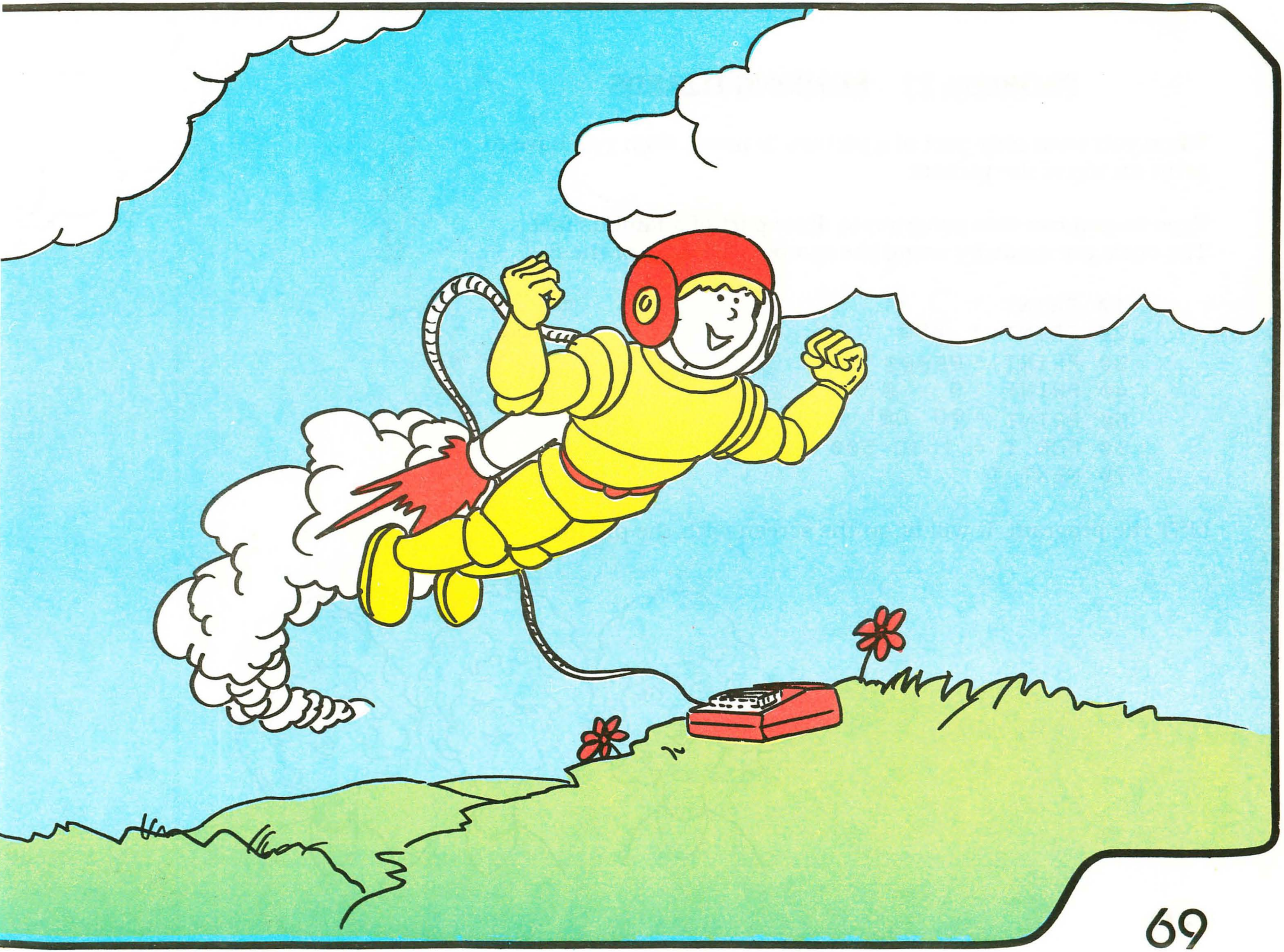
20 PRINT _____

30 IF _____

40 IF _____

50 NEXT _____





PROBLEM 21 BLINKING LIZARDS

When you want only part of a picture to move, then you have to print on top of the picture.

Type in and run this program to draw part of a famous face. The curls are made by using the symbol @ on top of the 8.

```
10 PRINT "  ↻  ";
20 PRINT "  @@@  "
30 PRINT " @@@@@ "
40 PRINT " @    @ "
50 PRINT " @O  O@ "
60 FOR T = 1 TO 500
70 NEXT T
```

LIST the program. It will fit on the screen with the picture.

The eyes are in the 4th row of the screen. That's three rows down from the top row. To get the cursor back to the upper left hand corner, type this line:

```
80 POS.2,0
```

or, you can spell it out:

```
80 POSITION2,0
```

Move the cursor down three rows and close the eyes:

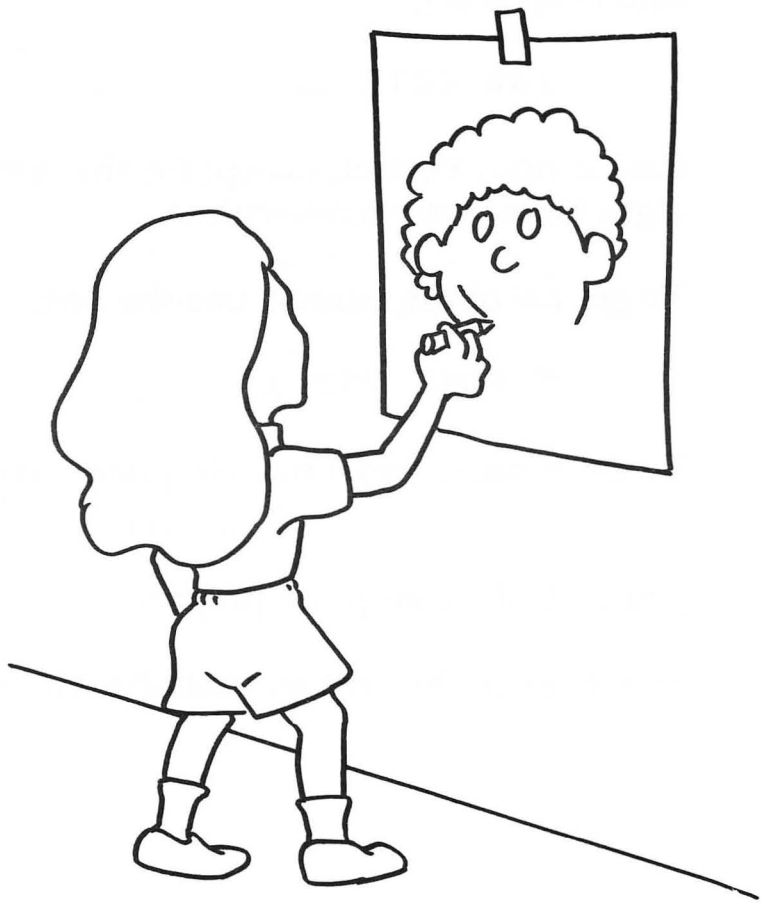
```
90 PRINT "E ↓ E ↓ E ↓ @- -@"
```

Run the program now.

To keep the eyes blinking, first put in a pause while they're closed:

100 _____

110 _____



Send the cursor back to the beginning of the screen:

```
130 POS.2,0
```

and keep going:

```
140 GOTO 20
```

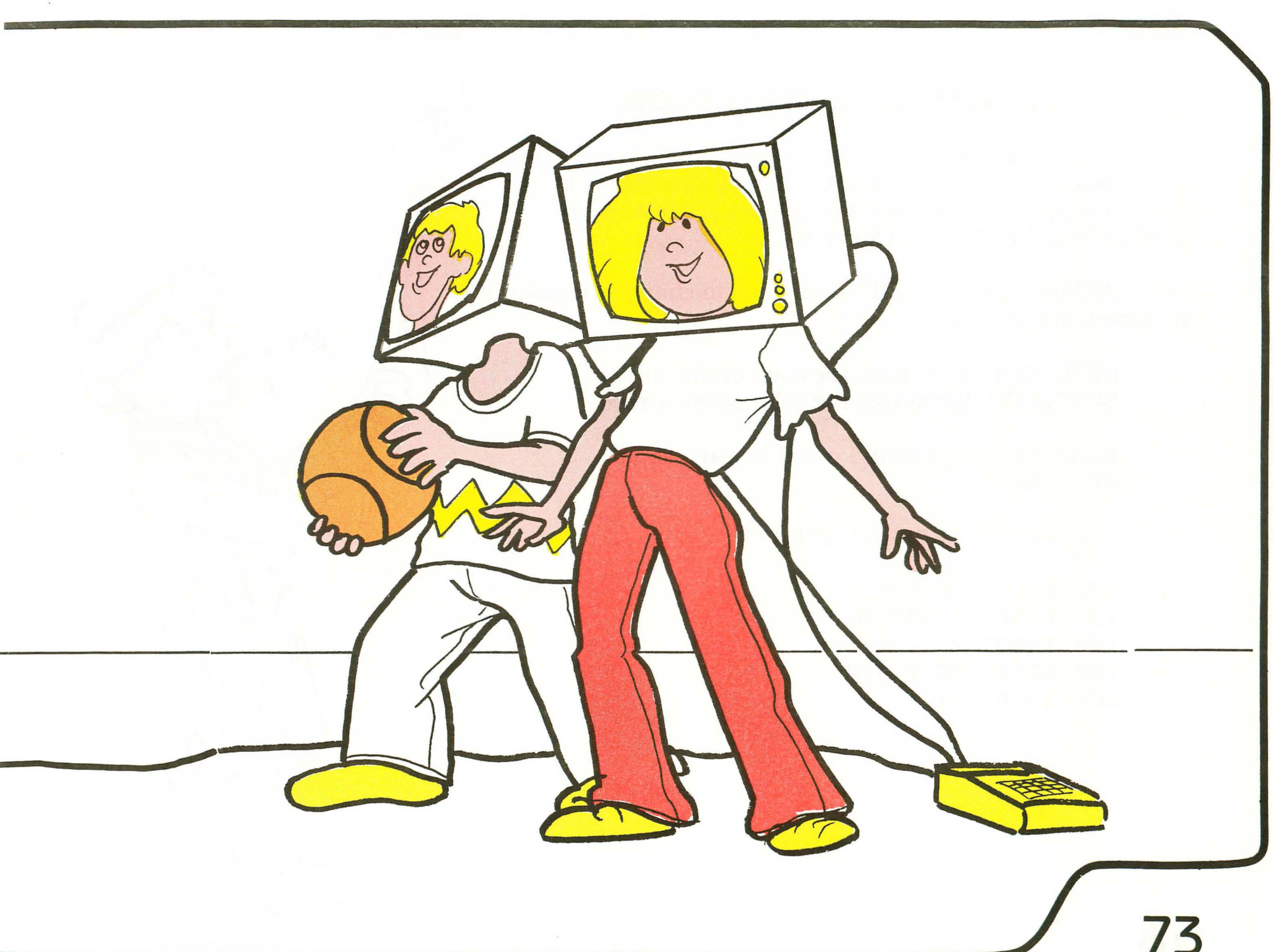
Run it now. Perfect, except for the cursor which keeps showing right below your masterpiece.

To get rid of the cursor, use the line:

```
5 POKE 752,1
```

To get it back, you'll have to press the SYSTEM RESET key after you BREAK to stop the program.

Now change the eyes so that she winks instead of blinks.



PROBLEM 22 SCREEN COLORS

The ATARI has 16 different basic colors for the screen. Each color has a code number. The darkest grey (it's almost black) has code number 0. The code for a cherry-red is 4. Every number from 0 to 15 is a color code

The ATARI has special BASIC commands to change the color of the screen and border:

SETCOLOR 4 changes the color of the border
SETCOLOR 2 changes the color of the screen.

Enter this program and run it to learn about SETCOLOR.

```
100 PRINT "SCREEN COLOR?"
110 INPUT S
120 SETCOLOR 2,S,0
130 PRINT "BORDER?"
140 INPUT B
150 SETCOLOR 4,B,0
160 GOTO 200
```

What color does each color code make on your television screen?

Code Number	Color	Code Number	Color
0	_____	8	_____
1	_____	9	_____
2	_____	10	_____
3	_____	11	_____
4	_____	12	_____
5	_____	13	_____
6	_____	14	_____
7	_____	15	_____

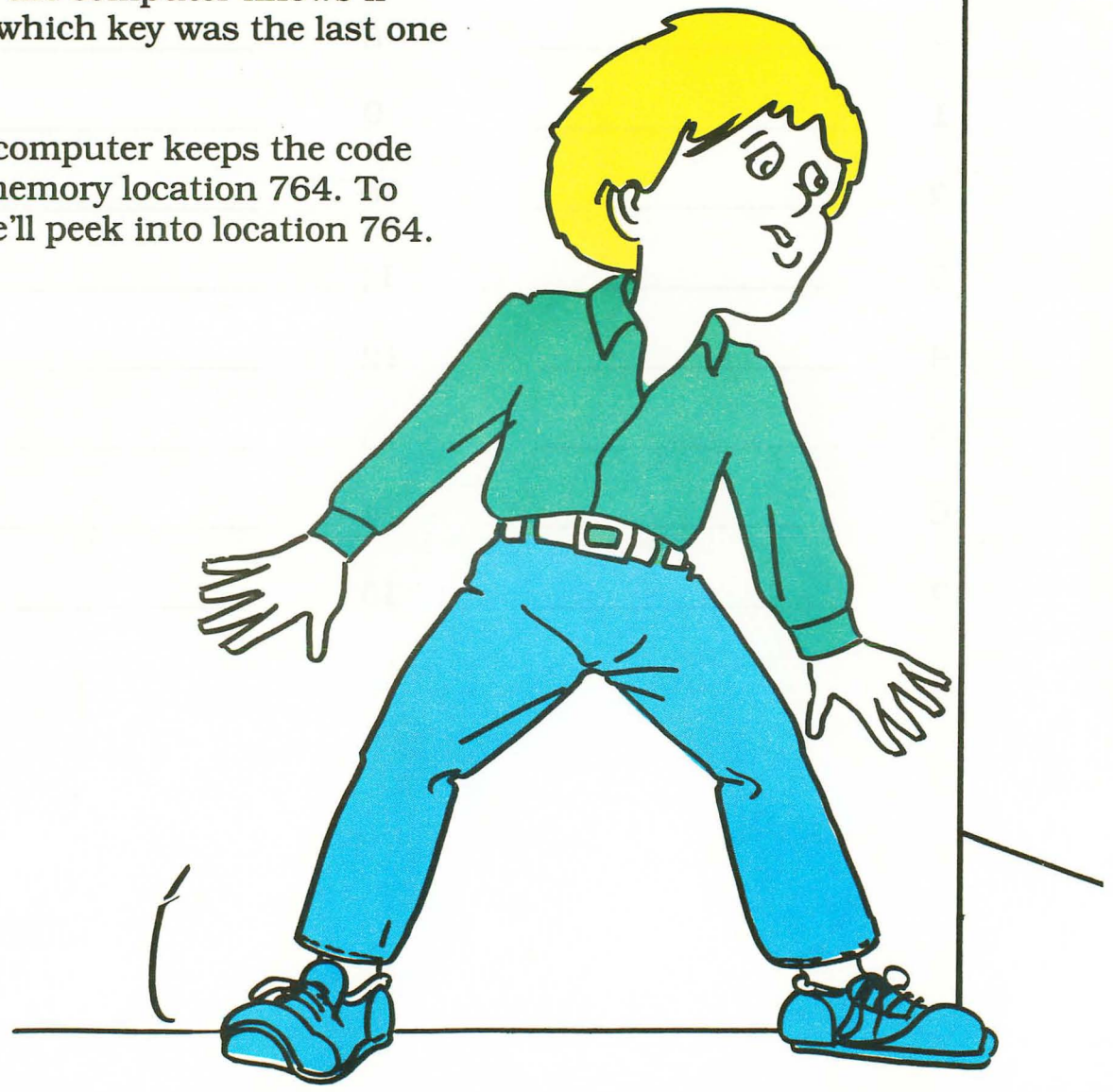
PROBLEM 23 PEEKABOO — THE COMPUTER IS WATCHING YOU

Even while it's running a program, the computer knows if you've pressed a key, and it knows which key was the last one touched.

Every key has a code number. The computer keeps the code number of the last key pressed in memory location 764. To learn some of the code numbers, we'll peek into location 764.

Enter this program:

```
10 K = PEEK(764)
20 PRINT K
30 GOTO 10
```



Run the program and press different keys while the numbers are running. The numbers will change each time you press a different key. Find the code numbers for these keys:

C ____ G ____ space bar ____

X ____ S ____ RETURN _____

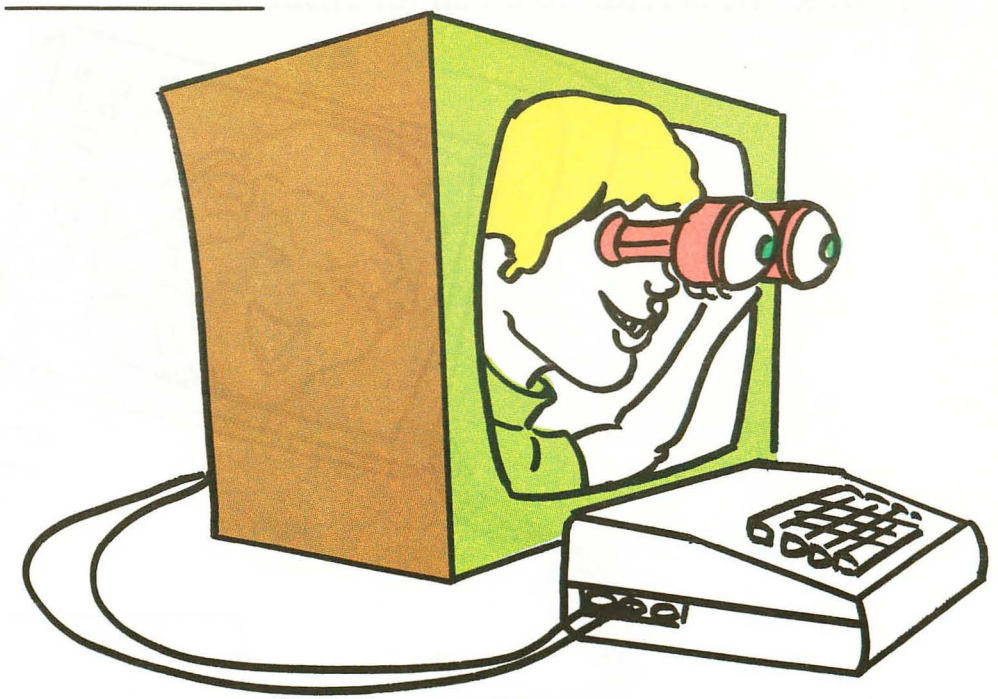
Z ____ 4 ____ * _____

What happens when you touch the BREAK key? _____

Which black keys don't change the numbers?

_____ _____

_____ _____

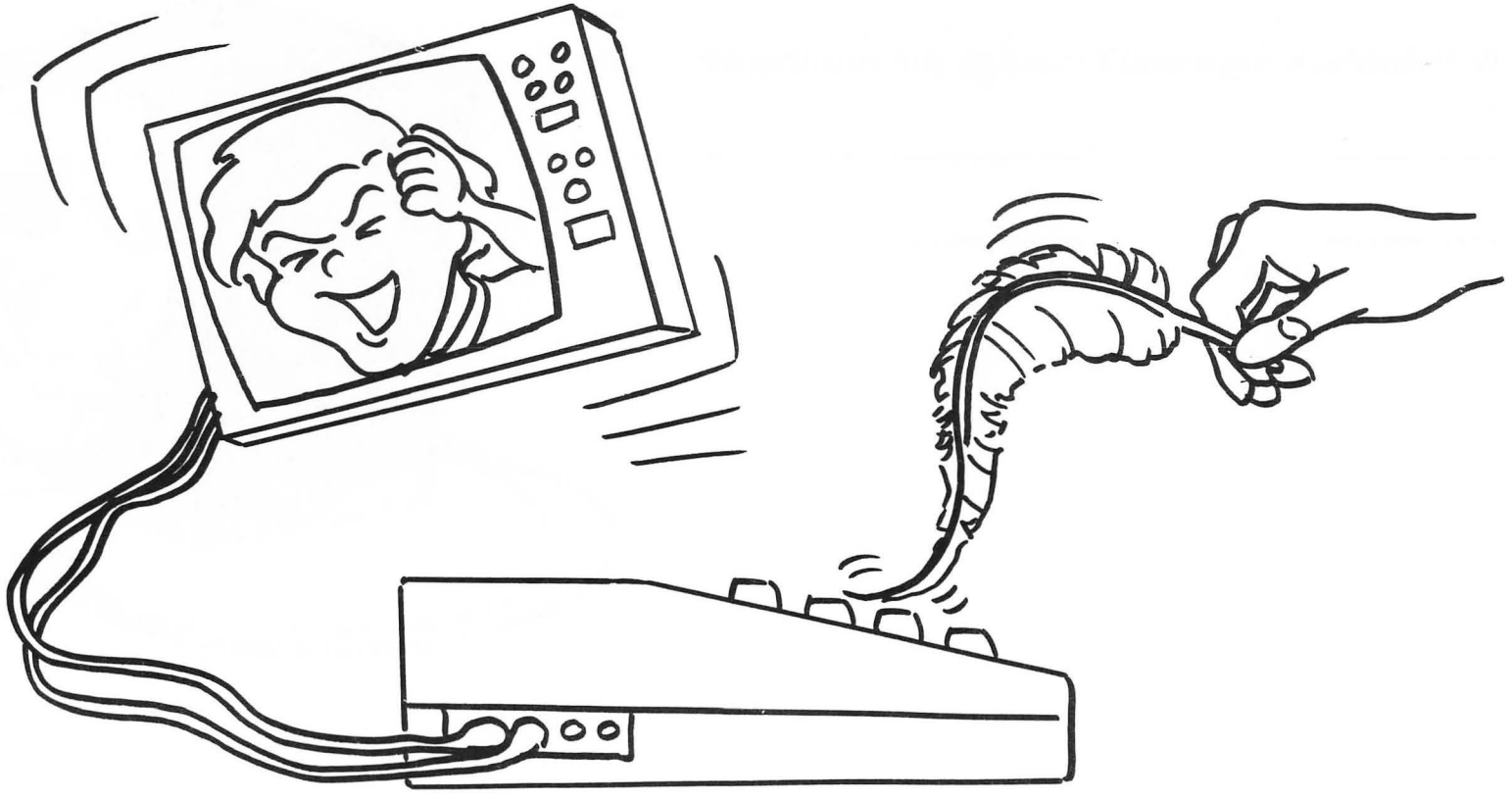


PROBLEM 24 FUNNYBONES AND SORE TOES

Try this one. In line 30 use the code number for C.

```
10 PRINT "IF YOU TOUCH THE C"
20 PRINT "I'LL SCREAM"

30 IF PEEK(764) = _____ THEN PRINT
"EEEK!"
40 GOTO 30
```

Change the scream to a call for HELP.



Give the computer a "funny key"; make it giggle when its funny key is touched. You select the key and use its code number in line 30. Write your funny-key program here.

10 _____

20 _____

30 _____

40 _____

Give the computer a sore key too. Make it say OUCH when the sore key is pressed. Use line 35 for the sore-key test.

35 _____

PROBLEM 25 QUIZ TIME

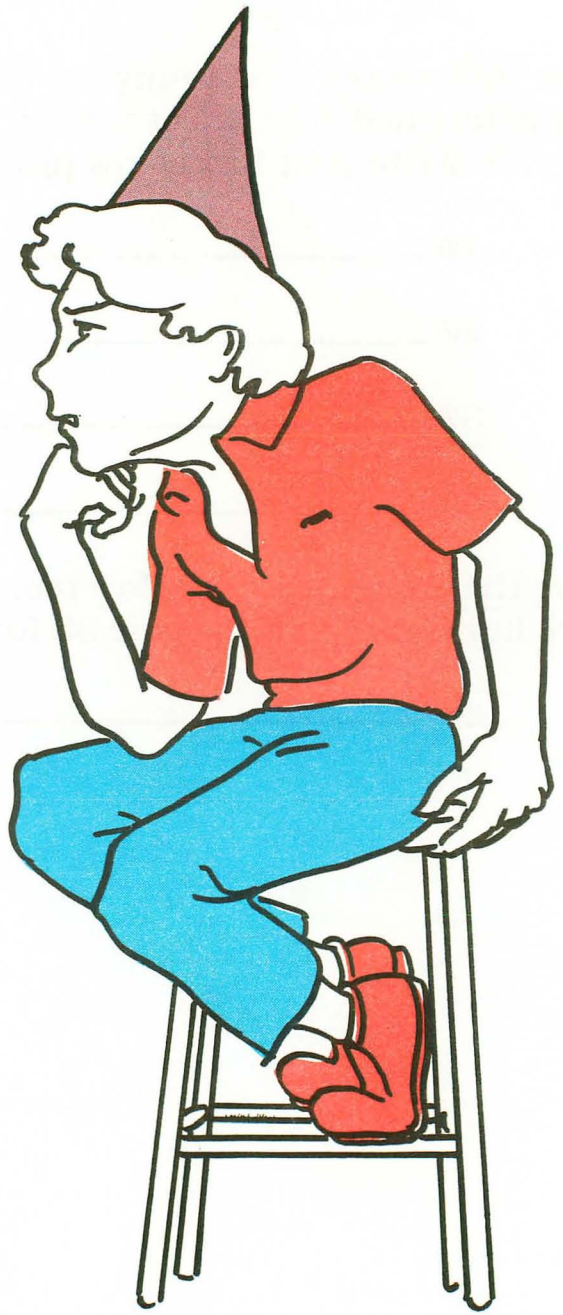
Read the program. How many questions are on the

quiz? _____

Complete line 90 so that the color question is asked again, if the answer is not GREEN.

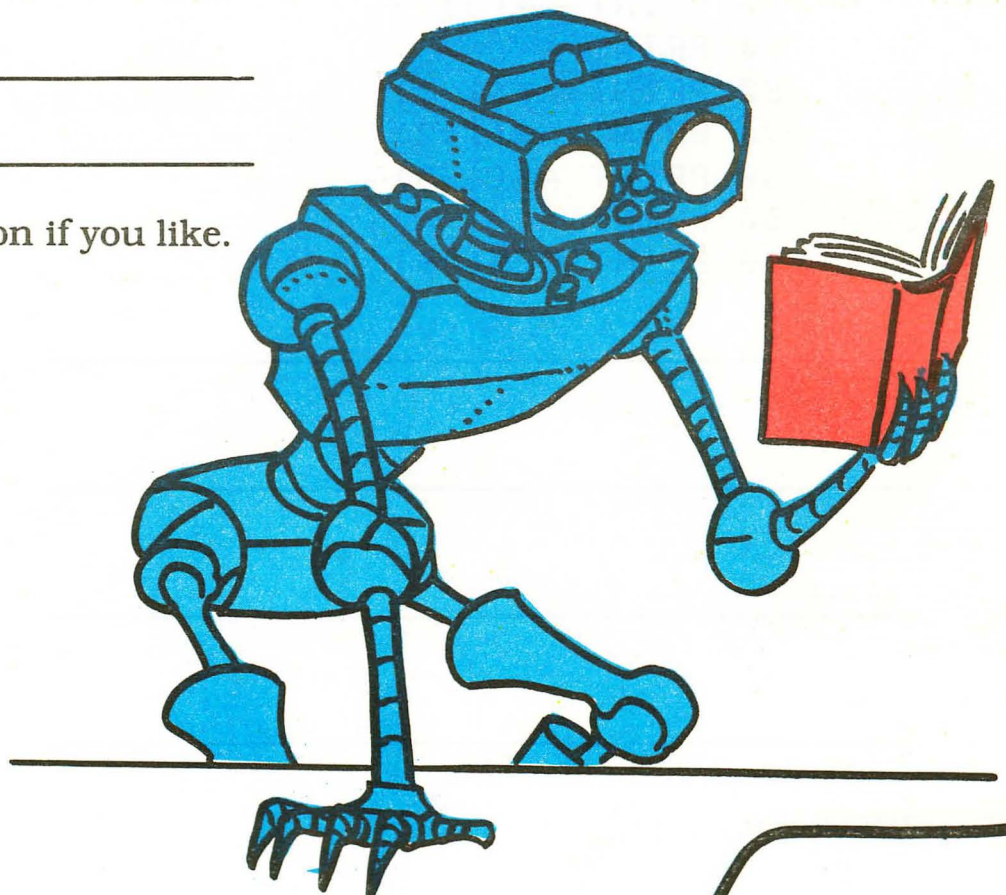
Then make up the last question on the quiz; use lines 120 – 150.

```
10 DIM A$(40)
20 PRINT "↻ QUIZ TIME"
30 PRINT
40 PRINT "WHAT IS THE CAPITAL OF
MICHIGAN?"
50 INPUT A$
60 IF A$ = "LANSING" THEN PRINT "RIGHT"
```



```
70 IF A$ <> "LANSING" THEN PRINT "NO,
LANSING"
80 PRINT "WHAT DO BLUE AND YELLOW MAKE?"
90 INPUT A$
100 IF A$ = "GREEN" THEN PRINT "GOOD"
110 IF A$ <> "GREEN" THEN PRINT "TRY

AGAIN": GOTO _____

120 PRINT _____
130 INPUT A$
140 IF A$ = _____

150 IF A$ <> _____
```

Have a friend try your quiz. Add another question if you like.



PROBLEM 26 PIECES OF WORDS

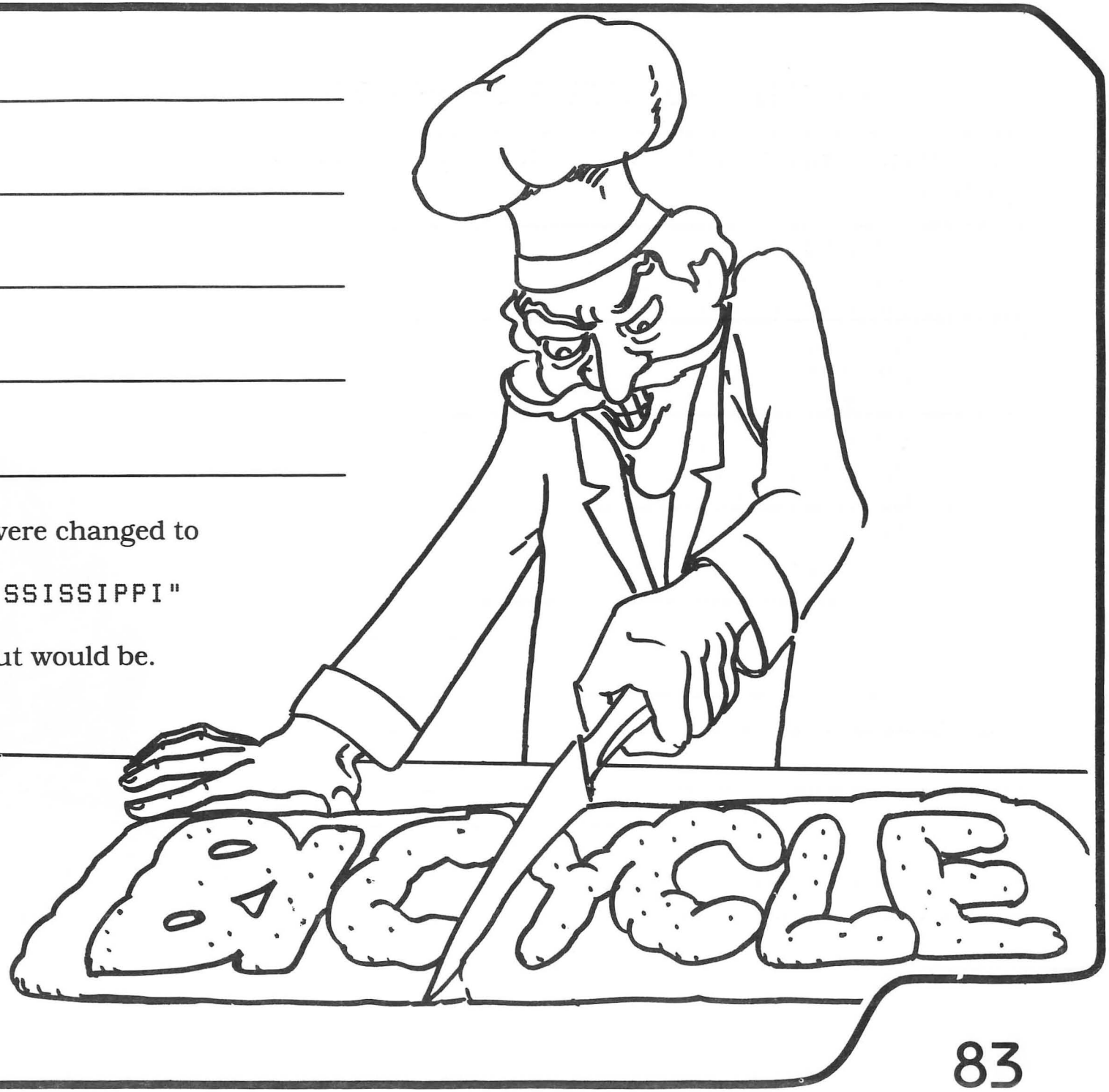
Run this program. Copy the output, then try to answer the questions.

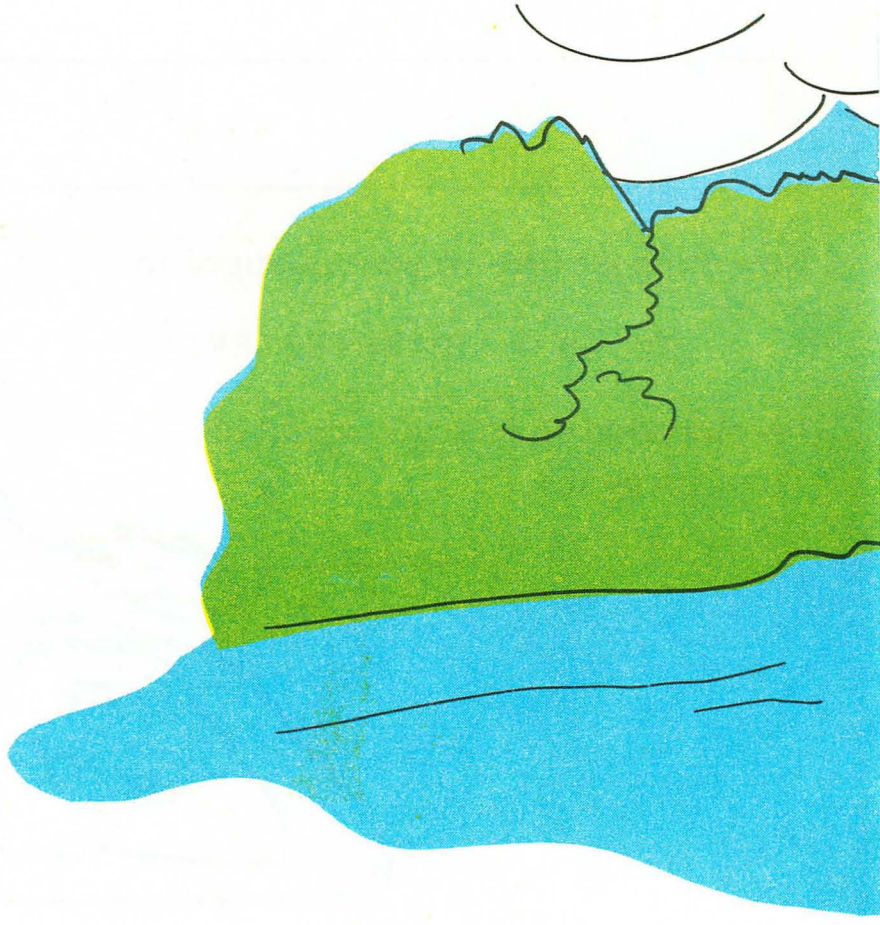
```
  5 DIM A$(20)
 10 A$ = "BICYCLE"
 20 PRINT A$(1)
 30 PRINT A$(2)
 40 PRINT A$(3)
 50 PRINT
 60 PRINT A$(1,1)
 70 PRINT A$(1,2)
 80 PRINT A$(1,3)
 90 PRINT
100 PRINT LEN(A$)
```

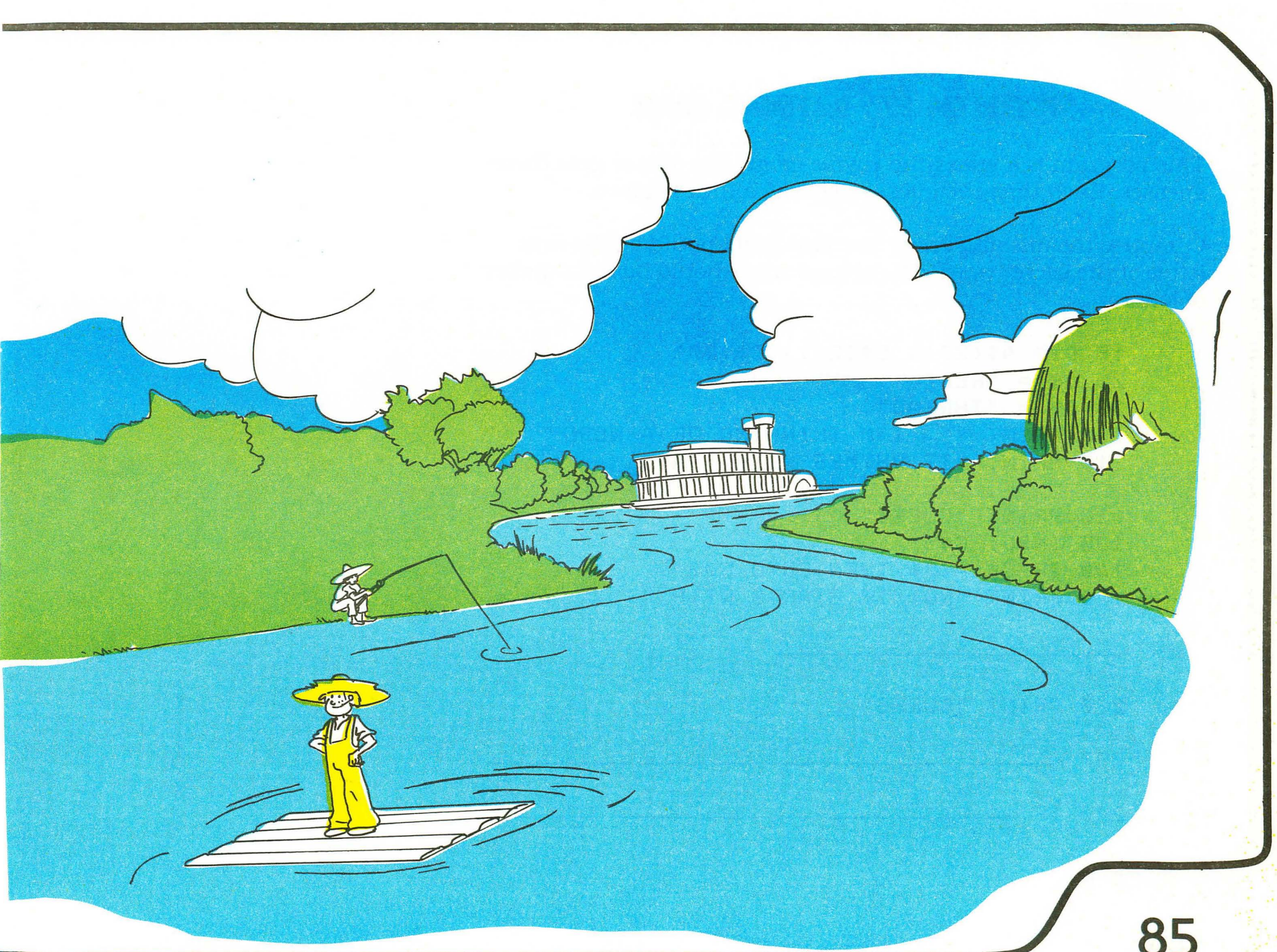
Now suppose line 10 were changed to

```
10 A$ = "MISSISSIPPI"
```

Predict what the output would be.
Then check it.







PROBLEM 27 RHYMING GAME

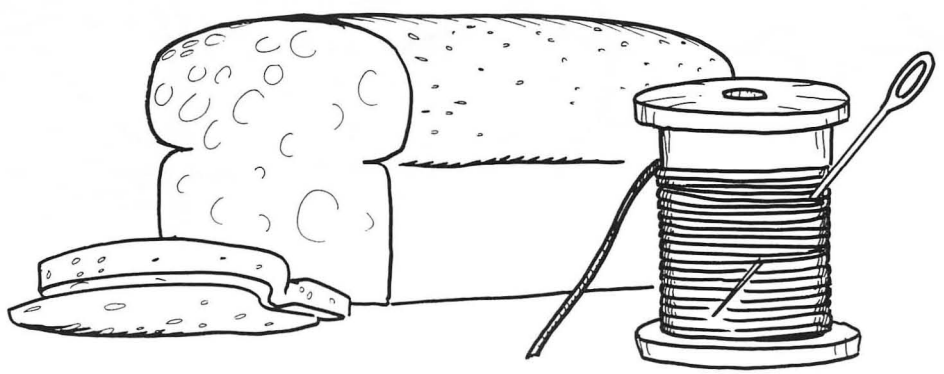
This program is a guess-the-rhyme game. The player gets three guesses. Each time the guess is wrong, a hint is given.

Complete the missing lines. The first hint should be the first letter of the secret word. The second hint should be the number of letters.

```
10 DIM A$(20), B$(20), C$(20)
20 A$ = "RED"
30 B$ = "THREAD"
100 PRINT "↻ I'M THINKING OF A WORD"
110 PRINT "IT RHYMES WITH ";A$

120 PRINT "GUESS"
130 INPUT G$
140 IF G$ = B$ THEN GOTO 500
150 PRINT "NO. IT STARTS WITH ";

_____

220 PRINT "GUESS"

230 _____

240 IF _____
```

```
250 ____________________________________

320 PRINT "LAST GUESS"

330 ____________________________________

340 ____________________________________

350 PRINT "NO, IT WAS ";

_____________________________

360 END

500 PRINT ____________________________________
```

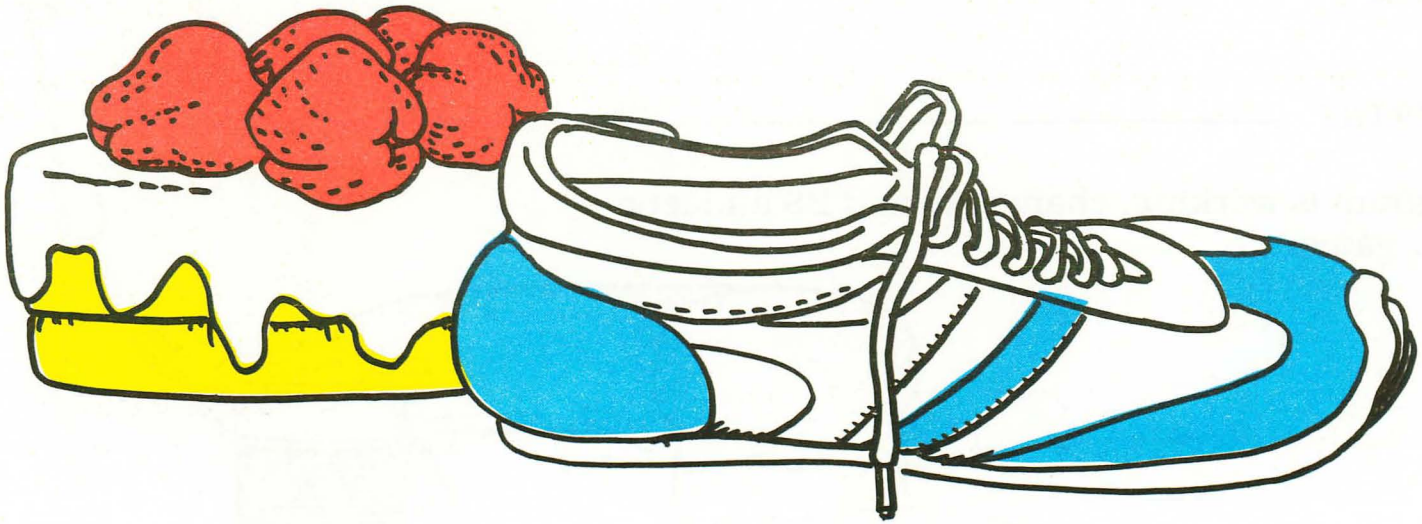
When the program is working, change A$ and B$ and let a friend try your game.



PROBLEM 28 WORD PUZZLES

Look at this program. PREDICT the output. Then check it, by entering and running the program.

```
10 DIM A$(20), B$(20)
20 A$ = "STAY"
30 B$ = "RED"
40 PRINT A$(1,3);B$(1,1)
50 PRINT B$(1,1);A$(3),A$(1,1)
```



Prediction _____

_____

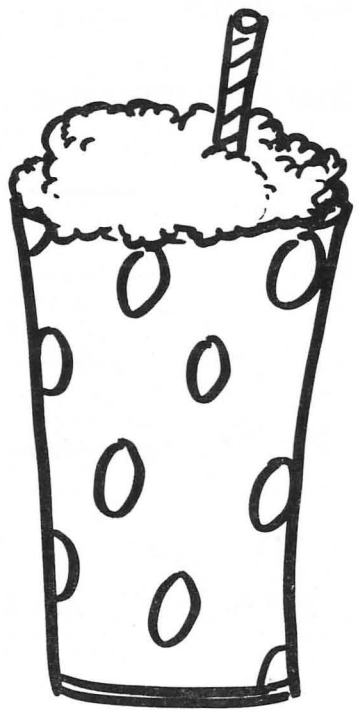
Now change A$ to "STRAWBERRY"
change B$ to "SHORTCAKE"

Change lines 40 and 50 so that the output is

```
BERRY
SHAKES
```

40 _____

50 _____



PROBLEM 29 DOING MATH

Enter and run this program:

```
10 PRINT "TELL ME A NUMBER"
20 PRINT "AND I'LL DOUBLE IT'
30 INPUT N
40 PRINT 2*N
```

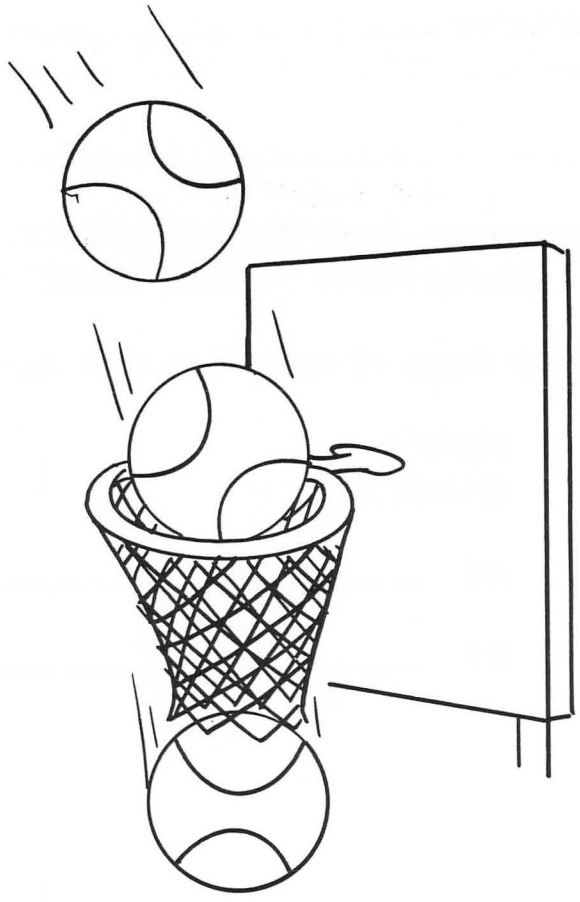
Check that the computer can double these numbers:

3 _____

20 _____

0 _____

-5 _____



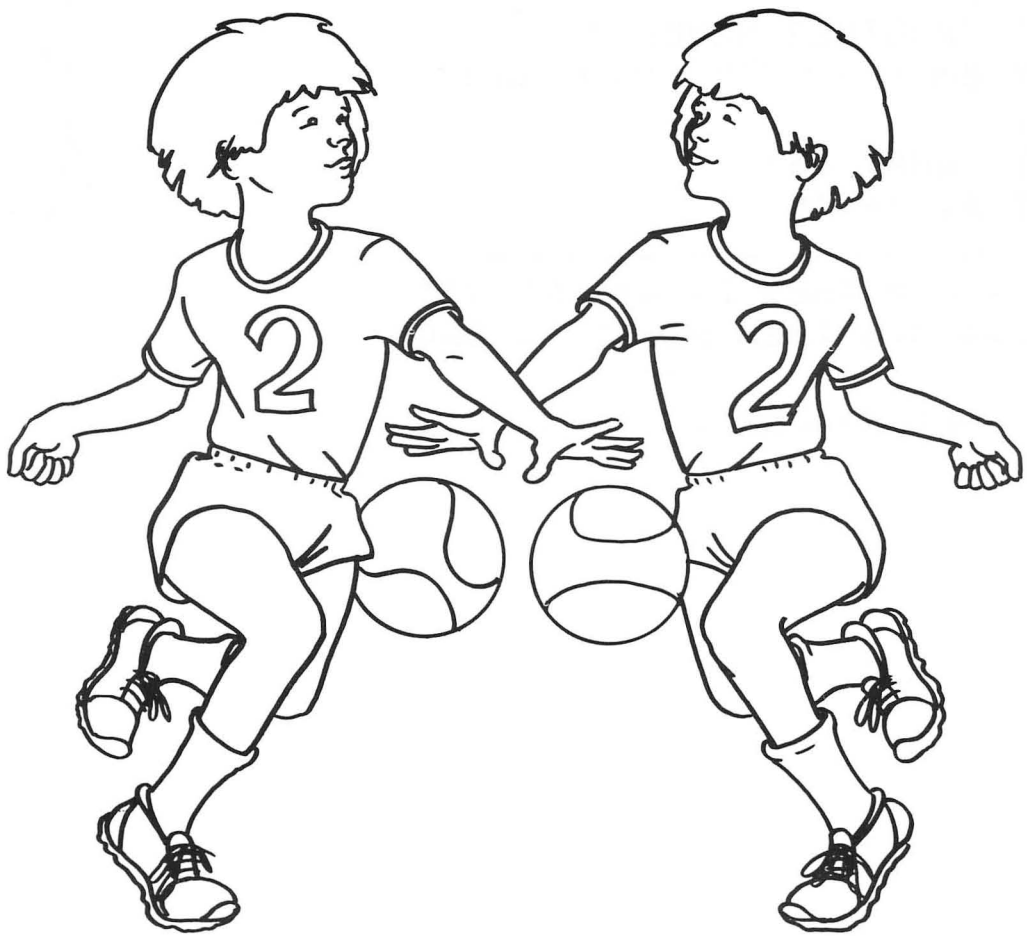
Make the computer triple the number. Try it out on some numbers and write down the results.

(If you tried an enormous number, such as

10000000000

your answer came in "Scientific Notation."

2E + 10 means a 2 with 10 zeros after it.)



PROBLEM 30 MATH TEST

Math quiz time. Run this addition quiz.

```
 10 PRINT "↻ MATH TEST"
 20 PRINT
 30 PRINT "TELL ME A NUMBER"
 40 INPUT A
 50 PRINT "ANOTHER NUMBER"
 60 INPUT B
 70 PRINT
 80 PRINT "WHAT IS"
 90 PRINT A; "+";B
100 INPUT C
110 IF C = A+B THEN PRINT "RIGHT"
120 IF C <> A+B THEN PRINT "WRONG"
```

Change this to a multiplication test.

What lines need to be changed?

_____

_____

_____

Make sure your changes are working.

PROBLEM 31 TIMES TABLES

This time we'll make the computer multiply the first ten counting numbers by 2.

Enter and run this program:

```
10 PRINT " ↶ TIMES TABLES"
20 FOR N = 1 TO 10
30 PRINT N, 2*N
40 NEXT N
```

We can make the output look like a math book,

```
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
```

by changing line 30. We want "2 x" and " = " to be printed each time. Change line 30 to

```
30 PRINT "2 x "; N; "="; 2*N
```

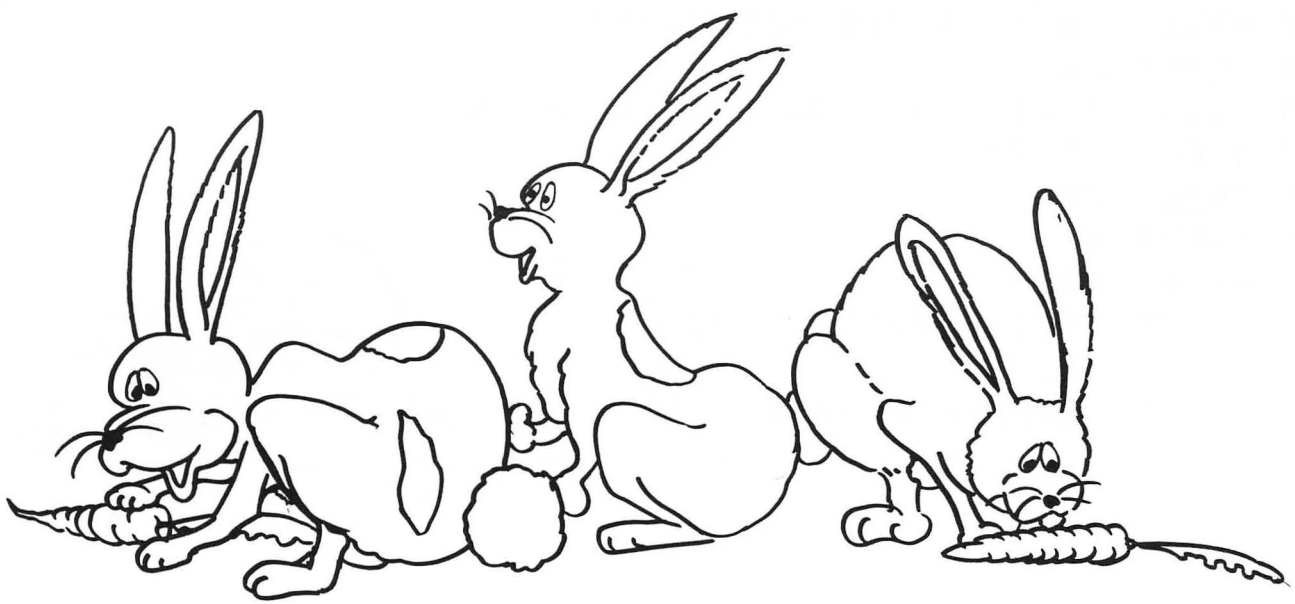
and run the program.

Now change line 30 so the computer does the SEVEN's table:

30 _____

Make another change to do the TWELVE's table. How much is

12 × 9? _____



PROBLEM 32 DAYS AND NIGHTS

If you spend two hours a day watching television, how many 24-hour days is that equal to?

If you sleep eight hours a night, how much sleep is that each year?

The computer can be taught to solve problems like these.

Read this program.

```
 50 DIM A$(40)
100 PRINT " ↻ "
110 PRINT "WHAT'S THE ACTIVITY?"
120 INPUT A$
130 PRINT "↓TELL ME HOW MANY HOURS YOU
SPEND EACH DAY"
140 PRINT A$
150 INPUT H
160 HY = H*365
170 PRINT "TOTAL HOURS ";HY
180 CD = HY/24
190 PRINT "COMPLETE DAYS ";CD
```

There are four variables in this program. Match their names with what they stand for:

A$	hours each day
CD	hours each year
HY	the activity
H	total days and nights

Run the program for activities you do such as:

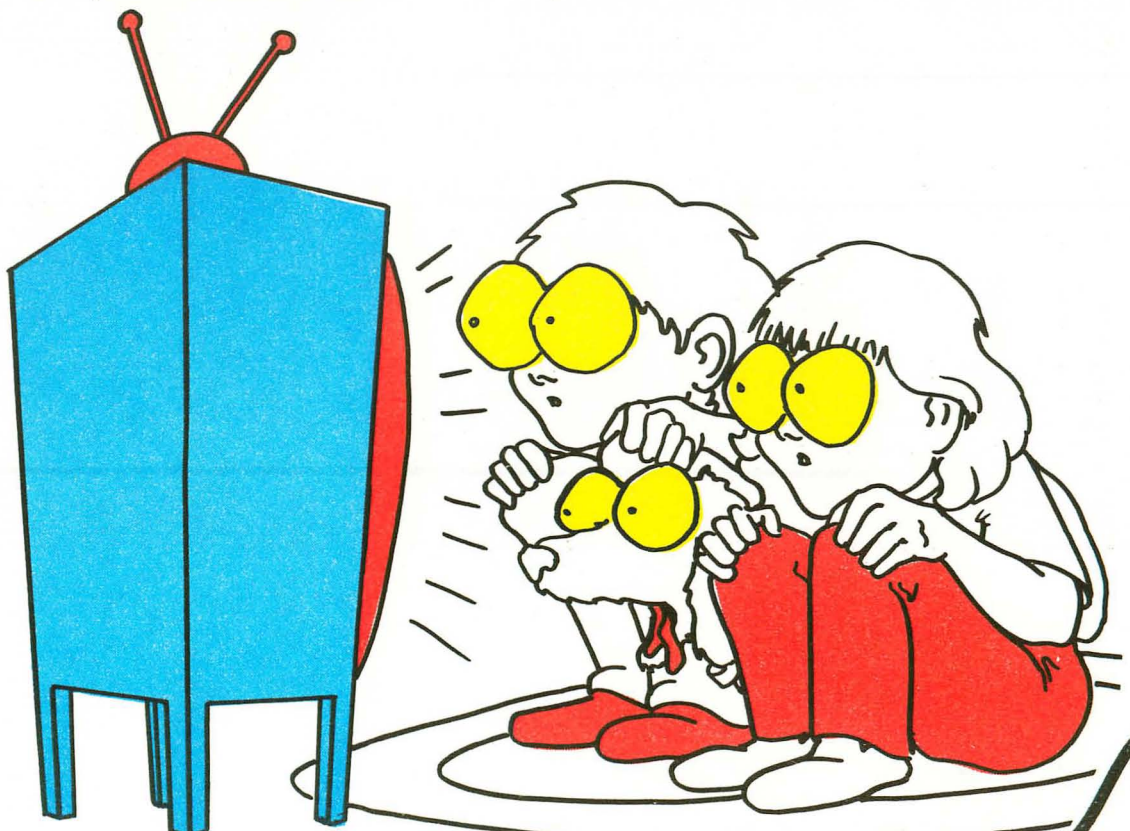
Playing the piano

Watching television

Sleeping

Reading

Looking for your socks



What things do YOU spend more than seven days each year doing?

_____

How do you convert 14 days to two weeks?

_____

Try to add two lines to the program to change CD, the number of complete days, to CW, the number of weeks.

200 _____

210 _____

_____

PROBLEM 33 SOUNDING OFF

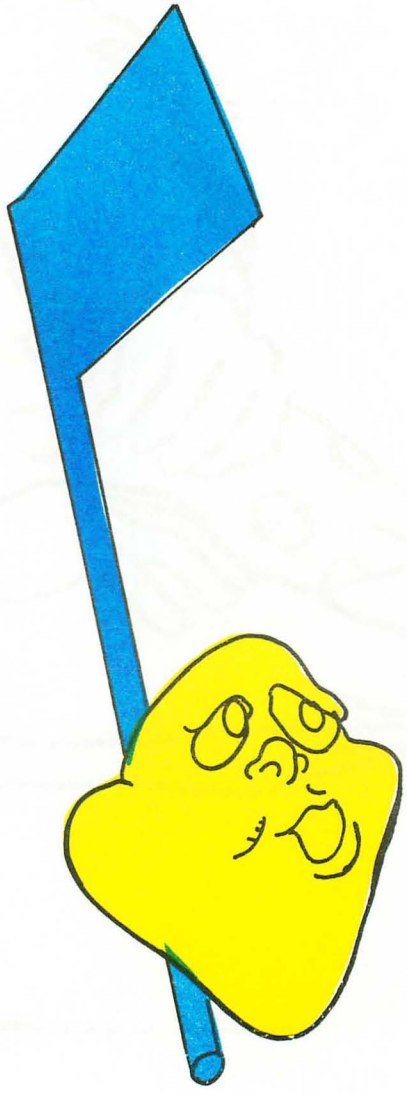
The way to turn OFF the sound is to use END. END can be in your program, or you can type it directly and press RETURN.

On the ATARI, there is a BASIC command called SOUND. This program will make a single, short sound:

```
10 PRINT "ENTER NOTE NUMBER"
20 INPUT N
30 SOUND 0,N,10,10
40 FOR K = 1 TO 100
50 NEXT K
60 END
```

Run the program a few times. Each time give it a number between 0 and 255. A 72 should match the A above middle C on the piano.

As the numbers get larger, do the notes get higher or lower?



Once a note starts, it keeps playing until either:

the program comes to another SOUND, or

the program comes to an END.

Here's how to change the program so you can enter four notes:

```
10 FOR Z = 1 TO 4
20 INPUT N
30 SOUND 0,N,10, 10
40 FOR K = TO 100
50 NEXT K
55 NEXT Z
60 END
```

Try these notes: 72, 108, 85, 164 (Press RETURN after each number.)

Try other sequences; change line 10 for more notes.

To make "automatic" music, we can use a special FOR...NEXT loop.

```
10 FOR N = 10 TO 40
20 SOUND 0,N,10,10
30 FOR K = 1 TO 100
40 NEXT K
50 NEXT N
60 END
```

Try changing the 100 in line 30, first to make the sounds shorter, then longer. You can change the numbers in line 10, too.

The keyboard code numbers can be used to make some weird music. Enter this program:

```
10 FOR N = 1 TO 50
20 N = PEEK(764)
30 SOUND 0,N,10,10
40 FOR K = 1 TO 100
50 NEXT K
60 NEXT N
70 END
```

Type different letters while this is running. For color, add the line

```
25 SETCOLOR 2,N,0
```

The faster you type, the more notes you'll hear. See what your name sounds like.

PROBLEM 34 COUNTDOWN

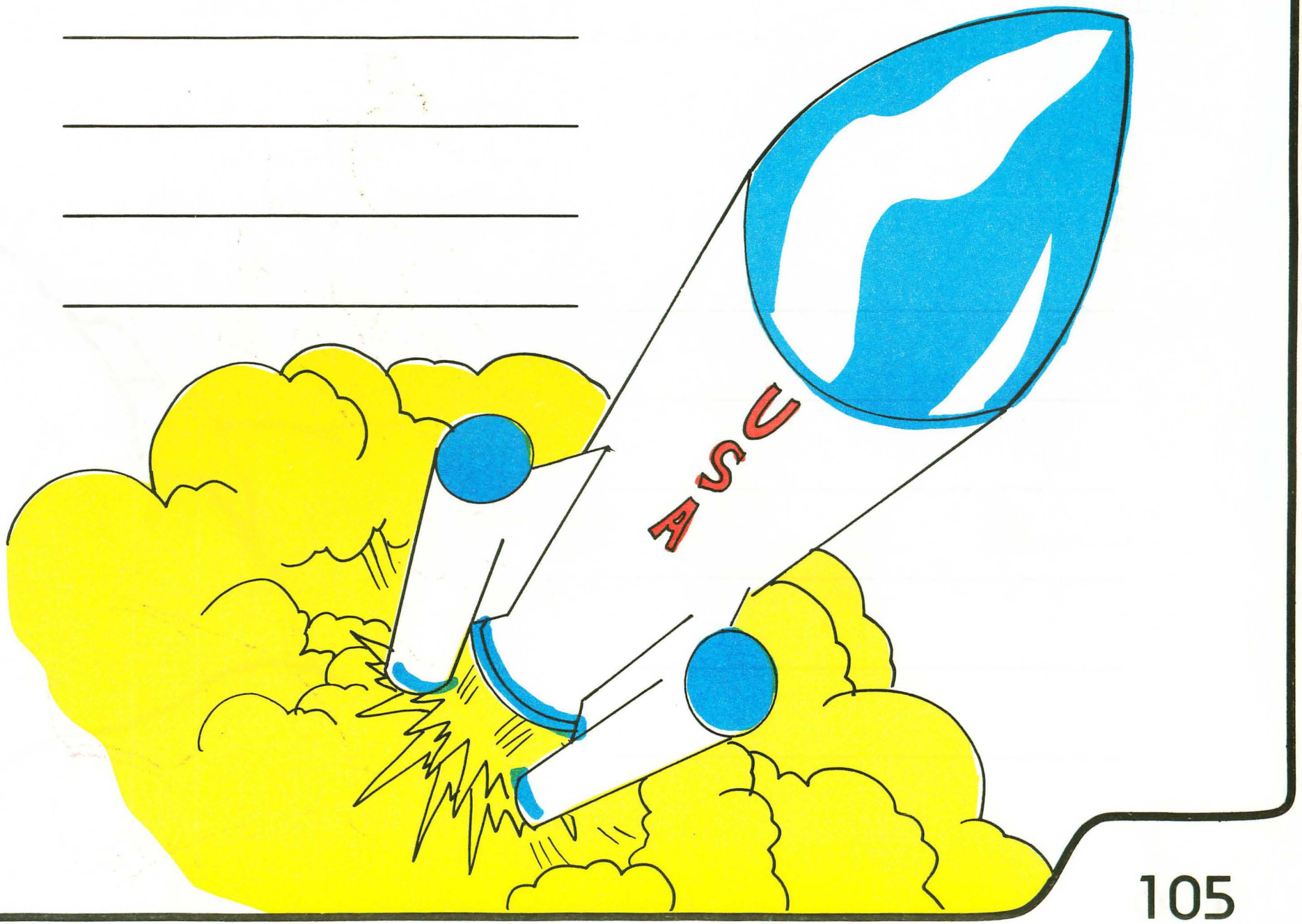
The computer can count backwards if you tell it to go backwards. Try this program:

```
100 FOR K = 5 TO 1 STEP -1
110 PRINT K
120 NEXT K
```

Write a program that will print:

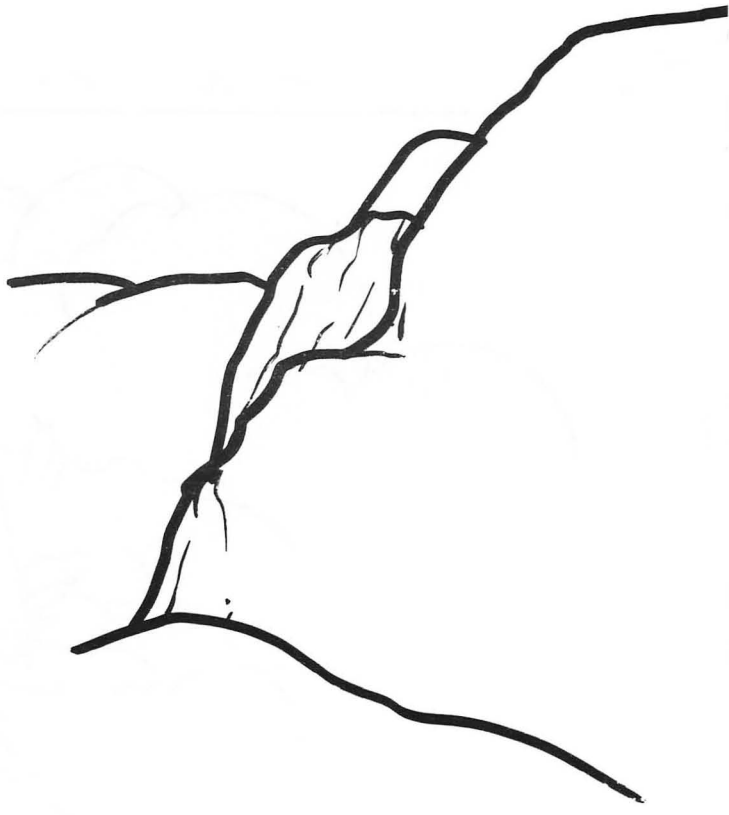
```
10
9
8
.
.
.
1
0
BLASTOFF!
```

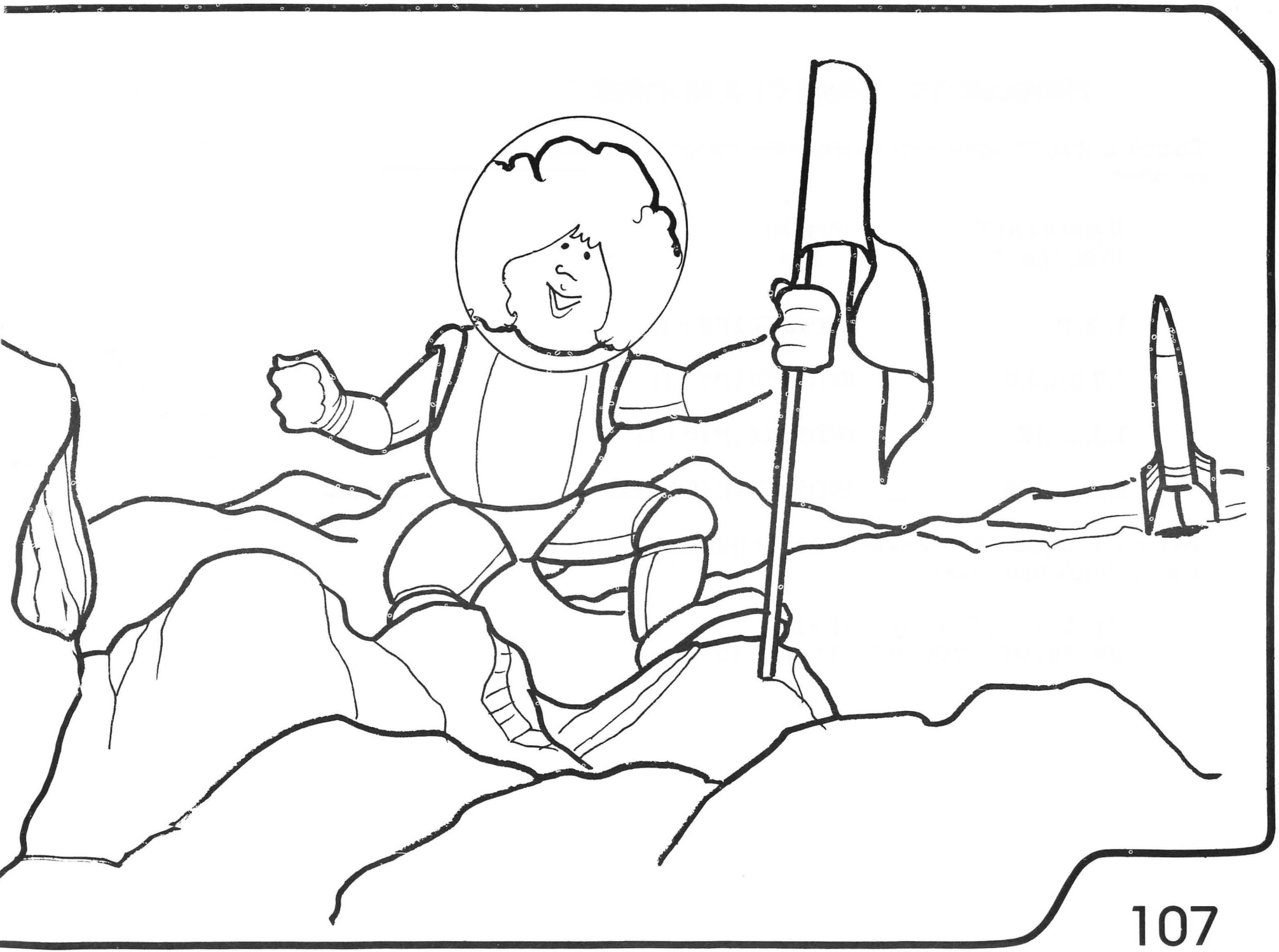
Copy your program here.



Write a program that counts from 1 to 4 and then back down to 1, like this:

```
1
2
3
4
3
2
1
```





PROBLEM 35 THINK OF A NUMBER

There's a special expression to make the computer "think" of a number.

If you want X to be one of	then let X =
1, 2, 3	INT(RND(1)*3+1)
1,2,3,4,5,6	INT(RND(1)*6+1)
1,2,....,10	INT(RND(1)*10+1)
1,2,....,100	INT(RND(1)*100+1)

To roll a die, you want the computer to think of a number from 1 to 6. Here's how to do it:

```
20 D = INT(RND(1)*6+1)
30 PRINT "YOU ROLLED A ";D
```

Either run this program six times or add lines 15 and 35 to make a FOR...NEXT loop to repeat the roll:

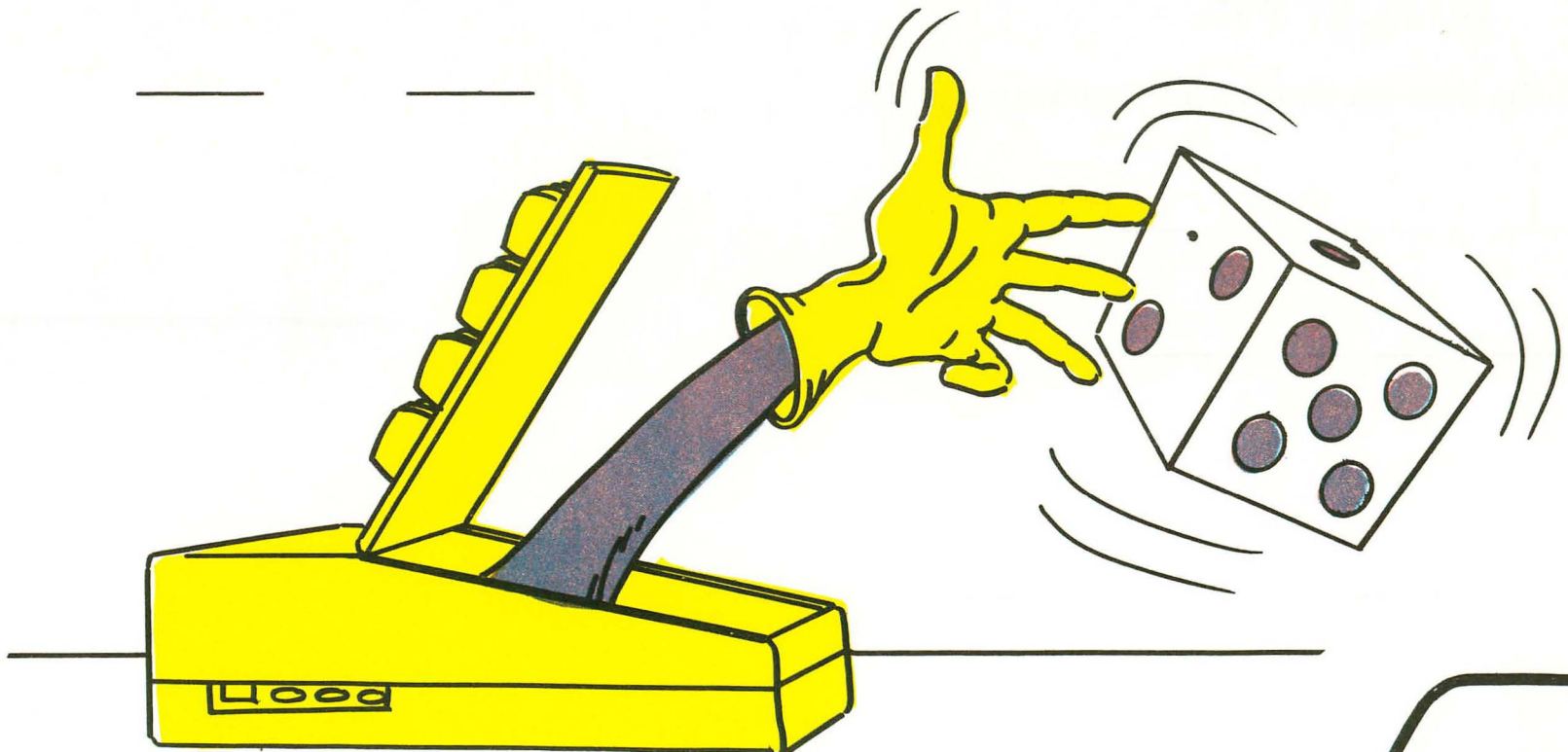
15 FOR _____

35 NEXT _____

Record your rolls here:

_____ _____ _____

_____ _____ _____



PROBLEM 36 TWO DICE

This time we want to roll TWO dice. Each time we roll a die, we must use

```
INT(RND(1)*6+1)
```

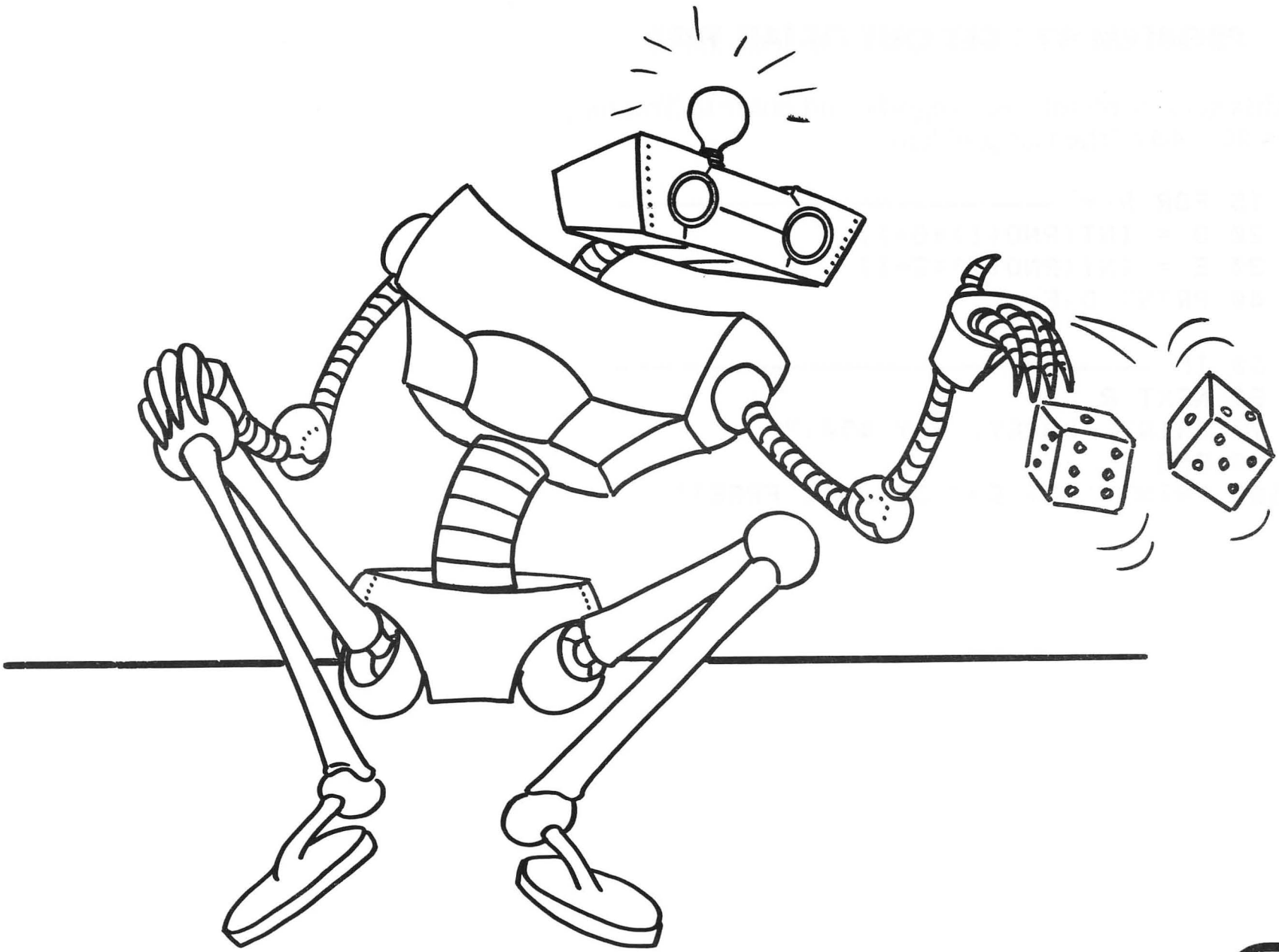
Enter this program. Either run it six times or put in a FOR....NEXT loop.

```
20 D = INT(RND(1)*6+1)
30 E = INT(RND(1)*6+1)
40 PRINT D,E
```

Roll the dice six times and copy your results:

_____ _____ _____

_____ _____ _____



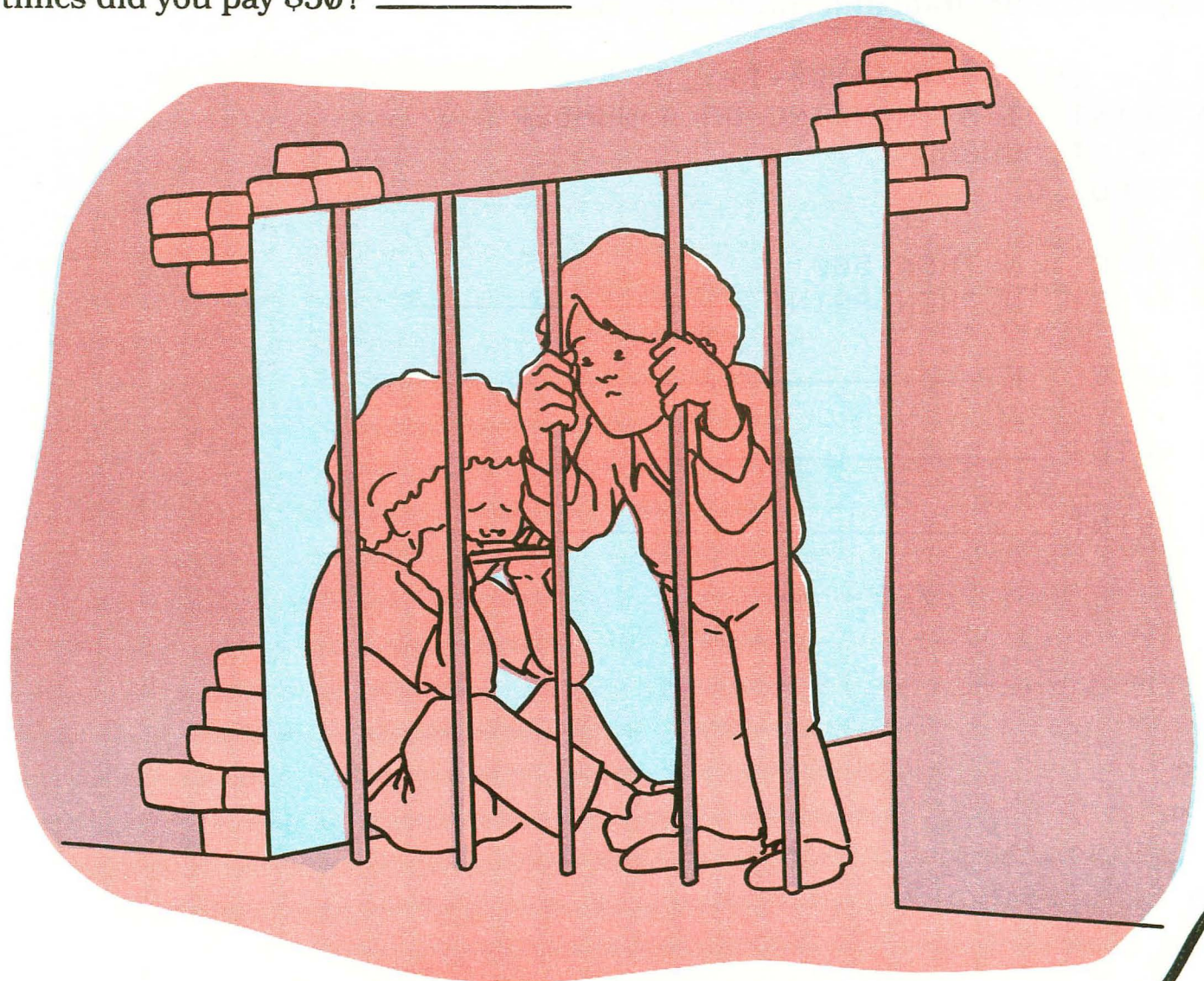
PROBLEM 37 GET OUT OF JAIL FREE

Finish this get-out-of-jail-free program and enter it. You may use lines 20 – 40 of the last problem.

```
 15 FOR R = ____________________________
 20 D = INT(RND(1)*6+1)
 30 E = INT(RND(1)*6+1)
 40 PRINT D,E

 50 IF ____________________________
 60 NEXT R
 70 PRINT "SORRY. PAY $50."
 80 END
100 PRINT "GET OUT OF JAIL FREE!"
```

Run this program six times. How many times did you get out of jail free? _____

How many times did you pay $50? _____



PROBLEM 38 NUMBER GUESS

In this program, the computer "thinks" of a number and you try to guess it. If you guess wrong, it tells you TOO BIG or TOO SMALL. Complete the program and run it to check it out.

```
 20 N = INT(RND(1)*100+1)
 30 PRINT "I'M THINKING OF A NUMBER"
 40 PRINT "GUESS"
 50 INPUT G

 70 IF G = N THEN 500
 80 IF G < N THEN PRINT _____

 90 IF G > N _____

100 GOTO _____

500 PRINT _____
```

What numbers did the computer think of? _____

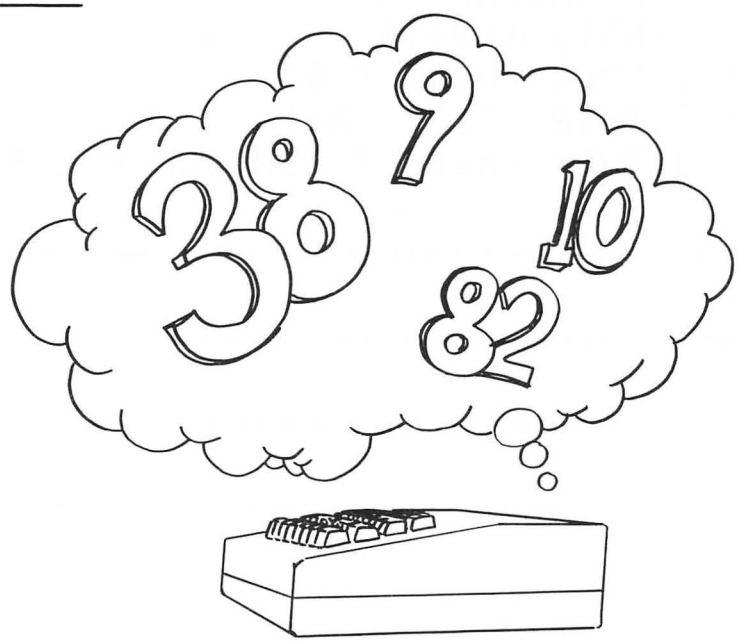
What's the biggest number it's allowed to think of? _____

To count the guesses, add these lines:

```
  5 C = 0
 60 C = C + 1
510 PRINT "IT TOOK YOU"
520 PRINT C; " GUESSES"
```

Run the program several times and keep track of how many guesses it took you.

_____



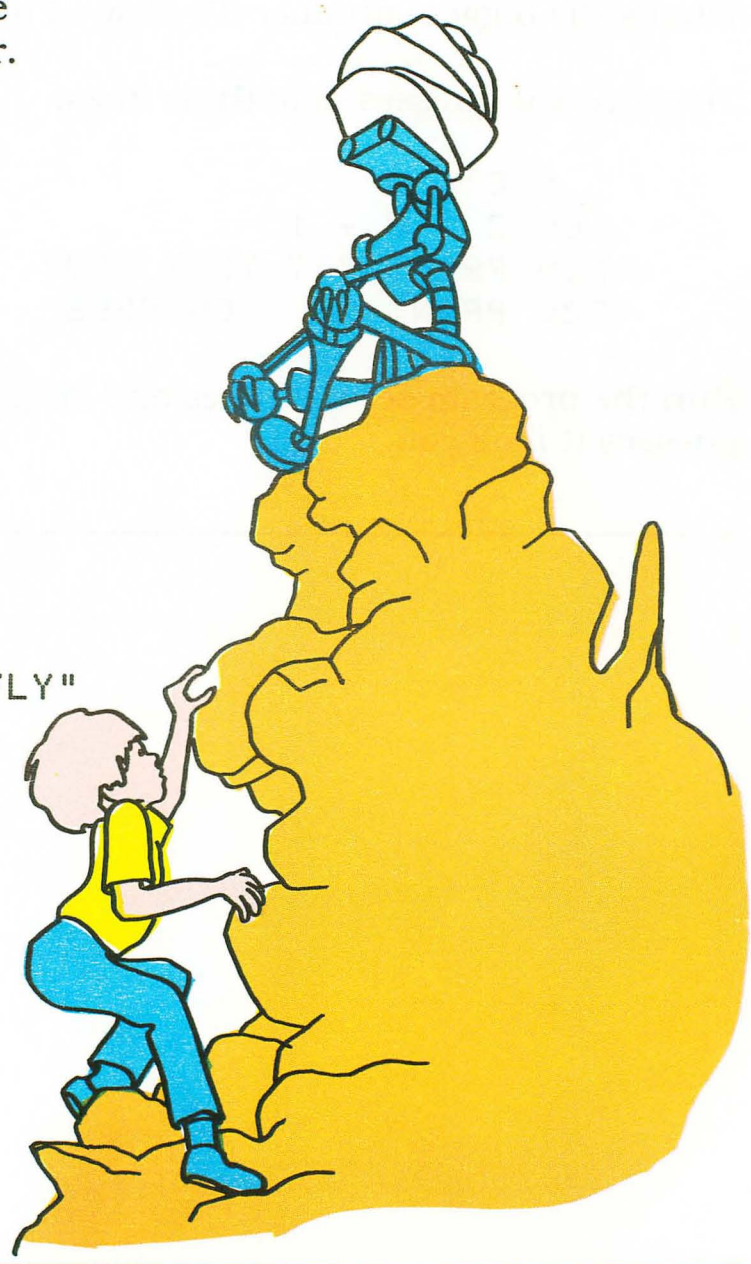
PROBLEM 39 THE ANSWER MACHINE

Is it going to rain tomorrow? Will I find my watch under the bed? Will my loose tooth come out today? Ask the computer.

We'll make up 10 answers and let the computer choose one each time a question is asked.

Finish the program and run it:

```
10 DIM Q$(50)
20 PRINT "QUESTION, PLEASE"
30 INPUT Q$
40 PRINT
50 PRINT "THE ANSWER IS"
60 N = INT(RND(1)*10+1)
70 IF N = 1 THEN PRINT "YES"
71 IF N = 2 THEN PRINT "NO"
72 IF N = 3 THEN PRINT "WHEN DONKEYS FLY"

73 _____

74 _____

75 _____

76 _____
```



```
77 _____
78 _____
79 _____
80 _____
90 GOTO 20
```

Put in a pause between lines 50 and 60 to make it look as though the computer is thinking about the problem:

```
55 _____
57 _____
```

Use lines 81 - 85 to add five more answers. Change line 60 so that the computer will think of a number from 1 to 15.

60 _____

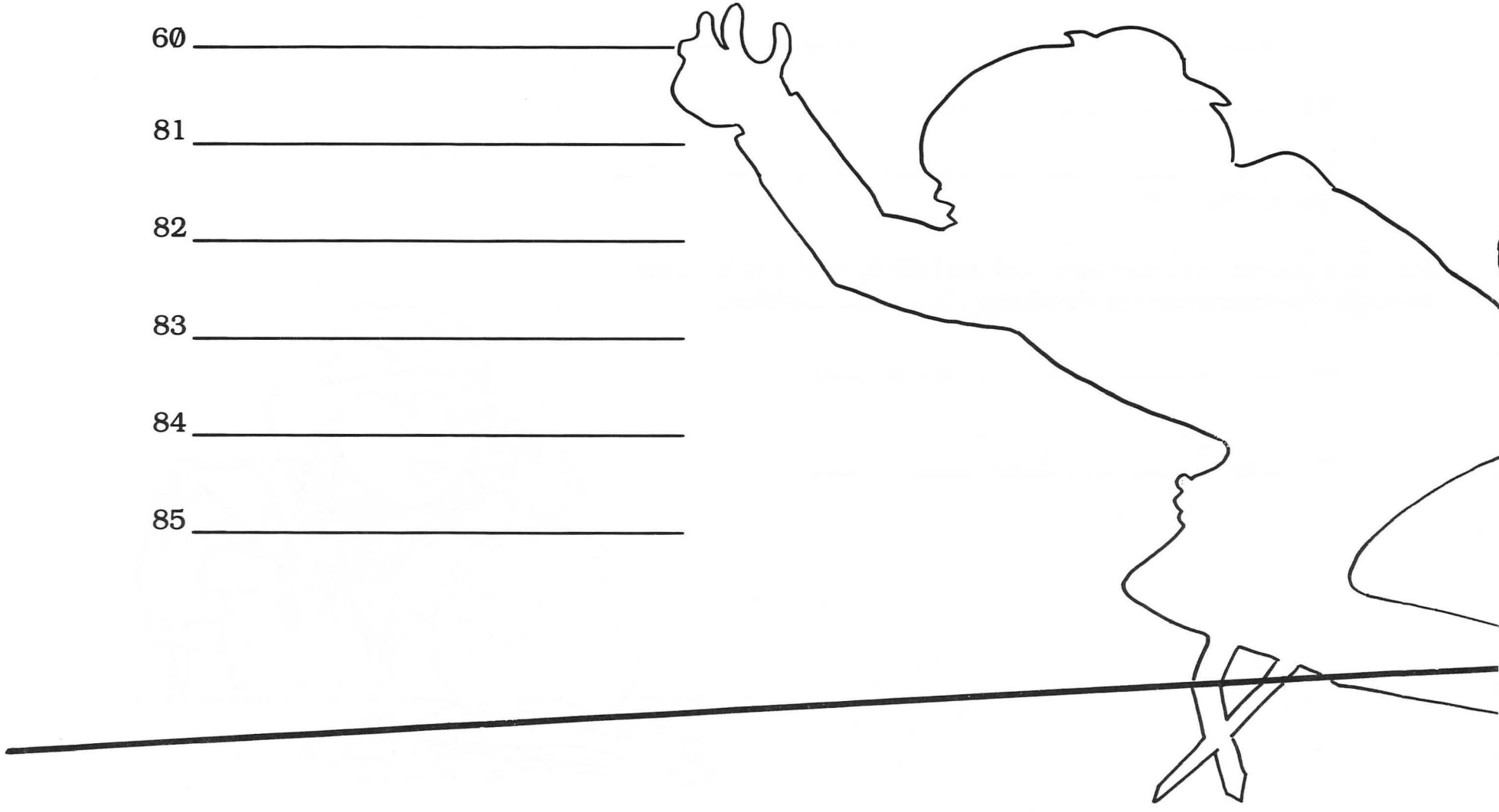
81 _____

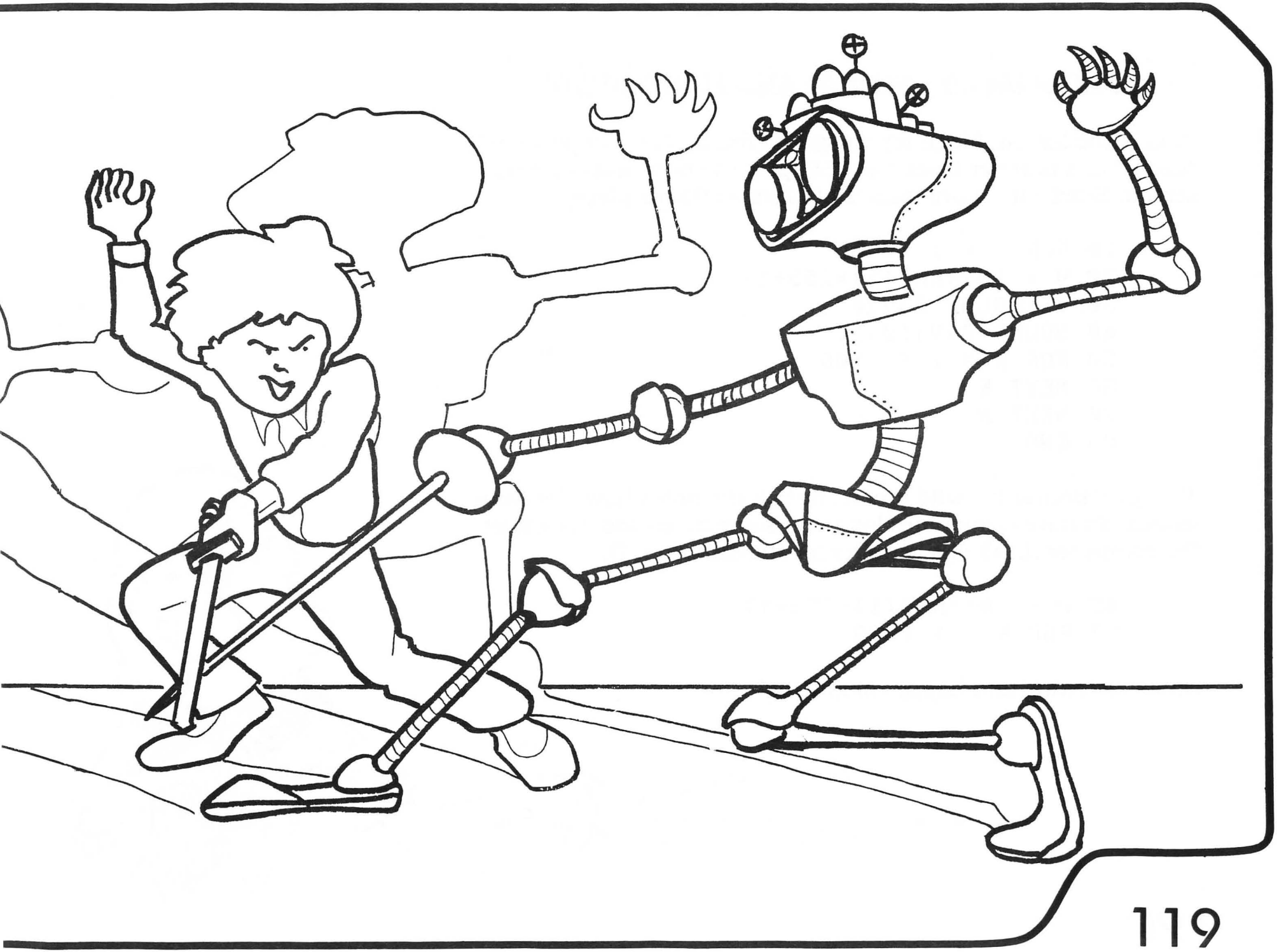
82 _____

83 _____

84 _____

85 _____





PROBLEM 40 SOUND AND LIGHT SHOW

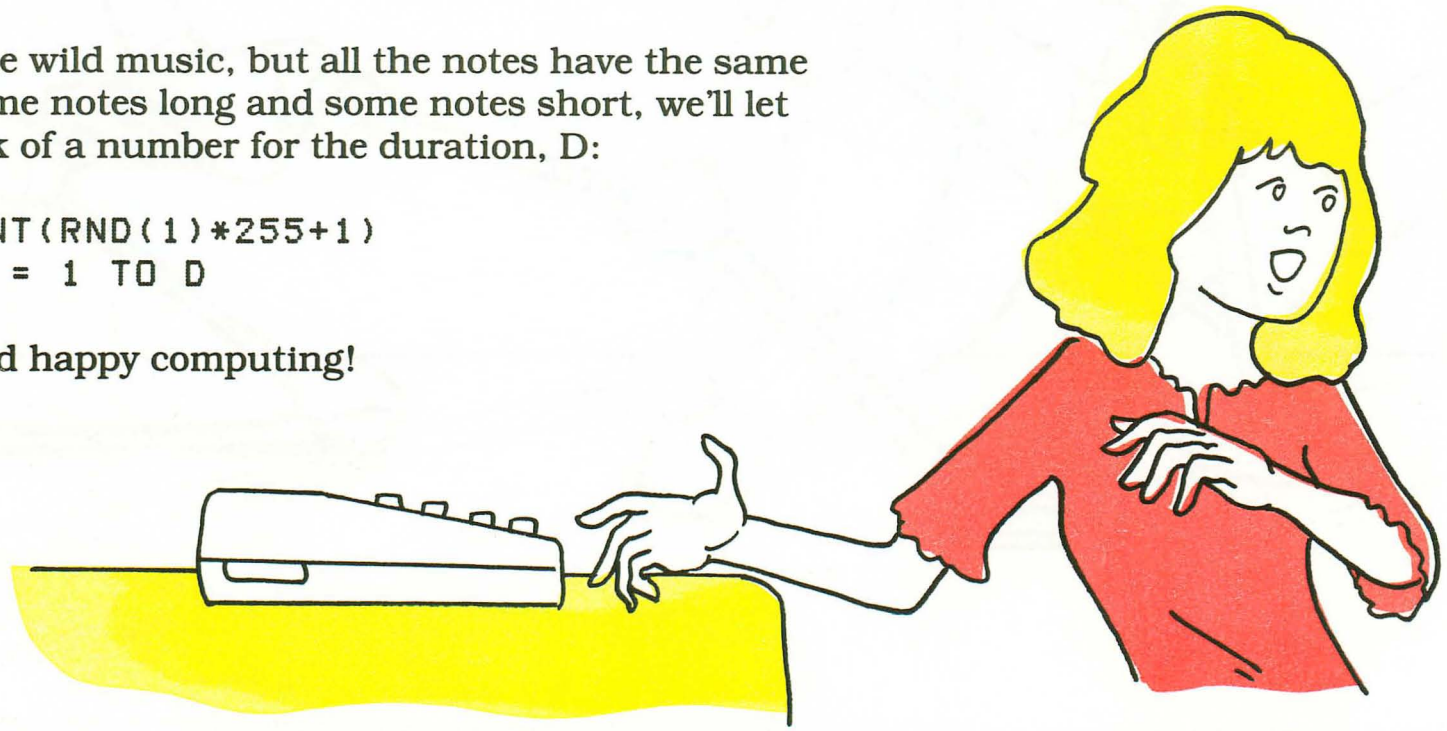
This is the last problem; it's the grand finale. The computer will "think" of a number from 1 to 255, play the note and color the screen. Start with a program that plays a 50 note piece.

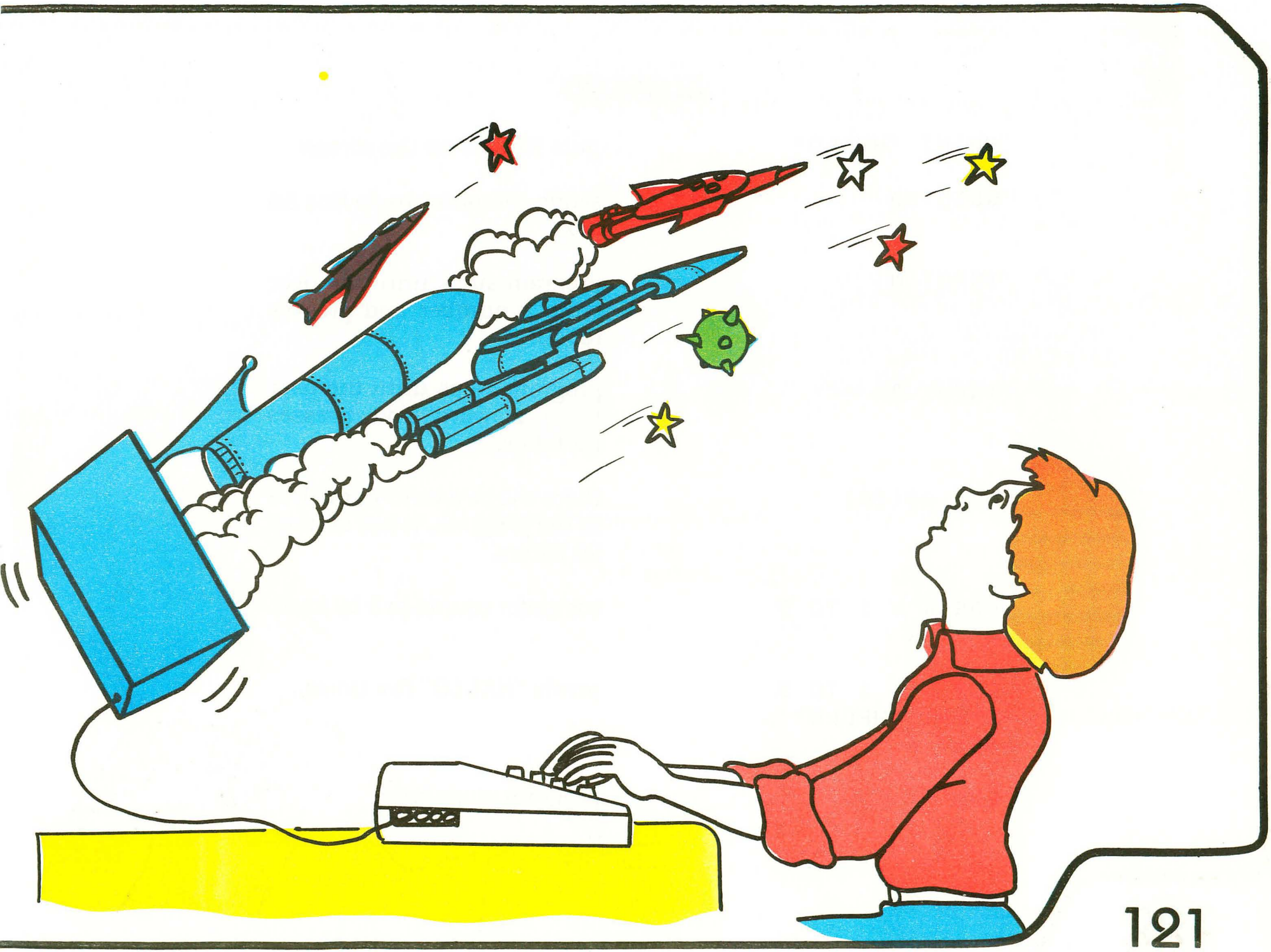
```
10 FOR N = 1 TO 50
20 V = INT(RND(1)*255+1)
30 SETCOLOR 2,V,0
40 SOUND 0,V,10,10
50 FOR K = 1 TO 100
60 NEXT K
70 NEXT N
80 END
```

This produces some wild music, but all the notes have the same length. To have some notes long and some notes short, we'll let the computer think of a number for the duration, D:

```
45 D = INT(RND(1)*255+1)
50 FOR K = 1 TO D
```

Happy listening and happy computing!





GLOSSARY

`PRINT "HELLO"`	puts HELLO on the screen.
`GOTO 50`	sends computer to do line 50 next.
`INPUT N`	program stops until the user types a number and presses RETURN.
`INPUT A$`	program stops until the user types a word and presses RETURN.
`DIM A$(20)`	there will be a word named A$ in the program. It has at most 20 letters.
`FOR K = 1 TO 5` `NEXT K`	computer counts to 5 by itself.

`FOR K = 1 TO 5`
`PRINT "HELLO"`
`NEXT K`
	prints "HELLO" five times.

`END`	program stops here.

```
IF A = 6 THEN...
IF A <> 6 THEN...
IF A$ = "HORSE" THEN...
IF A = 6 OR A = 7 THEN...
```
testing whether things are the same or different.

```
SETCOLOR 2,12,0
```
changes screen to color 12.

```
SETCOLOR 4,0,0,
```
changes border to color 0.

```
POKE 752,1
```
makes the cursor disappear.

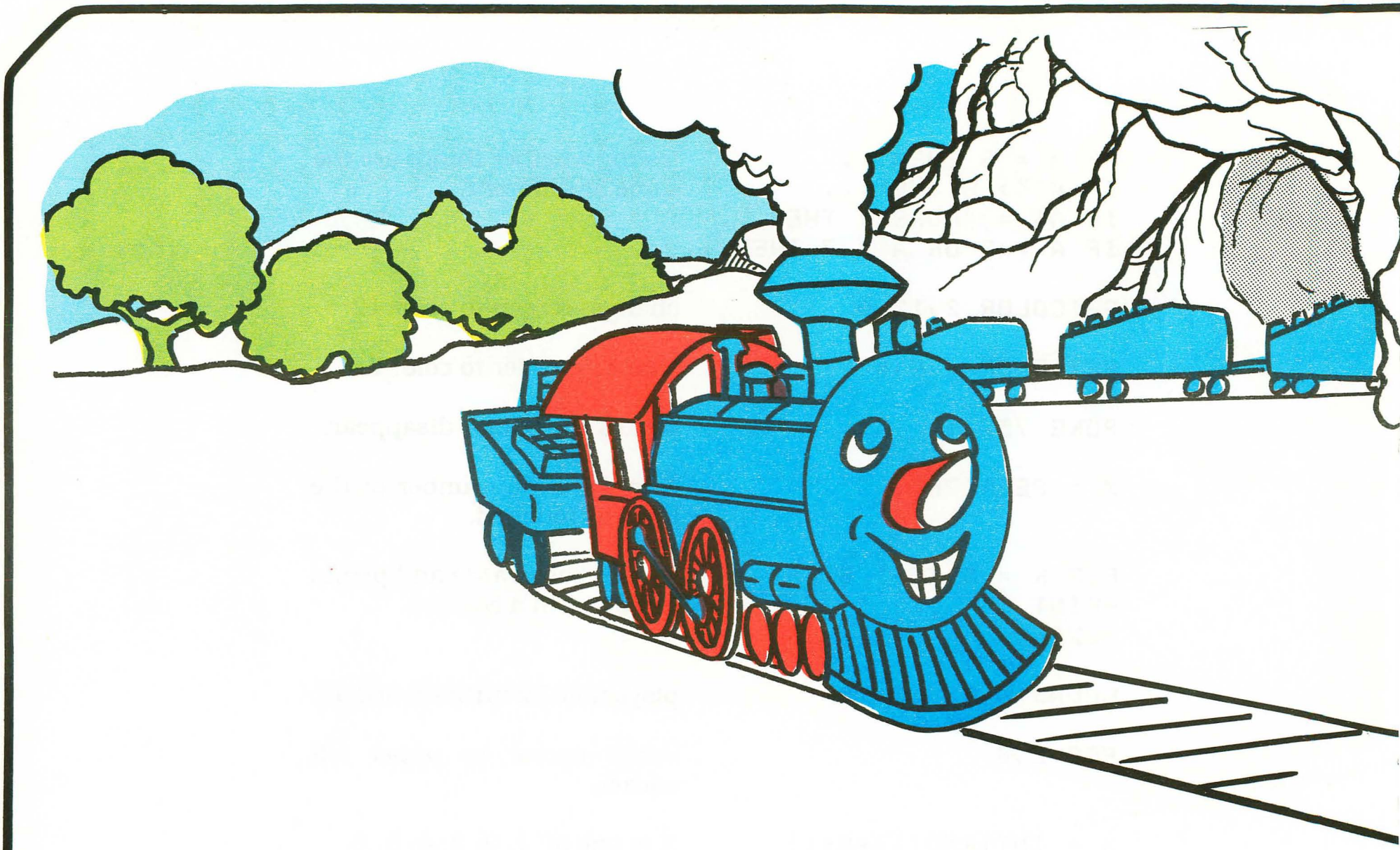
```
X = PEEK(764)
```
X is the code number of the last key touched.

```
FOR K = 5 TO 1 STEP -1
PRINT K
NEXT K
```
counts backwards and prints 5,4,3,2,1 in a column.

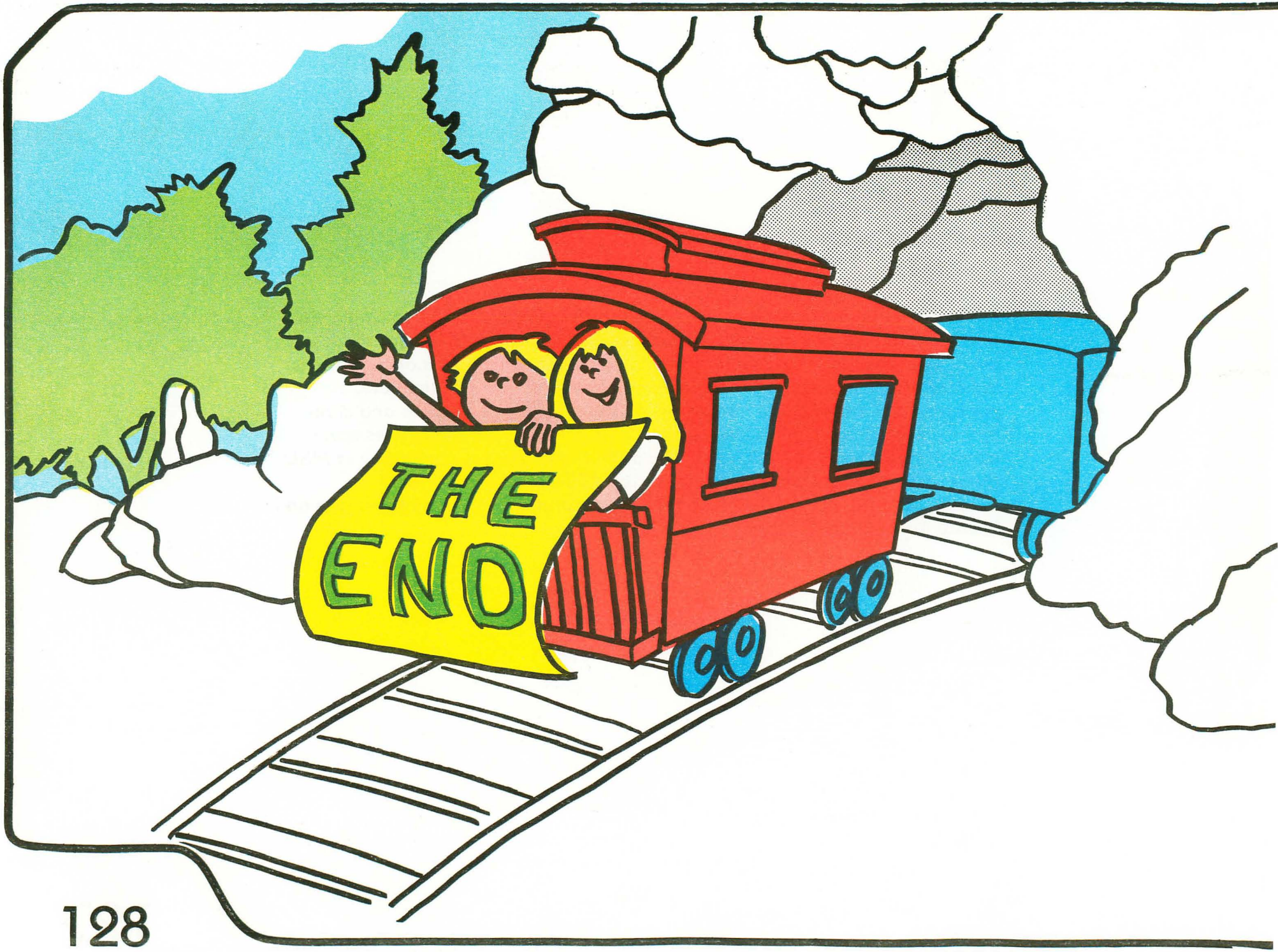
```
SOUND 0,72,10,10
```
plays note 72 until turned off.

```
POS.2,0
```
sends cursor to upper left corner.

```
X = INT(RND(1)*6+1)
```
X is one of: 1, 2, 3, 4, 5, 6.



*Mary Jean Winter is a professor of Mathematics at Michigan State University. She has been actively involved in mathematics education (K-12) for the last 8 years. She was formerly a numerical analyst in industry and academia. She is the author of Chivalry, Great Adventure, Wordspot and Witchnumber, published by Comm*Data; numerous articles on using calculators and computers in the classroom, as well as calculator activities for Scott Foresman's elementary series math text book. The author also teaches computing at MSU Summer Computer Camps for children ages 10-16. She has a Ph.D. in mathematics from Carnegie Mellon University and an A.B. in mathematics and history from Vassar College.*


THE
END

COMPUTER ACTIVITIES FOR THE YOUNG CHILD

COMPUTER PLAYGROUND, by Mary Jean Winter, is based on a child's natural interest in words, games and graphics. A collection of BASIC computer activities for the Atari 400/800/1200, this book is intended for children in grades 2 through 6.

Each activity is presented as a "Problem" in a workbook format. Children will find the problems interesting, fun and geared to their level of understanding. They'll type in and run sample programs — learn how to modify them — and complete partially written programs. Everything is kept simple so the child learns the programming process step-by-step.

As the child progresses, he or she will learn how to use many of the features of BASIC—PRINT, GOTO, INPUT, FOR . . . NEXT, IF . . . THEN, POKE and PEEK commands are a few. Working in this manner, the child will build up a collection of programs which are understandable and entertaining.

COMPUTER PLAYGROUND is illustrated with over 70 color cartoons and pictures to color. It's BASIC programming playtime!

COMPUTER PLAYGROUND is also available for the Commodore 64/VIC 20 and Apple II, II+ and //e computers.

OTHER POPULAR COMPUTER BOOKS BY DATAMOST:

Kids & the Apple
Kids & the Atari
Kids & the Panasonic
Kids & the TI 99/4A
Kids & the VIC 20
Kids & the Commodore 64
by Ed Carlson

How to Write an Apple Program
How to Write an IBM-PC Program
How to Write a TRS-80 Program
How to Write a Program Vol. II
by Ed Faulk

The Elementary Apple
The Elementary Atari 400, 800, 1200
The Elementary Commodore 64
The Elementary Timex/Sinclair
The Elementary VIC 20
by William Sanders

Using 6502 Assembly Language
p-Source
by Randy Hyde

Games Apples Play
by Mike Weinstock & Mark Capella

ISBN 0-88190-190-3

DATAMOST™ INC
8943 Fullbright Avenue
Chatsworth, CA 91311-2750
(213) 709-1202

0 48831 00164