Atari® in Wonderland

HAYDEN

Atari® in Wonderland

Atari® in Wonderland

Fred D'Ignazio

HAYDEN BOOK COMPANY, INC.
Hasbrouck Heights, New Jersey

For my team of young programmers, Howard Boggess, Angela Bradshaw, Joni Burdette, Brian François, Beth Ann Hostutler, Howard Levine, Mack McGhee, Melissa Perdue, and Scott Rainey, and their hardworking coach David James.

Acquisitions Editor: GARY MARKMAN
Production Editor: MARSHALL E. OSTROW
Art Director: JIM BERNARD
Cover art by: SUSAN STURGILL/GEORGE BAQUERO
Illustrated by: NETWORK GRAPHICS/SUSAN STURGILL
Compositor: ART, COPY, & PRINT, INC.
Printed and bound by: ARCATA BOOK GROUP/FAIRFIELD GRAPHICS DIV.

Library of Congress Cataloging in Publication Data

D'Ignazio, Fred.
Atari in Wonderland.

Summary: Contains twenty-two short, simple games to use with an Atari computer on such topics as colors, sounds, multiplication, state names, spelling, fractions, and Spanish and French languages.
1. Computer games—Juvenile literature. 2. Atari computer—Programming—Juvenile literature. [1. Computer games. 2. Games. 3. Atari computer—Programming. 4. Programming (Computers)] I. Title
GV1469.2.D52 1983 794.8'2 83-18590
ISBN 0-8104-5771-7

2 3 4 5 6 7 8 9 PRINTING
83 84 85 86 87 88 89 90 91 YEAR

PREFACE

This is a book of learning games you can type in on your Atari computer. You and your children can use this book to learn with the computer and about the computer. And while you are learning you are going to have fun.

If your children need to practice their spelling, have them play Scrambled Bees. If they are learning about fractions, give them some practice dividing up a pizza pie in Pepperoni, Please!

There are also programs on phonetics (Fat Cat), colors and sounds (The Pied Piper), multiplication (The Hamburger Contest), learning state names (Al's Tour of the States), Spanish (Uno! Dos! Tres!), and French (Un! Deux! Trois!).

There is even a child-size word processor program called Book Report. Children can use it to write poems, stories, and, of course, book reports.

There are twenty-two learning games in this book. The games are grouped by subject area. They are educational, simple, and designed to engage your children's imaginations.

Most of the games are short. That means they are easy to enter into the computer. You can learn a lot about programming in BASIC by reading the description of each game and typing it in.

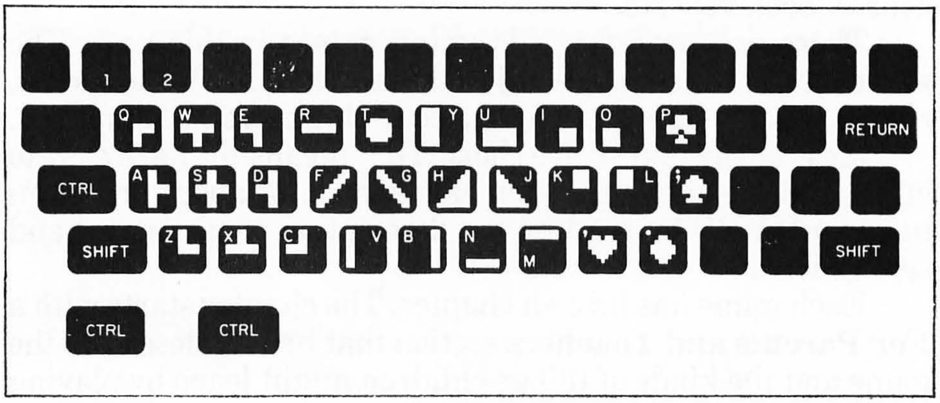
Each game has its own chapter. The chapter starts with a **For Parents and Teachers** section that briefly describes the game and the kinds of things children might learn by playing it.

Next is a **For Kids** section that weaves a story around each game and encourages children to use their imaginations when they play the game.

The Program comes next. It is the listing of the commands in the game. Following the program listing (in several chapters) is a **Typing Hints** section that explains how to type special parts of the program into the computer. Then comes a **Highlights** section that points out the major sections of the program. This is followed by a **Variables** section that lists all the variables (the pigeonholes in the computer's memory that store important numbers and letters). The chapter ends with a **Do-It-Yourself** section that gives you suggestions for expanding or enhancing the original game.

BASIC statements and commands in this book are written in capital letters: for example, FOR, NEXT, SAVE, GET, and END. References to buttons on the keyboard are written in bold capital letters: for example, **CTRL**, **BREAK**, and **RETURN**.

Many of the games use graphics (picture-making) characters. The **Typing Hints** sections tell you what keys to type to make these characters appear. For your handy reference, the Control Graphics Keyboard is shown here. When you use these special characters, remember to hold down the **CTRL** key while pressing the selected graphics key.



If you are using the Atari 600XL, 800XL, 1200XL, 1400XL, or 1450XLD rather than the Atari 400 or Atari 800, remember that the Atari-symbol key (⅄), which is used to get "reverse video" on the 400 and 800, is replaced by the ◩ key. On the Atari 600XL and 800XL this key is located at the lower righthand corner of the keyboard. On the Atari 1200XL, 1400XL, and 1450XLD this key is located above the keyboard near the righthand side.

Thanks to My Young Assistants . . .

Young people wrote several of the learning games in this book. My editor at Hayden, Gary Markman, suggested that I find some high school students to help me with the programming. I spoke to David James, the computer science instructor at Patrick Henry High School, which is only two blocks from my home, in Roanoke, Virginia. Within a few days David and his students were writing programs for this book.

The students came to school early and left late. They even got permission to leave other classes in order to work on the programs for this book. David spent several weeks carrying their one Atari computer back and forth between the school and the students' homes, so the students could work on the programs in the evenings and on weekends. At one point all the programs mysteriously disappeared.

But somehow, the programs all got done. To celebrate the completion of the programs and to thank the students, I took David and his team of young programmers out for a pizza dinner, compliments of Hayden.

According to David, this book is just the start for his students. Now they are anxious to begin working on new books and maybe even start their own "Learning Games" software company.

Make These Programs Your Own . . .

These programs are intentionally short and simple. That makes them easy to type in, easy to understand, and easy to use. It also makes them easy to modify. As you are typing them in, make these programs your own. For example, add comment lines (REM commands) **in your own words** that explain what each program does. Also, at the beginning of each game, add PRINT commands to explain the game rules to your children.

I hope you and your children have as much fun using this book as I had writing it!

Fred D'Ignazio

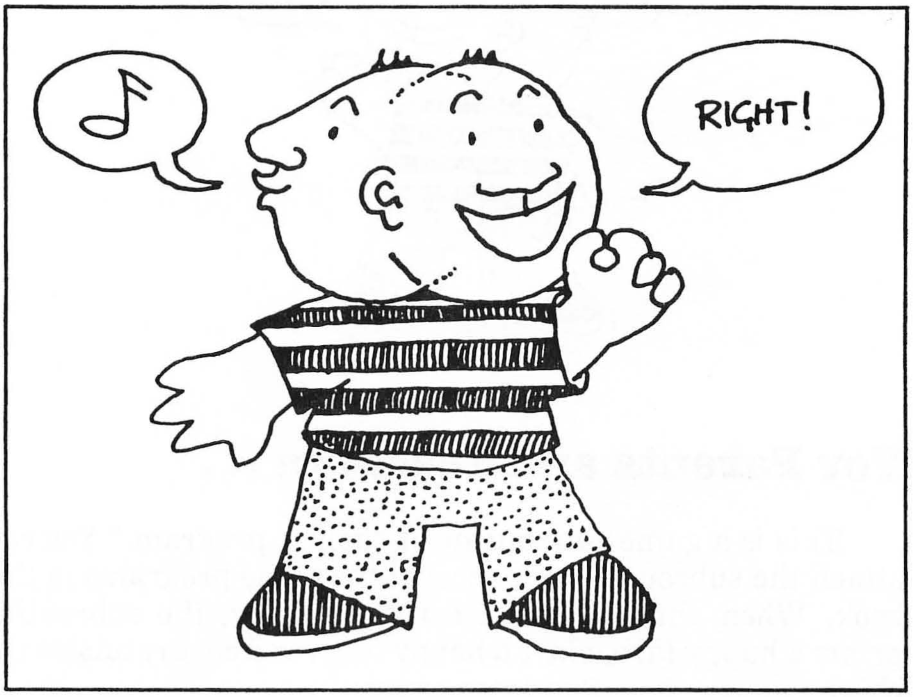
EQUIPMENT NEEDED

To use the programs in this book, you will need the following equipment:

- An Atari 400, 800, 600XL, 800XL, 1200XL, 1400XL, or 1450XLD, with 16K RAM (minimum).
- A black and white or, preferably, a color monitor (TV).
- An Atari 410 Program Recorder or an Atari 810 Disk Drive.

CONTENTS

FACES



1 HAPPY FACE

For Parents and Teachers ...

This is a game **subroutine** or "helper program." You can attach the subroutine onto most of the game programs in this book. When children get a correct answer, the subroutine prints a happy face, plays a happy tune, and congratulates the child.

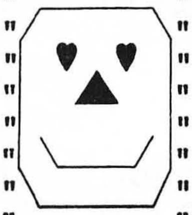
For Kids ...

When you do something right, you expect a smile and a word of praise, right?

Here's a little program that draws a happy face on the TV screen, prints out the message "RIGHT!" and plays a happy "whistle."

The Program . . .

Program Name: **HAPPY**



```
1000 REM ***HAPPY FACE & SOUND
1010 POSITION 14,12:PRINT "        "
1020 POSITION 14,13:PRINT "        "
1030 POSITION 14,14:PRINT "        "
1040 POSITION 14,15:PRINT "        "
1050 POSITION 14,16:PRINT "        "
1060 POSITION 14,17:PRINT "        "
1070 POSITION 14,18:PRINT "        "
1080 POSITION 14,19:PRINT "        "
1090 POSITION 15,21:PRINT "RIGHT!"
1110 FOR W=1 TO 10:FOR A=50 TO 25 STEP -
4:SOUND 0,A,10,10:FOR T=1 TO 2:NEXT T
1120 NEXT A:NEXT W:SOUND 0,0,0,0
1125 FOR I=12 TO 19:POSITION 14,I:PRINT
"        ":NEXT I:POSITION 15,21:PRINT "
     "
1130 RETURN
```

Typing Hints . . .

To make the happy face, you need to use graphics characters. The **Control Graphics Keyboard** is pictured in the Preface of this book.

You can type a graphics character two ways:

1. Hold the control key (**CTRL**) down and push the button with the character you want to type. For example, hold down **CTRL** and press the **T** key and you have a .

2. Hold the **CTRL** key down and push the **CAPS LOWR** key. Now you can type the special graphics characters without holding down the **CTRL** key. (To get out of this mode, hold the **SHIFT** key down and push the **CAPS LOWR** key again.)

Here are the graphics characters you will need to draw the happy face.

In line Number ...	To get ...	Which is (are) ...	Press ...
1010, 1060	—	top of face and mouth	CTRL N
1020	/ and \	left and right upper corners of face	CTRL F and CTRL G
1030, 1040 1050, 1060	| and |	left and right sides of face	CTRL V and CTRL B
1030	♥	eyes	CTRL ,
1040	◢ and ◣	left and right sides of nose	CTRL H and CTRL J
1060, 1070	\ and /	left and right corners of mouth and lower left and right corners of face	CTRL G and CTRL F
1080	—	bottom of face	CTRL M

Highlights ...

This is one of the simplest programs in the book. On lines 1010 to 1080, it prints out the happy face. On line 1090, it prints the message "RIGHT!" On lines 1110 to 1125 it makes the whistle sound.

The program uses the sound command—SOUND—to make the whistle.

Look at all the FOR and NEXT commands. These commands make the computer do something over and over. This is called a computer **loop**. The loop begins with the FOR command. It ends with the NEXT command. Every command between the FOR and NEXT commands is part of the loop.

The FOR command determines how many times the computer circles around the loop. For example, look at FOR W = 1 TO 10 on line 1110. This tells the computer to set the loop counter W to 1. Each time the computer races around the loop it adds one to the loop counter (W). The computer does ten loops (from W = 1 to W = 10). Then it goes on and does something else.

Also, note that the program starts on line 1000 and ends on line 1130 with the command RETURN. That is because this program is really a subroutine—a helper program. It is not supposed to run on its own. It is supposed to help another program. (You can still test this subroutine by itself. When you finish typing it in, just type GOSUB 1010.)

You will want to attach this subroutine onto the tail end of almost all the other programs in the book. That way, whenever you get the right answer in any of the programs, the computer will draw a happy face, whistle, and shout "RIGHT!"

Be sure to save this subroutine on a disk by using the LIST command, or by using the LIST "C:" command if you are using a tape recorder.

When you have typed in one of the other programs, copy this subroutine back into memory using the ENTER command if the subroutine is on disk or ENTER "C:" if it is on tape.

Once it is entered, this subroutine will become part of the other program. Then you can save the combined, new program with a SAVE (on the disk) or CSAVE (on the tape) command.

Variables . . .

W	Whistle counter—10 whistles.
A	Pitch counter and pitch (notes played in the whistle).
T	Delay counter—slows down whistle.
I	Row counter—used in erasing face.

Do-It-Yourself ...

Try this happy face for awhile. When you get tired of it, change it to a new face. All you have to do is type in new graphics characters. Change the eyes. Change the nose. Change the mouth. How about hair? Or a hat? And some ears?

2

SAD FACE

For Parents and Teachers ...

This game subroutine is to be used along with the Happy Face subroutine. It can be attached to most of the games in this book. When children answer the computer's question incorrectly, the subroutine draws a sad face on the TV screen, makes a sad sound, and prints the message "SORRY ... TRY AGAIN."

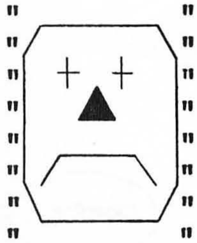
For Kids ...

When you miss a question or get the wrong answer, what do you expect? Probably a sad face. Maybe a sad sound. And, hopefully, encouragement to try again.

Here is a little program that does all that. It prints a sad face on the TV screen. Under the face it prints the message "SORRY ... TRY AGAIN." And it makes a sad sound.

The Program ...

Program Name: **SAD**



```
2000 REM ***SAD FACE & SOUND
2010 POSITION 14,12:PRINT "         "
2020 POSITION 14,13:PRINT "         "
2030 POSITION 14,14:PRINT "         "
2040 POSITION 14,15:PRINT "         "
2050 POSITION 14,16:PRINT "         "
2060 POSITION 14,17:PRINT "         "
2070 POSITION 14,18:PRINT "         "
2080 POSITION 14,19:PRINT "         "
2090 POSITION 9,21:PRINT "SORRY ... TRY
AGAIN"
2110 SOUND 0,100,10,8
2120 FOR PAUSE=1 TO 100:NEXT PAUSE
2130 SOUND 0,240,10,14
2140 FOR PAUSE=1 TO 200:NEXT PAUSE
2150 SOUND 0,0,0,0
2160 FOR PAUSE=1 TO 300:NEXT PAUSE
2165 FOR I=12 TO 19:POSITION 14,I:PRINT
"         ":NEXT I:POSITION 9,21:PRINT "
                    "
2170 RETURN
```

Typing Hints ...

➧ To make the sad face, you need to use graphics characters. The **Control Graphics Keyboard** is pictured in the preface of this book.

In line Number ...	To get ...	Which is (are) ...	Press ...
2010	———	top of face	**CTRL N**
2020, 2060	/ and \	left and right upper corners of face and left and right corners of mouth	**CTRL F** and **CTRL G**
2030, 2040 2050, 2060	❙ and ❙	left and right sides of face	**CTRL V** and **CTRL B**
2030	+	eyes	**CTRL S**
2040	◢ and ◣	left and right sides of nose	**CTRL H** and **CTRL J**
2060, 2080	———	the mouth and bottom of face	**CTRL M**
2070	\ and /	left and right lower corners of face	**CTRL G** and **CTRL F**

Highlights ...

The sad face is similar to the happy face. Lines 2010 to 2080 print out the sad face. Line 2090 prints the sad message. Lines 2110 to 2160 play the sad sound. Line 2165 erases the sad face and the message.

To test the sad face, just type GOSUB 2010.

Note that the sad face starts on line 200. The Sad Face is a subroutine, just like the Happy Face. You can attach it to the tail end of all your other programs.

First, save it on a disk with a LIST command. Or save it on a tape with a LIST "C:" command.

After you have typed another program, copy the Sad Face back into memory with an ENTER command (disk) or ENTER "C:" (tape).

Using the ENTER or ENTER "C:" command you can add the happy **and** sad faces to all your other programs.

First you type in a new program.

Then you ENTER the Happy Face.

Then you ENTER the Sad Face.

Last, you save the combined, new program with a SAVE command (disk) or CSAVE (tape).

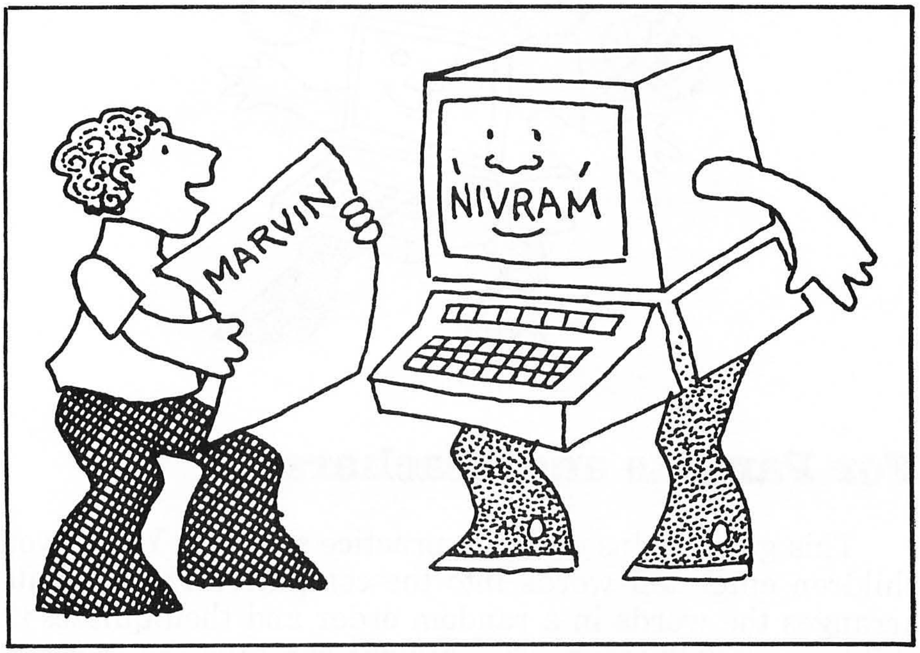
Variables . . .

PAUSE	Delay Counter—Computer pauses between sounds.
I	Row counter—used in erasing face.

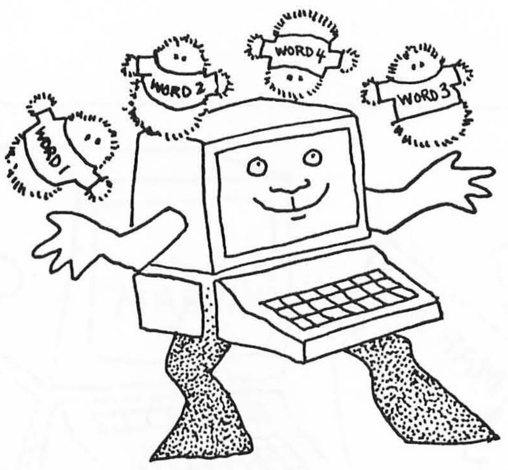
Do-It-Yourself . . .

Try **this** sad face for awhile. When you get tired of it, change it to a new face. All you have to do is type in new graphics characters. Change the eyes. Change the nose. Change the mouth. How about hair? Or a hat? And some ears?

WORDS



3
SCRAMBLED BEES



For Parents and Teachers ...

This game helps children practice spelling. You or your children enter ten words into the computer. The computer arranges the words in a random order and then quizzes the children on each word. The children can take the computer quiz over and over until they have mastered all the words.

For Kids ...

In this chapter we create a "scrambled" spelling bee. The computer juggles the words around. Then it gives you the spelling bee with the words in the new order. And if you ask the computer to quiz you again on the same words, it will scramble the words again. Then it will give you the spelling bee.

The Game ...

Program Name: **SCRAMBEE**

```
50 REM *** SCRAMBLED SPELLING BEE
60 DIM A$(25),B$(25)
62 GRAPHICS 0:POKE 752,1
65 POSITION 8,8:PRINT "***  SPELLING BEE
  ***"
66 FOR PAUSE=1 TO 1000:NEXT PAUSE
70 DIM W1$(25),W2$(25),W3$(25),W4$(25),W
5$(25)
75 DIM W6$(25),W7$(25),W8$(25),W9$(25),W
10$(25)
80 DIM S$(40)
81 S$="975318642090817263543928170645675
8493021"
100 FOR SPELL=1 TO 10
110 PRINT " \ ":POSITION 13,8:PRINT "WORD
 #";SPELL;:INPUT A$
120 ON SPELL GOSUB 3010,3020,3030,3040,3
050,3060,3070,3080,3090,3100
130 NEXT SPELL
135 M=INT(RND(1)*4)
140 FOR SPELL=1 TO 10
150 ON VAL(S$(M*10+SPELL,M*10+SPELL))+1
GOSUB 4010,4020,4030,4040,4050,4060,4070
,4080,4090,4100
160 PRINT " \ ":POSITION 13,8:PRINT "WORD
:  ";A$
170 FOR PAUSE=1 TO 800:NEXT PAUSE
180 PRINT " \ ":POSITION 13,8:PRINT "WORD
";:INPUT B$
190 IF A$=B$ THEN GOSUB 1010:GOTO 210
200 GOSUB 2010:GOTO 160
210 NEXT SPELL
220 PRINT " \ ":POSITION 8,8:PRINT "SAME
WORDS AGAIN";:INPUT A$
230 IF A$(1,1)="Y" THEN 135
240 IF A$(1,1)<>"N" THEN 220
250 POKE 752,0:PRINT " \ ":END
3000 REM *** STORE SPELLING WORDS
3010 W1$=A$:RETURN
```

```
3020 W2$=A$:RETURN
3030 W3$=A$:RETURN
3040 W4$=A$:RETURN
3050 W5$=A$:RETURN
3060 W6$=A$:RETURN
3070 W7$=A$:RETURN
3080 W8$=A$:RETURN
3090 W9$=A$:RETURN
3100 W10$=A$:RETURN
4010 A$=W1$:RETURN
4020 A$=W2$:RETURN
4030 A$=W3$:RETURN
4040 A$=W4$:RETURN
4050 A$=W5$:RETURN
4060 A$=W6$:RETURN
4070 A$=W7$:RETURN
4080 A$=W8$:RETURN
4090 A$=W9$:RETURN
4100 A$=W10$:RETURN
```

Highlights ...

In this program, the computer scrambles the words to be spelled. It has four possible "canned" sequences to use. These sequences are stored in the variable S$ on line 81.

The computer chooses the sequence on line 135.

On line 150, the computer uses the following function to pick each of the 10 spelling-bee words:

```
VAL(S$(M*10+SPELL,M*10+SPELL))+1
```

We interpret this function (as the computer does) from the inside out. First, we know the variable M was chosen on line 135. It is a random number between 0 and 3. Let's say that M equals 3.

Second, let's say that this is the first time through the FOR-NEXT loop (lines 140 to 210). SPELL is the loop counter, so SPELL is equal to 1.

If we plug in the values of SPELL and M, we get the function:

```
VAL(S$(3*10+1,3*10+1))+1
```

This translates to VAL(S$(31,31)+1. We know that S$ (31,31) selects the 31st character in the string S$. The 31st character in S$ is a 6.

If we plug in the 6 we get the function: VAL ("6") +1. The VAL function converts a number in string (nonarithmetic) form into a true number that can be used in arithmetic. The result of VAL ("6") is the number 6.

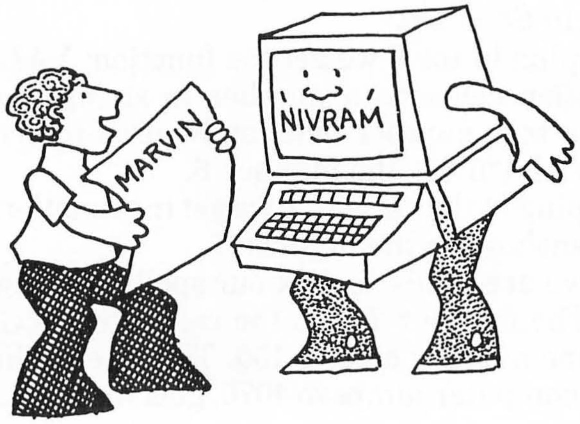
If we plug in the number 6 we get the function: 6+1. This, of course, translates to the number 7.

Now we are ready to pick our spelling-bee word. Look at line 150. The number 7 tells the computer to GOSUB to the **seventh** line number on line 150. The seventh line number is 4070. The computer jumps to 4070, gets the spelling-bee word, and returns.

Voila!

Note that on line 230, we now send the computer back to line 135. Line 135 is where the computer juggles the words and chooses a new scrambled word sequence.

4
BACKWORDS



For Parents and Teachers ...

This game is mostly for fun. But it teaches children to look at words in a playful, original way. It shows them that words aren't fixed and frozen. They can be flipped over and turned around. The result is amusing and often surprising.

For Kids ...

Did you ever wonder what your name looks like spelled backwards? I did. So I tried this computer game.

When the computer typed "FRONTWORDS?" I typed my name—FRED.

A message flashed on the TV screen:

```
BACKWORDS MACHINE NOW WORKING
```

Moments later, the computer printed my name forwards and backwards. Forwards it was FRED. Backwards it was DERF!

What does **your** name look like backwards? Or your mom's name? Or your dad's? Or your favorite hero's name?

Use this game to find out.

The Game ...

Program Name: **BACKWORD**

```
50 REM *** THE BACKWORDS MACHINE
60 POKE 752,1:PRINT "↰"
65 POSITION 4,8:PRINT "***  THE BACKWORD
S MACHINE  ***"
66 FOR PAUSE=1 TO 1000:NEXT PAUSE
70 DIM A$(1)
90 DIM M$(110),M2$(110)
100 PRINT "↰"
105 C=0:M$="":M2$=""
110 PRINT "FRONTWORDS";:INPUT M$
160 PRINT "↰"
170 PRINT "BACKWORDS MACHINE NOW WORKING
"
180 FOR PAUSE=1 TO 1500:NEXT PAUSE
190 FOR I=LEN(M$) TO 1 STEP -1
193 C=C+1
195 IF M$(I,I)=" " THEN M2$(C,C)=" ":GOT
O 210
200 M2$(C,C)=CHR$(ASC(M$(I,I)))
210 NEXT I
220 PRINT "↰"
230 PRINT "FRONTWORDS:"
240 PRINT
250 PRINT M$
260 PRINT :PRINT :PRINT
270 PRINT "BACKWORDS:"
280 PRINT
290 PRINT M2$
300 PRINT :PRINT :PRINT
310 PRINT "PLAY AGAIN";:INPUT A$
```

```
320 IF A$="Y" THEN 100
330 IF A$<>"N" THEN 310
340 POKE 752,0:PRINT "  ":END
```

Typing Hints ...

➽ In line number 170, the print statement "BACKWORDS MACHINE NOW WORKING" is to be typed in **reverse video,** which is shown in the "For Kids" section of this chapter. To enter letters or symbols in reverse video, you need to press the Atari-symbol key (▣), type in the letters of the message itself, and then press (▣) to restore the keyboard to normal video.

Highlights ...

The key to this program is the loop on lines 190 to 210. The name you typed in is stored in M$. The computer takes the last letter of the name in M$ and places a copy of it in the **first** position in M2$. Then it takes the next-to-the-last letter in M$, copies it, and then places it in the **second** position in M2$. The computer continues this process until it has reached the very first letter in M$. It puts this at the **end** of the letters it has stored in M2$. In this way, the computer turns the entire word around. What went frontwards into M$ comes out backwards into M2$.

Do-It-Yourself ...

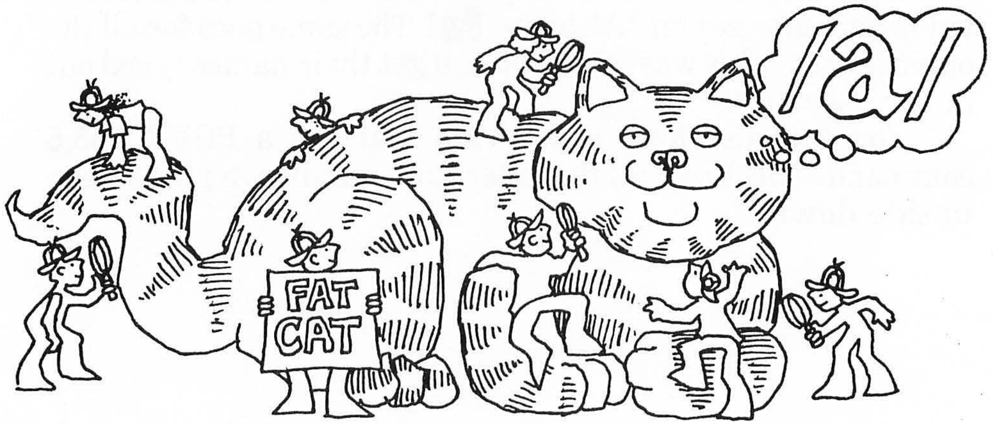
You might modify this program so that you can store the "backwords" on tape or disk.

Also, "backwords" are just an example of the limitless ways you can play around with words on your computer. Electronic words aren't fixed or frozen. They are plastic or clay—infinitely malleable.

One way to get your computer to play with words is to translate all the words from regular video into **reverse video.** For example, the upper-case letters in the alphabet (A to Z) are represented by Atari ASCII values of 65 to 90. The upper-case letters in reverse video are represented by 193 to 218. You can change the "Backwords" program so that every time you type in a letter A, for example, the program adds 128 to the ASCII value and changes an "A" to an "A." The same goes for all the other letters. This way, children can get their names typed out in reverse video.

You can also have a program that has a POKE 755,6 command. This turns all the letters and numbers on the screen **upside down!**

5
FAT CAT



For Parents and Teachers ...

This game helps children learn phonetic symbols and phonetic sounds. The computer displays the phonetic symbols on the TV screen. Then it flashes a word on the screen. The child says the word aloud and tries to guess which phonetic sound is hidden inside the word. An arrow points to the first phonetic symbol. To move the arrow to a new symbol, the child presses the **SPACE** bar. To select a symbol the child presses the **RETURN** button.

For Kids ...

Did you ever notice how certain words rhyme or sound alike? Words like **red** and **bed**, or **big** and **pig**, or **fat** and **cat**?

The reason the words rhyme is that they have the same phonetic sound (or **phoneme**) hidden inside them. For example, hiding inside the words red and bed is the phoneme /**e**/. Hidden inside the words big and pig is the phoneme /**i**/. And inside the words fat and cat is the phoneme /**a**/.

In this game you get to play detective. The computer will flash a word on your TV screen. Say the word out loud a couple of times. Now look at the list of phonemes on the screen. Your job as a sound detective is to discover which phoneme is hidden inside the word.

The Game ...

Program Name: **FATCAT**

```
50 REM *** FATCAT GAME
60 PRINT " \ ":POKE 752,1
65 POSITION 6,8:PRINT "***  THE FAT CAT
GAME  ***"
66 FOR PAUSE=1 TO 1000:NEXT PAUSE
70 DIM A$(1)
80 DIM W$(6)
100 FOR M=1 TO 5
102 TRY=0
105 GRAPHICS 2:POKE 756,226:SETCOLOR 0,0
,0:POKE 752,1
110 GOSUB 3010:REM * SET UP SCREEN
115 IF TRY=0 THEN GOSUB 4010:REM * SELEC
T WORD
120 GOSUB 4110:REM * PRINT WORD
125 GOSUB 3510:REM * MOVE ARROW
130 IF PH/10=C THEN GRAPHICS 0:POKE 752,
1:GOSUB 1010:GOTO 150
140 GRAPHICS 0:POKE 752,1:GOSUB 2010:TRY
=1:GOTO 105
150 NEXT M
160 GRAPHICS 0:POKE 752,1:POSITION 15,8:
PRINT "PLAY AGAIN";:INPUT A$
170 IF A$="Y" THEN 100
180 IF A$<>"N" THEN 160
190 GRAPHICS 0:POKE 752,0:END
3000 REM *** SET UP PHONETIC SOUNDS
```

```
3010 POSITION 1,0:PRINT #6;"/a/"
3020 POSITION 1,2:PRINT #6;"/ch/"
3030 POSITION 1,4:PRINT #6;"/e/"
3040 POSITION 1,6:PRINT #6;"/f/"
3050 POSITION 1,8:PRINT #6;"/l/"
3060 POSITION 6,0:PRINT #6;"/sh/"
3070 POSITION 6,2:PRINT #6;"/s/"
3080 POSITION 6,4:PRINT #6;"/th/"
3090 POSITION 6,5:PRINT #6;" _"
3100 POSITION 6,6:PRINT #6;"/+/"
3110 POKE 656,0:POKE 657,6:PRINT "RETURN
 = YES      SPACE = NO"
3120 RETURN
3500 REM *** ARROW SUBROUTINE
3510 OPEN #1,4,0,"K:"
3520 OLDX=-1:OLDY=-1:C=0
3550 FOR X=0 TO 5 STEP 5:FOR Y=0 TO 8 ST
EP 2
3552 C=C+1
3555 IF OLDX<>-1 THEN GOSUB 3655
3560 POSITION X,Y:PRINT #6;CHR$(27);:POK
E 85,X:PRINT #6;CHR$(159)
3570 GET #1,K
3580 IF K=155 THEN POP :CLOSE #1:GOTO 36
30
3590 IF K=32 THEN 3610
3600 GOTO 3570
3610 OLDY=Y:OLDX=X
3620 IF X=5 AND Y=6 THEN GOSUB 3655:POP
:GOTO 3520
3625 NEXT Y:NEXT X
3630 RETURN
3650 REM *** ERASE ARROW
3655 POSITION OLDX,OLDY:PRINT #6;CHR$(32
)
3670 RETURN
4000 REM *** SELECT SOUND/WORD
4010 PH=(INT(RND(1)*9)+1)*10
4020 RESTORE 5000+PH
4030 WD=INT(RND(1)*5)+1
4040 FOR L=1 TO WD
4050 READ W$
4060 NEXT L
4090 RETURN
```

```
       4100 REM *** PRINT WORD
       4110 POSITION 12,4:PRINT #6;"         "
       4120 POSITION 12,4:PRINT #6;W$
       4130 RETURN
       5010 DATA fat,cat,man,hat,fast
       5020 DATA rich,witch,lunch,catch,much
       5030 DATA red,hen,bed,leg,pet
       5040 DATA stuff,off,cuff,muff,huff
    »  5050 DATA bell,still,will,roll,sell
       5060 DATA ship,fish,splash,wish,dish
       5070 DATA grass,glass,cross,dress,mess
       5080 DATA thin,bath,cloth,with,thick
       5090 DATA that,this,then,than,them
```

Typing Hints ...

» To make the slanted lines (/)on either side of the phonetic symbol, hold down the **CTRL** button and type **F**.

» To make the special phonetic symbol (Ŧ), you need to use two graphics characters. On line 3090, hold down the **CTRL** button and type **N**. On line 3100, type the two slanted lines. Between the lines, hold down the **CTRL** button and type **S**.

» To make the message on line 3110 in reverse video, you need to push the Atari-symbol key (▲), type the letters of the message itself, and then press the Atari-symbol key again. This gets you back to normal video.

» To make the lower-case letters on lines 3010 to 3080 and lines 5010 to 5090, press the **CAPS LOWR** key. To get back to the upper-case mode, hold down the **SHIFT** key and press the **CAPS LOWR** key again.

Highlights ...

This game uses four major subroutines. The first subroutine (called on line 110) prints the list of phonetic symbols. The second subroutine (called on line 115) selects the word with the mystery phoneme hidden inside. The third subroutine (called on line 120) prints the word on the TV screen. The fourth subroutine (called on line 125) moves the "choice" arrow each time you press the **SPACE** bar.

This game is played in Graphics Mode 2—with a difference. On line 105 the command POKE 756,226 changes the Mode 2 character set from numbers and upper-case letters to graphics symbols and lower-case letters.

Try this in a small test program. What do you see?

You see **hearts**—Valentine's Day hearts everywhere. The heart shape—Atari ASCII code 32—is equivalent to the space character in alternate Graphics Mode 2.

Hearts are okay, but in this game we don't need them. To eliminate them we use the SETCOLOR 0,0,0 command (on line 105). Now all the hearts are changed into invisible spaces.

Variables . . .

PAUSE	Delay loop counter.
A$	For child's yes-no answer to "Play Again?"
W$	Word with mystery phoneme.
M	Loop counter—controls number of words/game.
TRY	If TRY=1, child has given an incorrect answer.
PH	Pointer to list of words containing a certain phoneme.
WD	Pointer to one word in the list.
C	Pointer to phoneme symbol (child's answer).
X	Current column position of arrow.
Y	Current row position of arrow.
OLDX	Old column position of arrow.
OLDY	Old row position of arrow.
L	Loop counter for randomly picking word from list.

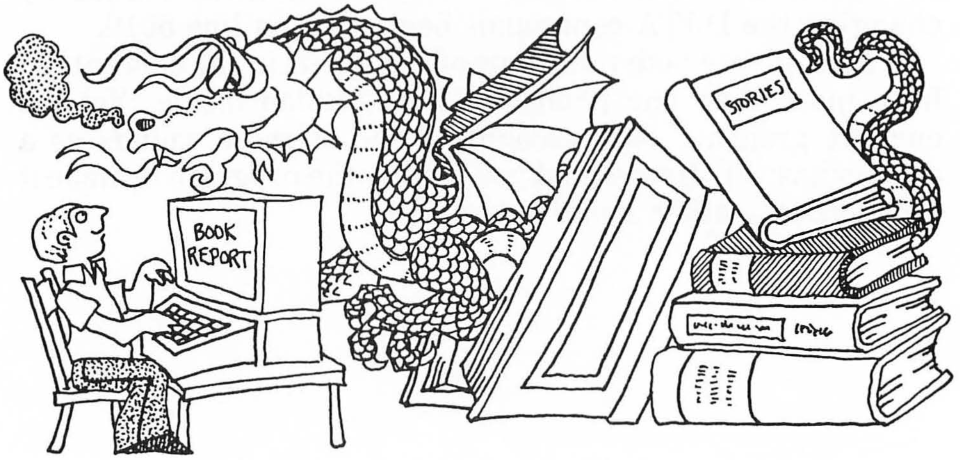
Do-It-Yourself ...

This is only a small sample of the different phonetic symbols that represent word sounds. You can add new sounds by changing the list of sounds in the subroutine beginning on line 3000. You can add new words with these sounds by changing the DATA commands beginning on line 5010.

Also, please note that some of the words in the current list have more than one phonetic sound hidden inside. Yet the current program only recognizes one of those sounds as a correct answer. How would you modify the program to make it recognize the other sound or sounds?

6

BOOK REPORT



For Parents and Teachers ...

This game will help children with their writing assignments, especially book reports. The program asks the child for pertinent information—the date, the child's name, the author, and the title of the book. Then the computer asks the child "WHAT HAPPENED FIRST?" When the child enters this information, the computer asks "WHAT HAPPENED NEXT?"

The child responds to the computer's questions and enters the report, sentence by sentence. When finished, the child types END or THE END. Then the computer prints the entire book report, all nicely formatted, on the TV screen.

For Kids ...

Did you ever have to write a book report or tell a story, and you didn't know where to start?

Here's a sure-fire way to get you going. Pretend that the computer is a person. And imagine that the person can hardly wait to hear about a neat book you just read or a story you made up inside your head.

What does the person ask you to get you started? He says, "What happened first?" So you tell him.

It sounds exciting, so he asks, "What happened next?" So you tell him that, too.

Each time you tell him a little bit of the story, he wants to hear more. So he keeps asking "What happened next?"

Before you know it, you've finished describing the book you read or the story you made up. When he asks you "What happened next?" you simply say "END" or "THE END."

Then, sit back. The computer hasn't forgotten a word you said. It is going to replay the entire story or book report right before your eyes.

As it types the story on the TV screen, you can be busy copying it in your notebook. Then it's ready to turn in to your teacher at school tomorrow morning.

The Game . . .

Program Name: **BKREPORT**

```
50 REM *** THE BOOK REPORT
60 PRINT " \ ":POKE 752,1
65 POSITION 7,8:PRINT "***  THE BOOK REP
ORT  ***"
67 FOR PAUSE=1 TO 1000:NEXT PAUSE
70 DIM NM$(15),T$(30),A$(25),R$(1000)
80 DIM L$(80),DT$(15)
90 DIM B$(37)
92 B$="
   "
94 DIM POS(20)
95 P=1
100 PRINT " \ "
110 PRINT "WHO'S OUT THERE";:INPUT NM$
120 PRINT " \ "
125 PRINT "TODAY'S DATE";:INPUT DT$
```

```
127 PRINT "↰"
130 PRINT "BOOK TITLE";:INPUT T$
140 PRINT "↰"
150 PRINT "AUTHOR";:INPUT A$
160 PRINT "↰"
170 PRINT "WHAT HAPPENED FIRST":INPUT L$

180 R$="  "
190 R$(3)=L$
200 PRINT "↰"
210 PRINT "WHAT HAPPENED NEXT":INPUT L$
215 IF L$="END" OR L$="THE END" THEN 300

217 IF L$="" THEN R$(LEN(R$)+1)="@  ":PO
S(P)=LEN(R$):P=P+1
220 IF R$(LEN(R$))="." THEN R$(LEN(R$)+1
)=" "
225 R$(LEN(R$)+1)=" "
230 R$(LEN(R$)+1)=L$
240 GOTO 200
300 PRINT "↰"
303 L$="BOOK REPORT BY "
304 L$(LEN(L$)+1)=NM$
305 GOSUB 3010:REM * CENTER LINE
306 POSITION X,0:PRINT L$
308 L$=DT$:GOSUB 3010:POSITION X,2:PRINT
 L$
310 L$=T$:GOSUB 3010:POSITION X,4:PRINT
L$
320 GOSUB 3510:REM *** TITLE LINE
330 POSITION X,5:PRINT L$
340 L$="BY ":L$(4)=A$:GOSUB 3010:POSITIO
N X,7:PRINT L$
345 PRINT :PRINT
350 GOSUB 4010:REM *** PRINT REPORT
355 IF PEEK(84)>=23 THEN GOSUB 4510
360 L$="THE END":GOSUB 3010:POSITION X,P
EEK(84)+2:PRINT L$
380 POKE 752,0
390 GOTO 390
3000 REM *** CENTER LINE SUBROUTINE
3010 X=((38-LEN(L$))/2)+1
3020 RETURN
3500 REM *** TITLE LINE SUBROUTINE
3510 FOR I=1 TO LEN(T$)
```

```
3520 L$(I,I)="-"
3530 NEXT I
3540 RETURN
4000 REM *** PRINT REPORT
4010 D=0:P=1
4015 IF LEN(R$(D+1))<=37 THEN PRINT R$(D
+1):RETURN
4016 RS=37
4020 IF R$(D+RS,D+RS)=" " THEN 4090
4030 FOR RS=36 TO 1 STEP -1
4040 IF R$(D+RS,D+RS)=" " THEN POP :GOTO
 4090
4050 NEXT RS
4090 IF PEEK(84)>=23 THEN GOSUB 4510
4095 IF POS(P)>D+RS OR POS(P)=0 THEN 410
0
4096 GOSUB 5010:GOTO 4015
4100 PRINT R$(D+1,D+RS)
4110 D=D+RS
4130 GOTO 4015
4500 REM *** PAUSE/CLEAR SCREEN
4510 FOR PAUSE=1 TO 3000:NEXT PAUSE
4520 FOR L=10 TO 22
4530 POSITION 2,L:PRINT B$
4540 NEXT L
4550 POKE 84,10
4560 RETURN
5000 REM *** PARAGRAPH
5010 RS=POS(P)-D-3
5020 P=P+1
5030 PRINT R$(D+1,D+RS)
5040 D=D+RS+3
5050 IF PEEK(84)>=21 THEN GOSUB 4510:GOT
O 5080
5060 POKE 84,PEEK(84)+1
5080 RETURN
```

Highlights . . .

The computer copies the entire story or book report into a giant character string called R$.

When you are done entering a book report or story into the computer and you type THE END, (or, simply, END), the computer prints out a title, the date, your name, and the entire text.

The subroutine beginning on line 3000 centers all the headings, including the book title, the author, the date, and your name.

The subroutine beginning on line 3500 underlines the book title. The subroutine beginning on line 4000 prints the text of the report or story.

If a word is too long to fit at the end of a line, the computer doesn't break the word in two. Instead, it lifts the entire word and places it on the next line.

When your child is writing the report or story and comes to the end of a paragraph, he or she should answer the computer's question ("WHAT HAPPENED NEXT?") by pressing the **RETURN** button (see line 217).

When you press the **RETURN** button without entering any other information, the computer receives a **null** character and inserts an @ sign. When the computer prints out the report it knows where the @ signs are because it stored their location in the string R$ inside the array POS. When it comes to an @ sign, it ends an old paragraph and creates a new, indented paragraph automatically.

If the book report or story is too big to fit on the screen, the computer pauses at the bottom of the screen for you to read what is there. Then it erases only the text on the screen (leaving the title information and your name intact), and fills the screen with new text. It keeps doing this until it comes to the end of the story or report. At the end, it prints the words "THE END," centered on the screen.

Did you notice how many PEEKs and POKEs there are to memory position 84? The computer goes to location 84 to find where it is on the TV screen as it prints out lines of text. Since it is in Graphics Mode 0, the computer has only 24 lines (from 0 to 23). The book report uses location 84 to keep the report or story from spilling onto a new screen.

Variables ...

PAUSE	Delay-loop counter.
NM$	Child's name.
T$	Book title.
A$	Author's name.
R$	The text of the story or report.
L$	A single text line.
DT$	The current date.
B$	A blank line (to erase a single screen line).
POS	Array—location of the @ signs in R$ (@ signs are paragraph symbols; R$ is the main text of the book report or story).
P	Points to the next unoccupied spot in POS.
X	The column position of centered text.
I	Loop counter for centering title line.
D	Starting position of current line in text string R$.
RS	Ending position of current line in text string R$.
L	Loop counter in loop that erases old text from screen.

Do-It-Yourself ...

You have the rough beginnings of a word processor here. It lets you enter reports, letters, stories, or whatever you choose in an orderly, neat-looking format.

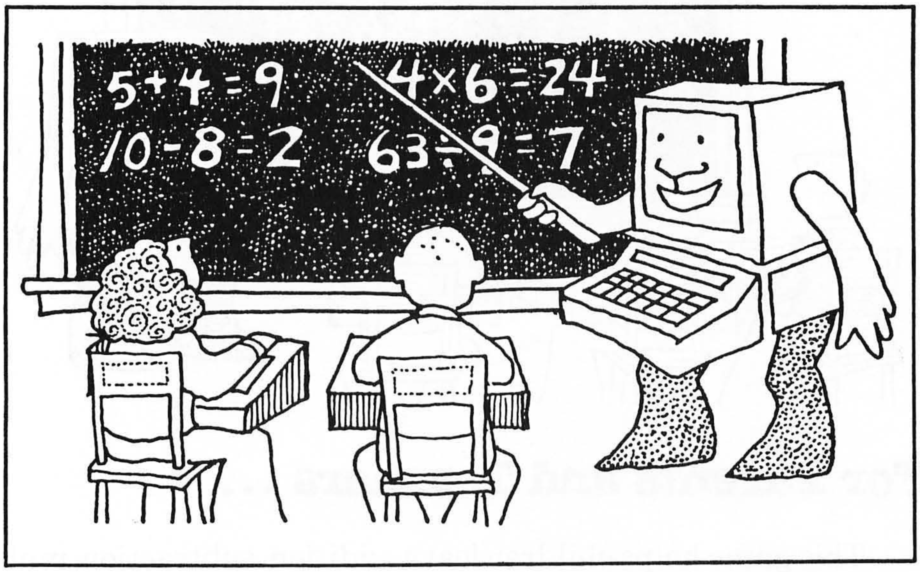
But there's a lot you can add. For example, when the text appears on the screen, the computer pauses (on line 4510) for a count of from 1 to 3000. Why not change that and make the computer display a certain screenful of text until you press the **RETURN** button. That way, you can relate how fast the computer jumps to a new screen to the speed of the reader. This would be especially helpful to younger children who still read very slowly.

And how about a subroutine that takes the report and stores it on a tape or disk? Your child might spend a long time typing in a report and might want to save sections of it at one time on tape or disk to work on later.

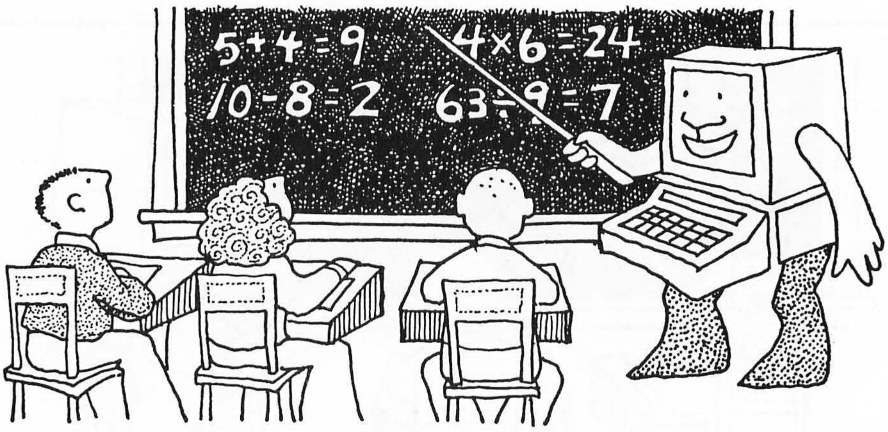
And how about printing the report? Do you have a printer? If you do, why make your child copy the whole report? Instead, add a subroutine that lets the child review the report on the TV screen and then type it out automatically on a printer.

There are a lot of things you can do with this program. With a few additions, it can turn your computer into the family's electronic typewriter.

NUMBERS



7
THE ARITHMETIC GAME



For Parents and Teachers ...

This game helps children learn addition, subtraction, multiplication, and division.

For Kids ...

When I was a child, I used to work on an arithmetic problem and then get frustrated because I didn't know if my answer was right or wrong. I wanted to know the answer right away. But I had to wait until the teacher graded my homework paper or my test. Sometimes that took days or even weeks. By the time I got my paper back, the problem was no longer fresh in my mind. If I got it right, I didn't know why. If I got it wrong, I didn't know why. And I no longer really cared.

Now you can solve arithmetic problems and learn instantly if you got them right or wrong. How? You can play the

Arithmetic Game. It's like hiring the computer as your own private arithmetic teacher.

When you RUN The Arithmetic Game, the computer displays a "menu" on the TV screen. If you press a **1**, the computer challenges you with five addition problems. If you press a **2**, it throws five subtraction problems at you. If you press a **3**, you get five multiplication problems. If you press a **4**, you get five division problems.

The computer displays the problem on the screen. Then you try to think up the answer. If you want to, get out a piece of scrap paper and a pencil. That's what I do when I'm working on arithmetic problems

Think up your answer to the problem, and then type it in. If you want to change your answer, press the **DELETE/BACK S** key. When your answer is on the screen, press the **RETURN** button.

If you get the answer right, the computer draws a happy face, says "RIGHT!," and plays a happy tune.

If you get the answer wrong, the computer draws a sad face, groans, and encourages you to "TRY AGAIN!"

The Game ...

Program Name: **MATH**

```
50 REM *** THE ARITHMETIC GAME
60 PRINT "\":POKE 752,1
65 POSITION 4,8:PRINT "***  THE ARITHMET
IC GAME  ***"
66 FOR PAUSE=1 TO 1000:NEXT PAUSE
70 DIM B$(37)
72 B$="
  "
80 DIM A$(1)
90 DIM N$(5)
100 GOSUB 3010:REM * MENU
165 OPEN #1,4,0,"K:"
170 FOR L=1 TO 5
180 N1=INT(RND(1)*10)+1
190 N2=INT(RND(1)*10)+1 N3=N2
```

```
200 GOSUB 3510:REM * GAME
250 NEXT L
255 CLOSE #1
260 GRAPHICS 0:POKE 752,1
265 POSITION 13,8:PRINT "PLAY AGAIN";:IN
PUT A$
270 IF A$="Y" THEN 100
280 IF A$<>"N" THEN 260
300 GRAPHICS 0:POKE 752,0:END
3000 REM *** MENU SUBROUTINE
3010 GRAPHICS 0:POKE 752,1
3015 TRAP 3010
3020 POSITION 11,6:PRINT "1.  ADDING"
3030 POSITION 11,8:PRINT "2.  SUBTRACTIN
G"
3040 POSITION 11,10:PRINT "3.  MULTIPLYI
NG"
3050 POSITION 11,12:PRINT "4.  DIVIDING"

3060 POSITION 4,18:PRINT "WHICH NUMBER (
1, 2, 3, OR 4)";:INPUT B
3070 IF B>=1 AND B<=4 THEN 3090
3080 GOTO 3010
3090 RETURN
3500 REM *** ARITHMETIC SUBROUTINE
3510 GRAPHICS 2:POKE 752,1
3512 TRAP 3510
3513 N$=""
3515 ON B GOSUB 6010,6110,6210,6310
3517 X=PEEK(85)-1:P=X
3520 GET #1,K
3522 IF (K<>126) OR (K=126 AND X=P) THEN
 3530
3524 X=X-1:POSITION X,4:PRINT #6;" ";
3525 IF LEN(N$)=1 THEN N$="":GOTO 3520
3526 N$=N$(1,LEN(N$)-1)
3528 GOTO 3520
3530 IF K=155 THEN 3600
3540 IF K<48 OR K>57 THEN 3520
3550 N$(LEN(N$)+1)=CHR$(K)
3570 POSITION X,4:PRINT #6;CHR$(K);
3575 X=X+1
3580 GOTO 3520
3600 ON B GOSUB 7010,7110,7210,7310
```

```
3605 IF W=VAL(N$) THEN GRAPHICS 0:POKE 7
52,1:GOSUB 1010:GRAPHICS 2:POKE 752,1:GO
TO 3650
3610 GRAPHICS 0:POKE 752,1:GOSUB 2010:GO
TO 3510
3650 RETURN
6000 REM *** ADDING SUBROUTINE
6010 POKE 657,15:PRINT "ADDING"
6020 POSITION 5,4:PRINT #6;N1;" + ";N2;"
 = ?";
6030 RETURN
6100 REM *** SUBTRACTING SUBROUTINE
6110 POKE 657,14:PRINT "SUBTRACTING"
6120 IF N2>N1 THEN W=N2:N2=N1:N1=W
6130 POSITION 5,4:PRINT #6;N1;" - ";N2;"
 = ?";
6140 RETURN
6200 REM *** MULTIPLYING SUBROUTINE
6210 POKE 657,14:PRINT "MULTIPLYING"
6220 POSITION 5,4:PRINT #6;N1;" X ";N2;"
 = ?";
6230 RETURN
6300 REM *** DIVIDING SUBROUTINE
6310 POKE 657,16:PRINT "DIVIDING"
6320 W=N1*N3:N2=W
6330 POSITION 5,4:PRINT #6;N2;" / ";N1;"
 = ?";
6340 RETURN
7000 REM *** ADDING SUBROUTINE
7010 W=N1+N2
7020 RETURN
7100 REM *** SUBTRACTING SUBROUTINE
7110 W=N1-N2
7120 RETURN
7200 REM *** MULTIPLYING SUBROUTINE
7210 W=N1*N2
7220 RETURN
7300 REM *** DIVIDING SUBROUTINE
7310 W=N2/N1
7320 RETURN
```

Highlights ...

The game begins (on line 100) by calling the "menu" subroutine beginning on line 3000. The subroutine lets you choose addition, subtraction, multiplication, or division.

The program uses a FOR-NEXT loop on lines 170 to 250 to control the number of the problems the computer "thinks up." Right now, the computer comes up with five problems per game. Then it asks if you want to "PLAY AGAIN?" If you type "Y" or "YES," the computer calls the menu subroutine. You can continue solving the same kind of problems (addition, subtraction, multiplication, or division), or you can try something new.

The RND functions on lines 180 and 190 are set to randomly choose two numbers between 1 and 10. These are the numbers added together, subtracted, multiplied, or divided.

The actual arithmetic game is a subroutine beginning on line 3500. It is called five times by a GOSUB command on line 200.

The arithmetic subroutine calls two special purpose subroutines, depending on which kind of arithmetic you have chosen (addition, subtraction, multiplication, or division).

The first subroutine (beginning on lines 6000, 6100, 6200, or 6300) displays the arithmetic problems on the screen in Graphics Mode 2. The second subroutine (beginning on lines 7000, 7100, 7200, or 7300) calculates the correct answer to the problem so that it can be matched with your answer.

The major part of the arithmetic subroutine (lines 3517 to 3575) controls the Graphics 2 screen, allowing you to enter your answer to the arithmetic problem. If you enter a number and then change your mind, you can press the **DELETE/BACK S** key, to erase the number and start again.

A TRAP command on line 3512 will take over if you enter too many digits in your answer. You can enter up to five digits (a number between 10,000 and 99,999) safely. If you try to enter six digits, the TRAP command will erase your answer and let you try again.

Similarly, a TRAP command back on line 3015 will take over if you incorrectly press a non-number key to answer the computer's question on line 3060: "WHICH NUMBER (1, 2, 3, or 4)?" The TRAP command makes the computer jump back to line 3010 and print the menu and the question again.

Variables ...

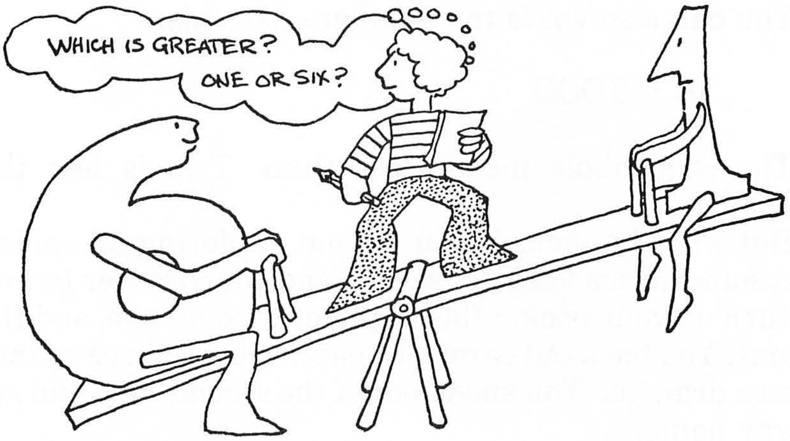
PAUSE	Delay-loop counter.
B$	Prints empty spaces—for erasing the screen.
A$	Your answer to the question "PLAY AGAIN?"
N$	Stores your answer to the arithmetic problem. (N$ is built from 1-byte key entries originally stored in K by the command GET #1,K. N$ lets you enter numbers having as many as five digits.)
L	Loop counter—main loop.
N1	First random number for arithmetic problem.
N2	Second random number for arithmetic problem.
B	Which type of arithmetic problems you choose (addition, subtraction, multiplication, or division).
X	Column on the TV screen where your answer will appear.
K	One digit of your answer.
P	Column on the TV screen where the first digit of your answer appears—the same column where the question mark appears when the problem is first displayed—the program prevents you from back spacing (to the left) past this column.
W	The computer's answer to the arithmetic problem.

Do-It-Yourself . . .

The program does not keep track of how many incorrect answers a child enters. It would be good to create an incorrect answer COUNT variable. Increment the variable (COUNT = COUNT +1) each time the child makes a wrong answer. When COUNT = 3, have the computer display the correct answer on the TV screen. Then have the program ask the child the same problem again.

8

GREATER THAN WHAT?



For Parents and Teachers ...

This game helps children learn the size relationships between numbers. It shows them two numbers, and then asks them if the first number is greater than the second; or less than the second; or greater than or equal to the second; or less than or equal.

The game helps children learn the number comparison symbols: > (greater than), < (less than), >= (greater than or equal), and <= (less than or equal).

For Kids ...

Numbers describe things—apples, dragons, mudpies, and worms. The bigger the number, the more things. For example, if a blue room has 1000 worms in it and a red room has 2 worms in it, which room has more worms in it? The blue room, right?

That's because a 1000 is **greater than** 2. You can compare the numbers by drawing a thousand worms and then two more worms. Or you can take a shortcut and write:

1000 > 2

The > symbols means **greater than.** A thousand is greater than two.

You can also write the numbers like this:

2 < 1000

The < symbols means **less than.** Two is less than a thousand.

But what happens if you are out exploring an enchanted forest some afternoon after school, and you discover two caves. You turn on your pocket flashlight, enter one cave, and find 14 dragons. You back out carefully, enter the second cave and find 14 more dragons. You sneak out of the second cave and run all the way home.

You call the police and the fire department. Moments later, they are at your front door. They ask you what you saw.

To describe what you just saw, you take out a piece of paper and draw 14 dragons and then 14 more dragons.

You look at the cave pictures with all the dragons. Which cave had more dragons? Neither. They both had the same number of dragons. You can draw a big = (equal sign) between the two pictures of dragons. Or you can take a shortcut and write:

14 = 14

But what if you change your mind. In the first cave (the dirty cave), you think you saw shadows of other dragons. That means the dirty cave might have more dragons than the clean cave. You know that the clean cave had only 14 dragons. But some extra dragons might have been hiding in the dirty cave, behind the dragon garbage and the piles of old bones and dirty treasure you saw.

Now you can't say how many dragons were in the dirty cave. You use the symbol X to represent this mystery number. Is there anything you can say about X?

Yes. You know X is at least 14 and maybe more. You can write this as:

X >= 14

This means that X is **greater than or equal to** 14. You can also write this as:

14 <= X

This means 14 is **less than or equal to** X (the number of dragons in the dirty cave).

The police and fire fighters are satisfied with your description. They fly in a dragon S.W.A.T. team from China. The team comes equipped with lots of silk dragon nets. You lead the team back to the caves. You hope that there aren't too many more dragons in the dirty cave.

After you return from dragon hunting, sit down at the computer and play the Greater Than What? game. The computer flashes two numbers on the screen. You have to decide if one number is greater than the other; or less than the other; or greater than or equal to the other; or less than or equal to the other.

Think of the numbers as dragons.

In this game the computer expects a yes or no answer. You give a "YES" answer by pressing the **RETURN** button. You give a "NO" answer by pressing the **SPACE bar.**

The Game ...

Program Name: **GREATER**

```
50 REM *** GREATER-THAN GAME
60 PRINT " ↰ ":POKE 752,1
65 POSITION 4,8:PRINT "***  THE GREATER-
```

```
THAN GAME  ***"
66 FOR PAUSE=1 TO 1000:NEXT PAUSE
80 DIM A$(1)
100 GOSUB 3010:REM * MENU
165 OPEN #1,4,0,"K:"
170 FOR L=1 TO 5
180 N1=INT(RND(1)*10)+1
190 N2=INT(RND(1)*10)+1
200 ON B GOSUB 3510,4010,4510,5010
250 NEXT L
255 CLOSE #1
260 GRAPHICS 0:POKE 752,1
265 POSITION 13,8:PRINT "PLAY AGAIN";:IN
PUT A$
270 IF A$="Y" THEN 100
280 IF A$<>"N" THEN 260
300 CLOSE #1:GRAPHICS 0:POKE 752,0:END
3000 REM *** MENU SUBROUTINE
3010 GRAPHICS 0:POKE 752,1
3015 TRAP 3010
3020 POSITION 3,6:PRINT "1.  GREATER THA
N ...........  >"
3030 POSITION 3,8:PRINT "2.  LESS THAN .
..............  <"
3040 POSITION 3,10:PRINT "3.  GREATER TH
AN OR EQUAL ...  >="
3050 POSITION 3,12:PRINT "4.  LESS THAN
OR EQUAL ......  <="
3060 POSITION 3,18:PRINT "WHICH NUMBER (
1, 2, 3, OR 4)";:INPUT B
3070 IF B>=1 AND B<=4 THEN 3090
3080 GOTO 3010
3090 RETURN
3200 REM *** INITIALIZE SCREEN
3210 GRAPHICS 2:POKE 752,1
3220 PRINT "      RETURN = YES     SPACE =
 NO"
3230 RETURN
3500 REM *** GREATER-THAN (>)
3510 GOSUB 3210
3515 POSITION 6,4:PRINT #6;N1;" > ";N2;"
 ?"
3520 GET #1,K
3530 IF (N1>N2 AND K=155) OR (N1<=N2 AND
 K=32) THEN GRAPHICS 0:POKE 752,1:GOSUB
```

```
1010:GOTO 3570
3540 IF (N1>N2 AND K=32) OR (N1<=N2 AND
K=155) THEN GRAPHICS 0:POKE 752,1:GOSUB
2010:GOTO 3510
3550 GOTO 3520
3570 RETURN
4000 REM *** LESS-THAN (<)
4010 GOSUB 3210
4015 POSITION 6,4:PRINT #6;N1;" < ";N2;"
 ?"
4020 GET #1,K
4030 IF (N1<N2 AND K=155) OR (N1>=N2 AND
 K=32) THEN GRAPHICS 0:POKE 752,1:GOSUB
1010:GOTO 4070
4040 IF (N1<N2 AND K=32) OR (N1>=N2 AND
K=155) THEN GRAPHICS 0:POKE 752,1:GOSUB
2010:GOTO 4010
4050 GOTO 4020
4070 RETURN
4500 REM *** GREATER THAN/EQUAL (>=)
4510 GOSUB 3210
4515 POSITION 6,4:PRINT #6;N1;" >= ";N2;
" ?"
4520 GET #1,K
4530 IF (N1>=N2 AND K=155) OR (N1<N2 AND
 K=32) THEN GRAPHICS 0:POKE 752,1:GOSUB
1010:GOTO 4570
4540 IF (N1>=N2 AND K=32) OR (N1<N2 AND
K=155) THEN GRAPHICS 0:POKE 752,1:GOSUB
2010:GOTO 4510
4550 GOTO 4520
4570 RETURN
5000 REM *** LESS THAN/EQUAL (<=)
5010 GOSUB 3210
5015 POSITION 6,4:PRINT #6;N1;" <= ";N2;
" ?"
5020 GET #1,K
5030 IF (N1<=N2 AND K=155) OR (N1>N2 AND
 K=32) THEN GRAPHICS 0:POKE 752,1:GOSUB
1010:GOTO 5070
5040 IF (N1<=N2 AND K=32) OR (N1>N2 AND
K=155) THEN GRAPHICS 0:POKE 752,1:GOSUB
2010:GOTO 5010
5050 GOTO 5020
5070 RETURN
```

Typing Hints ...

➧ To get the words on line 3220 to appear in reverse video, press the Atari-symbol button (⏏). To turn off the reverse video, press the button again.

Highlights ...

This game is similar to The Arithmetic Game. When the program begins, it calls the menu subroutine beginning on line 3000. You can choose four different kinds of number comparison problems: greater than (>), less than (<), greater than or equal (>=), or less than or equal (<=).

The FOR-NEXT loop on lines 170 to 250 is the main program loop. It sets the number of problems per game. Right now the loop is set to offer you five problems per game.

The two RND functions on lines 180 and 190 randomly select the numbers to be compared. The ON B GOSUB command on line 200 calls the appropriate subroutine, based on your choice of problem type (greater than, less than, greater than or equal, less than or equal).

The four subroutines beginning at lines 3500, 4000, 4500, and 5000 handle the four types of problems. They display the problem on the TV screen in Graphics Mode 2 (see the subroutine beginning on line 3200). They accept your answer, and check to see if it is correct. If it is correct, they call the Happy Face subroutine. If it is incorrect, they call the Sad Face subroutine.

You give a yes answer by pressing the **RETURN** button (an Atari ASCII value of 155). You give a no answer by pressing the SPACE bar (an Atari ASCII value of 32).

Variables ...

PAUSE	Delay-loop counter.
A$	Answer to the question “TRY AGAIN?”

L	Loop counter—main program loop.
N1	First number the computer randomly selects.
N2	Second number the computer randomly selects.
B	Your choice of problem type (>, <, >=, or <=).
K	Your answer (**RETURN** = Yes, **SPACE** bar = No).

Do-It-Yourself . . .

You can add a new problem type—not equal (<>).

You can combine the Greater Than What? game and The Arithmetic Game. First the children would have to do an arithmetic problem. Then they would have to use one of the five number comparison sumbols (>, <, >=, <=, or <>).

Or you can combine the two games by asking questions like:

```
5 + 9 > 14 + 0 ?
```

Or you could help older children practice algebra with problems like:

```
IF X=100 THEN IS X/5 >= .1X + 9 ?
```

9 THE HAMBURGER CONTEST

For Parents and Teachers ...

This game helps children practice multiplication.

The computer draws from one to nine hamburgers and from one to nine people. It asks, for example, if four people eat three hamburgers each, how many hamburgers are eaten?

For Kids ...

Do you like hamburgers?

I hope so, because you've been elected by your school to go to the first national KIDS' HAMBURGER EATING CONTEST.

You go to Washington, D.C. You and eight other children sit down at a huge picnic table. On top of the table are nine stacks of juicy hamburgers. The stacks of burgers are so high that they reach into the sky.

The President of the United States comes to the table, wishes you all good luck, and shouts "Ready, set, go!" You and the other children start munching burgers.

Half an hour later, the President ends the contest. "Boy!" he says. "A lot of burgers were eaten here today. You each ate seven burgers, and there are nine of you. If nine people each eat seven burgers, how many burgers are eaten?"

You still have a half-eaten burger in your mouth. But you have the answer. You raise your hand. "MFFF!" you say.

"What was that?" the President asks.

You spit the hamburger out of your mouth. Now you can talk. "63!" you yell. "63 hamburgers."

"RIGHT!" says the President.

The Game ...

Program Name: **BURGER**

```
5 REM *** HAMBURGER CONTEST
10 REM MACK MCGHEE
15 REM HAM GENERATES # OF HAMBURGERS
20 REM MAN GENERATES # OF MEN
25 REM J AND A SETS   COLUMN
30 REM I SETS    ROW
35 REM FF STORES PLAYER'S ANSWER TO PROB
LEM
40 REM B AND T STORE NUMBER OF BURGERS A
ND MEN
45 REM A$ STORES PLAYER'S RESPONSE TO PL
AY AGAIN
90 DIM A$(1)
100 GRAPHICS 2+16:POKE 752,1
105 POSITION 5,4:PRINT #6;"HAMBURGER"
106 POSITION 6,6:PRINT #6;"CONTEST"
107 FOR DLAY=1 TO 1000:NEXT DLAY
120 MAN=INT(RND(0)*9+1)
121 HAM=INT(RND(0)*9+1)
125 GRAPHICS 7:POKE 752,1
126 COLOR 1
129 T=HAM:ON HAM GOTO 130,131,132,133,13
```

```
3,134,135,136,135
130 COL=0:ROW=0:GOTO 137
131 COL=0:ROW=31:GOTO 137
132 COL=0:ROW=62:GOTO 137
133 COL=21:ROW=31:GOTO 137
134 COL=21:ROW=62:GOTO 137
135 COL=42:ROW=62:GOTO 137
136 COL=42:ROW=31:GOTO 137
137 FOR J=0 TO COL STEP 21
138 FOR I=0 TO ROW STEP 31
139 PLOT 90+J,10+I:PLOT 90+J,9+I:PLOT 91
+J,8+I
140 PLOT 92+J,7+I:DRAWTO 98+J,7+I:DRAWTO
 100+J,9+I:PLOT 100+J,10+I
150 PLOT 90+J,14+I:PLOT 100+J,14+I:PLOT
90+J,13+I
160 PLOT 91+J,15+I:DRAWTO 99+J,15+I:PLOT
 100+J,13+I
180 PLOT 89+J,11+I:DRAWTO 101+J,11+I
190 PLOT 89+J,12+I:DRAWTO 101+J,12+I
191 IF HAM=4 THEN ROW=31
192 IF HAM=5 THEN ROW=62
193 IF HAM=7 OR HAM=8 THEN HAM=HAM-3
200 NEXT I
210 NEXT J
225 B=MAN
230 ON MAN GOTO 240,250,260,270,280,290,
300,310,320
240 COL=0:ROW=0:GOTO 400
250 COL=0:ROW=31:GOTO 400
260 COL=0:ROW=62:GOTO 400
270 COL=21:ROW=31:GOTO 400
280 COL=21:ROW=31:GOTO 400
290 COL=21:ROW=62:GOTO 400
300 COL=42:ROW=62:GOTO 400
310 COL=42:ROW=31:GOTO 400
320 COL=42:ROW=62:GOTO 400
400 FOR A=0 TO COL STEP 21
405 FOR I=0 TO ROW STEP 31
410 PLOT 16+A,7+I:DRAWTO 24+A,7+I
430 PLOT 25+A,8+I:DRAWTO 25+A,11+I
440 PLOT 24+A,12+I:PLOT 23+A,13+I
450 PLOT 23+A,14+I:PLOT 22+A,15+I
460 PLOT 21+A,16+I:DRAWTO 19+A,16+I
```

```
470 PLOT 18+A,15+I
480 PLOT 15+A,8+I:DRAWTO 15+A,11+I
490 PLOT 16+A,12+I:PLOT 17+A,13+I:PLOT 1
7+A,14+I
495 PLOT 18+A,9+I:PLOT 22+A,9+I:PLOT 20+
A,10+I
500 PLOT 20+A,11+I:PLOT 19+A,13+I:DRAWTO
 21+A,13+I
506 IF MAN=4 THEN ROW=31
507 IF MAN=5 THEN ROW=62
508 IF MAN=7 OR MAN=8 THEN MAN=MAN-3
510 NEXT I
520 NEXT A
525 COLOR 2
530 PLOT 75,38:DRAWTO 80,46
540 PLOT 80,38:DRAWTO 75,46
550 TRAP 550
555 PRINT " \"
557 POKE 656,0:POKE 657,4
558 IF B=1 AND T=1 THEN PRINT "IF ";B;"
PERSON EATS ";T;" BURGER,":GOTO 565
559 IF B>1 AND T=1 THEN PRINT "IF ";B;"
PEOPLE EAT ";T;" BURGER EACH,":GOTO 565
560 IF B=1 AND T>1 THEN PRINT "IF ";B;"
PERSON EATS ";T;" BURGERS,":GOTO 565
561 IF B>1 AND T>1 THEN PRINT "IF ";B;"
PEOPLE EAT ";T;" BURGERS EACH,":GOTO 565

565 PRINT
567 POKE 657,4:PRINT "HOW MANY BURGERS A
RE EATEN";:INPUT FF
570 IF FF=B*T THEN GRAPHICS 0:POKE 752,1
:GOSUB 1010:GOTO 595
580 IF FF<>B*T THEN GRAPHICS 0:POKE 752,
1:GOSUB 2010:GOTO 125
595 GRAPHICS 0:POKE 752,1
600 POSITION 13,8:PRINT "PLAY AGAIN";:IN
PUT A$
610 IF A$="Y" THEN 120
615 IF A$<>"N" THEN 595
620 GRAPHICS 0:POKE 752,0:END
```

Background . . .

The Hamburger Contest game was designed and programmed by Mack McGhee, a student at Patrick Henry High School in Roanoke, Virginia.

Mack writes: "Working on this program was a small adventure for me. When I began taking computer science at the beginning of this year, I never dreamed that after only a couple months I would be writing a computer program for a book.

"After changing the program many times and completely starting over once, I noticed the deadline approaching. My teacher, David James, began taking us home from school at 5 or 6 every night. He carried the class's one computer to all the different kids' houses. Sometimes he would arrive with the computer at my house as late as 10:30 PM. Once, on a school night, I worked on my program until 2:30 in the morning.

"Doing this program was a good experience. I hope you have as much fun playing it as I did making it."

Highlights . . .

The computer randomly selects the number of hamburgers and people with RND functions on lines 120 and 121. The number of people and hamburgers varies between one and nine.

On lines 137 to 210, the computer uses two loops (representing columns and rows) to draw the pictures of the hamburgers on the TV screen.

On lines 400 to 520, the computer uses two more loops (representing columns and rows) to draw the pictures of the people in the hamburger-eating contest.

Only a single hamburger and only a single person is actually plotted. But the loops vary the columns and rows so that between one and nine copies of the hamburger and person are made on the screen.

Variables ...

DLAY	Delay-loop counter.
A$	Your answer to the question PLAY AGAIN?
HAM	Random number of hamburgers selected by the computer.
MAN	Random number of people selected by the computer.
T	Stores same value as HAM.
B	Stores same value as MAN.
J	Column position of a hamburger displayed on the TV screen.
I	First-loop (lines 137-210)—row position of a hamburger; Second-loop (lines 400-520)—row position of a person.
COL	First-loop—Last column of last hamburger; Second-loop—Last column of last person.
ROW	First-loop—Last row of last hamburger; Second-loop—Last row of last person.
A	Column position of a person displayed on the TV screen.
FF	Your answer—the number of burgers that you think are eaten.

Do-It-Yourself ...

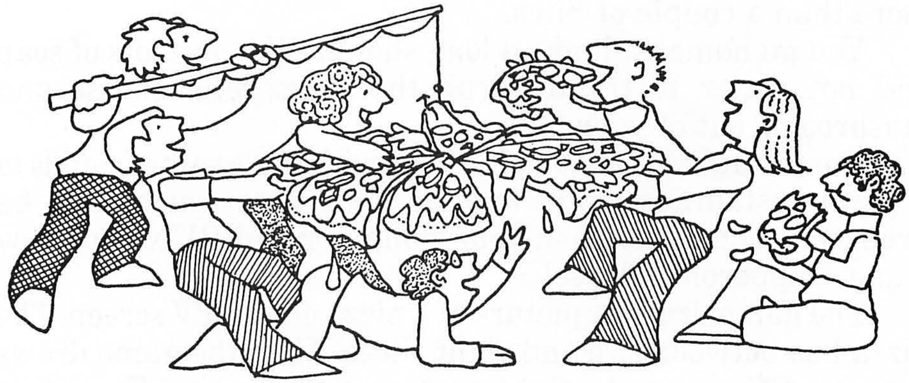
You can add a wrong-answer counter to the game. You can increment the counter by one each time a child gives the wrong answer. If a child gives the wrong answer three times in a row, you can have the computer print out the correct answer and then give the child the same problem again.

You can experiment with ways to make the color of the people different from the color of the hamburgers.

You can add sound-effects to the game, such as the munching noises you might hear in a real hamburger contest.

How about making the people different shapes? You can have the computer draw fat faces and skinny faces, faces with long hair and short hair, and big noses and little noses.

10 PEPPERONI, PLEASE!



For Parents and Teachers ...

This game will help children practice their division and fractions.

The computer draws a pizza pie on the TV screen. It divides the pie into anywhere from two to eight slices. Then it draws a random number of people on the screen. For example, the computer might ask: If there are three people trying to eat six slices of pizza, how many slices does each person get?

When the child has correctly answered the question (or answers the question incorrectly three times in a row), the computer prints the correct answer (2 slices) and displays, as a fraction, the amount of the whole pizza each person gets (2/6 or 1/3).

For Kids ...

Imagine that one night you and a bunch of your friends go to a pizza restaurant. You order the biggest pizza, but there are a lot of people, and everyone is starved.

The pizza arrives, and people attack it. But no one has figured out how much pizza each person should get. Everyone tries to grab as many slices of pizza as they can reach. Fights break out. People have pieces sticking out of their ears, their pockets, and their socks. The pizza is gone, and nobody got more than a couple of bites.

You go home and take a long shower. You use lots of soap and hot water to try to scrub the mozzarella cheese and mushrooms out of your hair.

Now imagine that you plan to go with these same friends to a pizza restaurant again tonight. This time you plan to be prepared. You sit down at your computer and RUN your new game, Pepperoni, Please!

The game draws a picture of a pizza on the TV screen. The pizza has between two and eight slices. Then the game draws anywhere from one to eight people on the screen. The game asks: "HOW MANY SLICES PER PERSON?" and "HOW MANY SLICES IN THE WHOLE PIZZA?"

You spend an hour playing the game. Now you're ready to go to the restaurant with your friends. No matter how many people go, and no matter how many slices of pizza there are, you will know how much pizza to give each person.

The Game ...

Program Name: **PIZZA**

```
5 REM *** PEPPERONI,PLEASE!
10 REM HOWARD BOGGESS
50 REM PEPPERONI,PLEASE!
55 DIM A$(1)
60 DIM P(21)
65 GOTO 100
```

```
   70 P(1)=X+66:P(2)=Y+4:P(3)=3:P(4)=5:P(5)
   =6
   80 P(6)=7:P(7)=9:P(8)=12:P(9)=14:P(10)=1
   8
   85 P(11)=23:P(12)=27:P(13)=61:P(14)=63:P
   (15)=65
   90 P(16)=66:P(17)=67:P(18)=69:P(19)=71:P
   (20)=-X+90
   95 P(21)=-Y+90
   99 GOTO 7010
   100 GRAPHICS 2+16
   105 PRINT #6;" ":PRINT #6;" "
   110 PRINT #6;" PEPPERONI,PLEASE!"
   115 PRINT #6;" ":PRINT #6;" "
   120 PRINT #6;"  AN EXERCISE IN"
   121 PRINT #6;" "
   125 PRINT #6;"      FRACTIONS"
   130 FOR DELAY=1 TO 1000:NEXT DELAY
   200 M=INT(RND(1)*7)+2
   205 GOSUB 9210:REM * SET UP SCREEN
   210 ON M GOSUB 0,3000,3500,4000,4500,500
   0,5500,6000
»  211 PRINT " \ ":POKE 656,0:PRINT "COUNT T
   HE SLICES"
   212 FOR DELAY=1 TO 1500:NEXT DELAY
»  213 PRINT " \ ":POKE 656,0:POKE 657,17:PR
   INT "COUNT THE PEOPLE"
   215 ON M GOTO 0,225,240,255,270,285,300,
   315
   220 REM 2 PEOPLE
   225 N=M
   230 GOTO 330
   235 REM 3 PEOPLE
   240 N=M
   245 GOTO 330
   250 REM 2 or 4 PEOPLE
   255 N=4/(INT(RND(0)*2+1))
   260 GOTO 330
   265 REM 5 PEOPLE
   270 N=M
   275 GOTO 330
   280 REM 2,3 or 6 PEOPLE
   285 N=6/(INT(RND(0)*3+1))
   290 GOTO 330
```

```
295 REM 7 PEOPLE
300 N=M
305 GOTO 330
310 REM 2,4 or 8 PEOPLE
315 N=(INT(RND(0)*4+1))
320 IF N=3 THEN GOTO 315
322 N=8/N
325 GOTO 330
330 GOSUB 7000:REM -DRAW PEOPLE
331 FOR DELAY=1 TO 900:NEXT DELAY
332 TRAP 332
333 PRINT "↰"
334 POKE 656,0:POKE 657,6
335 PRINT "HOW MANY SLICES PER PERSON";:
INPUT YY
336 PRINT "↰"
337 POKE 656,0
340 PRINT "HOW MANY SLICES IN THE WHOLE
PIZZA";:INPUT YZ
380 IF YY=M/N AND YZ=M THEN GOTO 395
382 PRINT "↰"
383 COUNT=COUNT+1
384 IF COUNT>2 THEN COUNT=0:GOTO 397
385 POKE 656,0:POKE 657,10:PRINT "SORRY!
! TRY AGAIN!!"
387 FOR DELAY=1 TO 500:NEXT DELAY
390 GOTO 331
395 GRAPHICS 0:POKE 752,1:GOSUB 1010:REM
 HAPPY FACE
397 GOSUB 8010:REM * DISPLAY FRACTION
400 GRAPHICS 0:POKE 752,1
410 POSITION 13,8:PRINT "PLAY AGAIN";:IN
PUT A$
420 IF A$="Y" THEN 200
430 IF A$<>"N" THEN 400
440 GRAPHICS 0:POKE 752,0:END
3000 REM 2 SLICES OF PIZZA
3010 PLOT 20,40:DRAWTO 60,40
3015 GOSUB 9000
3020 RETURN
3500 REM 3 SLICES OF PIZZA
3510 PLOT 22.8,50.2:DRAWTO 40,40
3520 PLOT 58,50.2:DRAWTO 40,40
3530 PLOT 40,20:DRAWTO 40,40
```

```
3535 GOSUB 9000
3540 RETURN
4000 REM 4 SLICES OF PIZZA
4010 PLOT 20,40:DRAWTO 60,40
4020 PLOT 40,20:DRAWTO 40,60
4025 GOSUB 9000
4030 RETURN
4500 REM 5 SLICES OF PIZZA
4510 PLOT 40,60:DRAWTO 40,40
4520 PLOT 21.2,46.823:DRAWTO 40,40
4530 PLOT 58.8,46.823:DRAWTO 40,40
4540 PLOT 52,24:DRAWTO 40,40
4550 PLOT 28,24:DRAWTO 40,40
4555 GOSUB 9000
4560 RETURN
5000 REM 6 SLICES OF PIZZA
5010 PLOT 22.8,50.2:DRAWTO 40,40
5020 PLOT 58,50.2:DRAWTO 40,40
5030 PLOT 40,20:DRAWTO 40,40
5040 PLOT 40,60:DRAWTO 40,40
5050 PLOT 21.5,31:DRAWTO 40,40
5060 PLOT 58,31:DRAWTO 40,40
5065 GOSUB 9000
5070 RETURN
5500 REM 7 SLICES OF PIZZA
5510 PLOT 20.4,36.02:DRAWTO 40,40
5520 PLOT 23.6,51.447:DRAWTO 40,40
5530 PLOT 24,52:DRAWTO 40,40
5540 PLOT 40,60:DRAWTO 40,40
5550 PLOT 56,52:DRAWTO 40,40
5560 PLOT 48,21.669:DRAWTO 40,40
5570 PLOT 31.6,21.849:DRAWTO 40,40
5580 PLOT 58,31:DRAWTO 40,40
5585 GOSUB 9000
5590 RETURN
6000 REM 8 SLICES OF PIZZA
6010 PLOT 26,25.718:DRAWTO 54,54.282
6020 PLOT 54,25.718:DRAWTO 26,54.282
6030 PLOT 40,20:DRAWTO 40,60
6040 PLOT 20,40:DRAWTO 60,40
6045 GOSUB 9000
6050 RETURN
7000 REM FIGURE SUB-ROUTINE
7005 GOTO 70
```

```
7010 A=54:B=4
7015 H=3
7020 FOR S=1 TO N
7025 A=A+12
7030 FOR X=-H TO H STEP 0.4
7035 Y=SQR(H*H-X*X)
7040 PLOT X+A,Y+B
7045 PLOT -X+A,-Y+B
7050 NEXT X
7055 IF S>N THEN RETURN
7060 IF S=1 THEN GOTO 7070
7065 ON S GOTO 0,7135,7135,7135,7135,713
5,7135,7135
7070 PLOT P(15),P(3):PLOT P(17),P(3)
7075 PLOT P(15),P(4):PLOT P(16),P(5):PLO
T P(17),P(4)
7080 PLOT P(16),P(6):DRAWTO P(16),P(10)
7085 PLOT P(16),P(7):DRAWTO P(19),P(8)
7090 PLOT P(16),P(7):DRAWTO P(13),P(8)
7095 PLOT P(13),P(8):DRAWTO P(16),P(9)
7100 PLOT P(19),P(8):DRAWTO P(16),P(9)
7105 PLOT P(16),P(10):DRAWTO P(18),P(11)

7110 PLOT P(16),P(10):DRAWTO P(14),P(11)

7115 PLOT P(18),P(11):DRAWTO P(18),P(12)

7120 PLOT P(14),P(11):DRAWTO P(14),P(12)

7125 NEXT S
7130 IF S>N THEN RETURN
7135 FOR Z=13 TO 20
7140 P(Z)=P(Z)+12
7145 NEXT Z
7150 GOTO 7070
8000 REM *** DISPLAY FRACTION
8010 GRAPHICS 2+16
8015 POSITION 0,1
8017 IF M/N=1 THEN PRINT #6;M/N;" SLICE
PER PERSON":GOTO 8030
8020 PRINT #6;M/N;" SLICES PER PERSON"
8030 POSITION 0,3
8040 PRINT #6;M;" TOTAL PIZZA SLICES"
8050 POSITION 2,6
```

```
   8060 PRINT #6;"EACH PERSON EATS"
   8070 POSITION 2,8
   8080 PRINT #6;M/N
   8090 PRINT #6;"  _  OF THE PIZZA."
   8095 POSITION 2,11
   8100 PRINT #6;M
   8110 FOR DELAY=1 TO 1500:NEXT DELAY
   8120 RETURN
   9000 REM *** DRAW PIZZA
   9010 H=20
   9045 REM For faster pizza,remove STEP 0.
   4 in line 9050
   9050 FOR X=-H TO H STEP 0.4
   9060 Y=SQR(H*H-X*X)
   9070 PLOT X+40,Y+40
   9080 PLOT -X+40,-Y+40
   9090 NEXT X
   9100 RETURN
   9200 REM *** INITIALIZE SCREEN
   9210 GRAPHICS 7
   9215 POKE 752,1
   9220 SETCOLOR 0,1,8
   9230 COLOR 1
➧  9235 PRINT "    PIZZA  PIE"
   9240 RETURN
```

Typing Hints ...

➧ In order to print the words on lines 211, 213, 335, 340, and 9235 in reverse video, press the Atari-symbol button (⩚). To turn off reverse video, press the button again.

Background ...

The Pepperoni, Please! game was designed and programmed by Howard Boggess, a student at Patrick Henry High School in Roanoke, Virginia.

Howard has been using computers for five years. In 1978 his dad bought him an INTERACT computer. Although the computer cost $500, it was primitive compared to today's computers. It didn't understand decimal numbers or **strings** (letters and words). Still, for Howard, it was an excellent learning tool.

Howard began working on this game, Pepperoni, Please!, two weeks before it was due. But he was plagued with bad luck. He writes: "I took the Atari home over a weekend, with only four days left to finish. I had finished six out of seven possible pizzas by Sunday morning at 3 AM. I was watching the snow fall outside my bedroom window and putting the finishing touch on the seventh pizza. That's when all the power went out and I lost everything!"

Howard worked all day Sunday on his program. He went to school Monday, but he got out of some of his classes to continue working on the program. He continues: "The program was due on Tuesday. Monday night I took the computer home and pieced together the whole program in three hours, eating pizza the whole time. After the program was finished I thought I'd be too sick to eat pizza ever again, but that's what I had for dinner that night."

Different people helped Howard in his efforts to get his program in on time. His teacher, David James, rushed computers and computer equipment back and forth between the school and Howard's house. And Howard's classmate in school, (another Howard) Howard Levine, rescued him from trying to draw the pizzas and the people's heads using DATA commands. Howard Levine pointed out that the pizzas and heads were circles. The boys could use the formula for a circle to draw them. Howard Levine grabbed his little notebook, thought for 30 seconds, and then wrote out the circle-plotting formula used in the program on lines 7030 and 7050 and on lines 9010 to 9090.

Highlights . . .

The computer draws the pizza and the people in Graphics Mode 7.

The subroutine beginning on line 9000 draws the pizza pie. The subroutine beginning on line 7000 draws the people.

The seven subroutines to draw the different number of pizza slices (randomly selected, from 2 to 8) begin on lines 3000, 3500, 4000, 4500, 5000, 5500, and 6000.

The subroutine to display the answer and the fraction of pizza allotted to each person begins on line 8000.

The computer chooses the number of pizza slices using an RND function on line 200.

Lines 220 to 325 take care of matching the number of randomly chosen pizza slices with the right number of people. To simplify the exercise, the computer chooses a number that results in each person getting whole slices of pizza (no fractional slices or pieces left over).

Variables ...

A$	Your answer to the question PLAY AGAIN?
P	21-Element array—stores PLOT points on the TV screen for people.
DELAY	Delay-loop counter.
DLAY	Delay-loop counter.
M	Number of slices in the pizza.
N	Number of people drawn on the screen.
YY	Your Answer—Number of slices per person.
YZ	Your Answer—Number of slices in the whole pizza.
A	The beginning X-value (column) for the people.
B	The beginning Y-value (row) for the people.
H	The radius of the pizza and the people's heads.
X	The column value for the circles.

Y	The row value for the circles.
S	The number of people on the TV screen.
Z	The amount added to the position of each person's figure so that the next figure can be plotted.

Do-It-Yourself . . .

You might experiment with making the pizza pie and the people appear in different colors. You can do this by changing the setting on the SETCOLOR and COLOR commands before drawing either the pizza or the people.

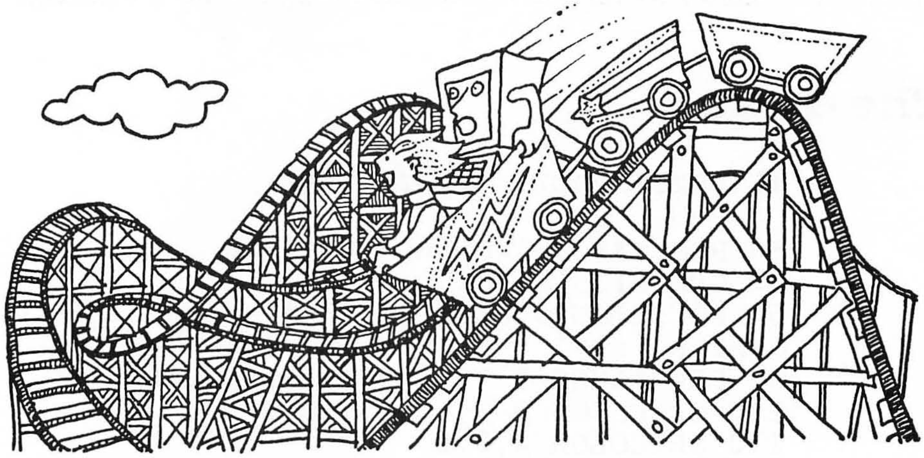
Also, by using the XIO command you can fill in pizza slices with color. (To use the XIO command, see the FAST game in the chapter entitled "How Fast Are You?")

You can also add sound-effects, such as noises that sound like people munching and slurping juicy pizza pies.

Also, you can show how some of the fractions can be reduced. Sometimes the program (see the subroutine beginning on line 8000) prints fractions like 2/8, 2/4, and 3/6. It would be nice if the program showed that these fractions are equivalent to 1/4, 1/2, and 1/2.

Finally, when the child answered the question correctly, you can have the number of pizza-pie slices that go to one person turn a different color or be filled in with the XIO command.

11
THE 3-D ROLLER COASTER



For Parents and Teachers ...

This game helps children learn about the **cosine** function in trigonometry. It draws a colorful 3-D cosine curve on the TV screen. With modifications, the program can draw curves of the other trigonometric functions.

For Kids ...

How would you like to become a video artist? You can make interesting, beautiful pictures by using a few simple recipes. The recipes you use are mathematical formulas.

If you didn't use the formulas, you would have to tell the computer to draw each square or point in the picture. The formulas take care of calculating all of the points automatically.

Also, video art is a good way to spice up your math homework. By themselves, mathematical formulas can be complicated and boring. But they all describe some sort of shape. That shape might be beautiful. You can get your computer to draw the shapes on the TV screen. It makes math more interesting, and it is a lot easier than drawing the shapes yourself on a piece of graph paper.

This game is an example of video art using a cosine function. The program draws a picture of a 3-D roller coaster. The roller coaster looks like it is standing up on the TV screen.

The Game ...

Program Name: **ROLLER**

```
50 REM *** 3-D ROLLER COASTER
62 GRAPHICS 0:POKE 752,1
65 POSITION 5,8:PRINT "***  3-D ROLLER C
OASTER  ***"
66 FOR PAUSE=1 TO 1000:NEXT PAUSE
100 GRAPHICS 9
110 SETCOLOR 4,3,0
120 DEG
130 FOR X=0 TO 79
140 COLOR ABS(INT(15*COS(9*X)))
150 PLOT X,20-(INT(19*COS(9*X)))
160 DRAWTO X,160-(INT(19*COS(9*X)))
170 COLOR 7
180 PLOT X,161-(INT(19*COS(9*X))):DRAWTO
 X,191
190 NEXT X
200 GOTO 200
```

Highlights ...

This is a very short program—only 15 lines.

On line 100 you set the computer in Graphics Mode 9. (The program uses the GTIA graphics chip.)

On line 120, you use the DEG command to put the computer into "degrees" mode (as opposed to radians). Now

BASIC treats all arguments of trigonometric functions as degrees.

The program uses a single loop—from lines 130 to 190. First the program draws one line of the roller coaster. Then it draws the "space" under the roller coaster. Then it goes back to the top of the loop (line 130) and draws a new line and a new space. It continues drawing lines and spaces until it reaches the righthand edge of the screen.

Variables ...

PAUSE	Delay loop counter.
X	Loop counter—the column on the TV screen where the roller coaster will be drawn—X also alters the color of each line.

Do-It-Yourself ...

This program is more than video art—more even than 3-D video art. It enables you to visualize simple wave functions in three dimensions. It changes numbers and formulas into a beautiful picture.

The program uses the COS (cosine) trigonometric function. You can experiment with it by using other functions—such as the SIN (sine), SQR (square root), and LOG (logarithm) functions.

Using the trigonometric equivalences shown in the accompanying table of derived functions, you can also create tangent, cotangent, secant, and cosecant functions.

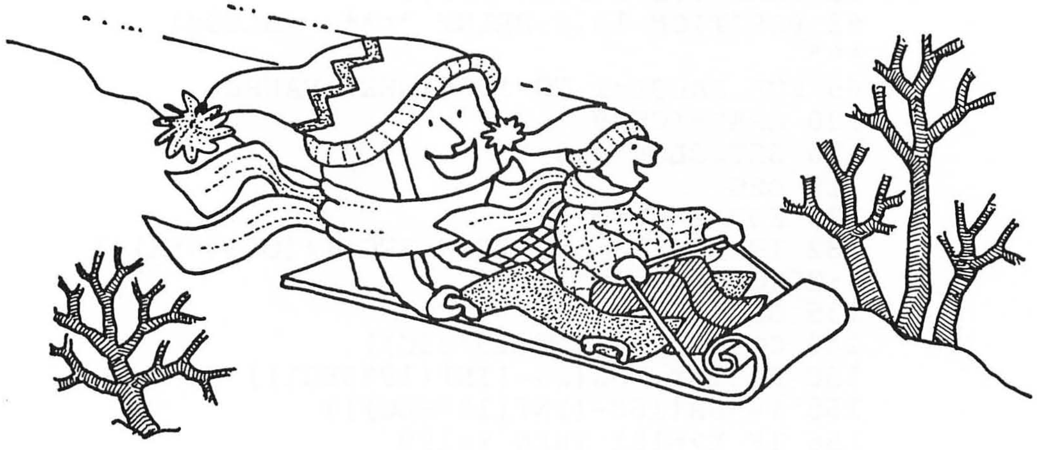
TABLE OF DERIVED TRIGONOMETRIC FUNCTIONS

```
TAN(X) = SIN(X)/COS(X)
COT(X) = COS(X)/SIN(X)
SEC(X) = 1/COS(X)
CSC(X) = 1/SIN(X)
```

To get your child to interact with these functions, you can add variables that change the shape of the roller coaster. The child can enter different numbers to see how they affect the roller coaster's shape.

You can add a SOUND command to have the computer make a new sound as it draws each new line in the roller coaster—higher sounds for roller coaster peaks and lower sounds for dips. This will heighten the illusion that the computer (or child) is riding the roller coaster.

12
SLEDS!



For Parents and Teachers ...

This game is similar to The 3-D Roller Coaster game. It helps children learn about trigonometric functions. The computer draws one function—the secant function—on the TV screen. Children can experiment with the game and make all sorts of different shapes.

For Kids ...

In the last chapter we created a 3-D Roller Coaster, and we suggested that you experiment with it using other trigonometric functions. In this chapter we look at just how you go about using the derived functions shown in the last chapter to create new pictures. By making only a couple of modifications to the roller coaster program, we end up with sleds!

The Game ...

Program Name: **SLEDS**

```
50 REM *** SLEDS
62 GRAPHICS 0:POKE 752,1
65 POSITION 10,8:PRINT "***   SLEDS!   *
**"
66 FOR PAUSE=1 TO 1000:NEXT PAUSE
100 GRAPHICS 9
110 SETCOLOR 4,3,0
120 DEG
130 FOR X=0 TO 79
132 IF COS(9*X)=0 THEN SEC=1/(COS(9*X)+1
):GOTO 140
135 SEC=1/COS(9*X)
140 COLOR ABS(INT(15*SEC))
150 PLOT X,ABS(20-(INT(19*SEC)))
155 Y=ABS(160-(INT(19*SEC)))
156 IF Y>=191 THEN Y=190
160 DRAWTO X,Y
170 COLOR 7
180 PLOT X,Y+1:DRAWTO X,191
190 NEXT X
200 GOTO 200
```

Highlights ...

This program is similar to the ROLLER program. For details take a look at the last chapter. For this example I chose the first and simplest function—secant. To get the secant, you invert the cosine:

SEC(X) = 1/COS(X)

This is simple—except for one problem. We are using degrees (see the DEG command on line 120), and when we compute the cosine of 90 degrees, the answer is 0. We get the secant function above by dividing the cosine of an angle by 1. If the cosine is 0 and we try to divide it by 1, we get an undefined number and the program crashes.

To solve this problem, we test for a cosine equal to 0 on line 132. If the cosine is equal to 0 we add 1 to it. The secant ends up being 1/1, or 1.

Later, on line 150, we have to compute the absolute value of the Y-value. If we don't, the Y-value becomes negative. You can't PLOT a point with a negative Y-value, and the program crashes. So you have to add an ABS (absolute value function) to keep the Y-value positive.

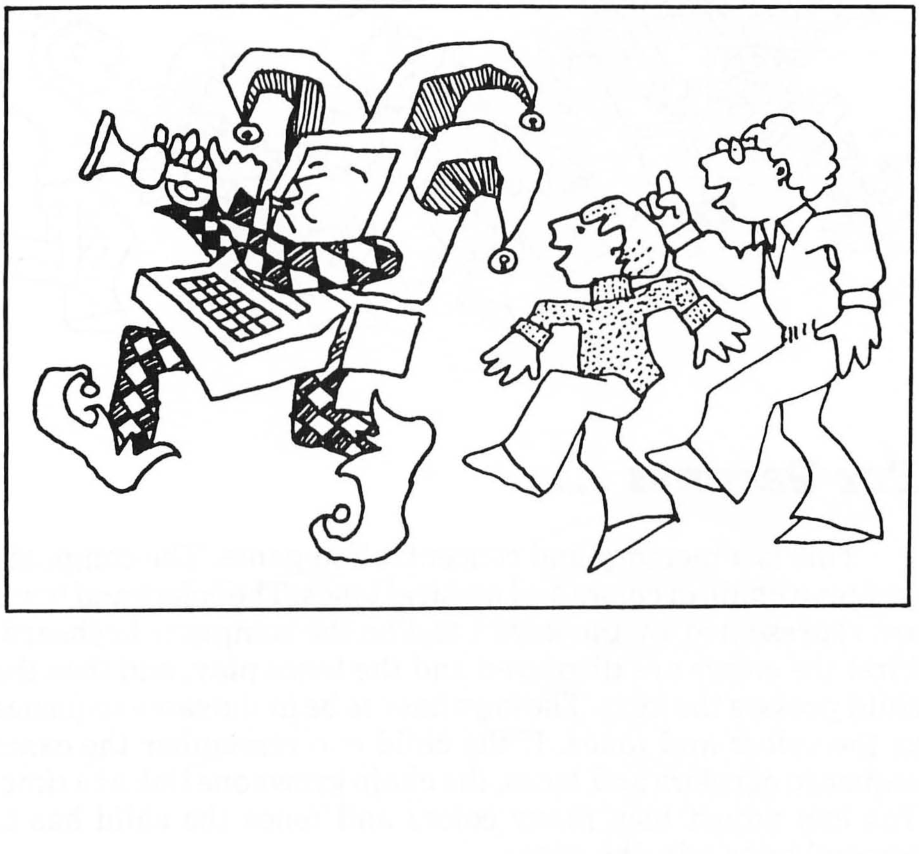
Finally, on lines 160 and 180, when the program uses the secant, the Y-value exceeds the maximum screen-column size of 191. To solve this problem, we calculate the Y-value on line 155. If the Y-value is too large, we replace it with 190. This works for the DRAWTO command on line 160. Then, on line 180, we need to add 1 to the Y-value to get 191—the largest acceptable Y-value for Graphics Mode 9.

Now the SLEDS program runs.

Do-It-Yourself ...

We've seen in this chapter that deriving new trigonometric functions can be tricky. But it's worth it. You can gain a lot of insight into the various trigonometric relationships. And, best of all, you can end up with some beautiful computer pictures.

COLORS



13
THE PIED PIPER

For Parents ...

This is a memory and concentration game. The computer creates a chain of colors and musical tones. The colors and tones are represented by the keys 1 to 4 on the computer keyboard. First the colors are displayed and the tones play, and then the child presses the keys. The keys have to be in the same sequence as the colors and tones. If the child can remember the exact sequence of colors and tones, the chain grows one link at a time. You can adjust how many colors and tones the child has to remember to win the game.

For Kids ...

In this game, the computer becomes a Pied Piper. It plays a musical tone and flashes a color on the TV screen, and you have to follow it.

The computer starts by playing a single tone and flashing a single color. If you press the right button, the same tone plays and the same color flashes on the TV screen. You won the game.

The computer asks you: "AGAIN? (PRESS RETURN)." If you press any other key, the game is over. If you press **RETURN**, the computer plays a new tone and flashes a new color. If you press the right button and match the color and tone, the computer then plays a second tone and a second color. Then it repeats the first tone and the first color. Now you have to press two buttons. The first button is for the first color and tone. The second button is for the second color and tone. If you get them both right, you win again.

But the computer still isn't done. Now it is ready to challenge you with three colors and three tones. If you press the right three buttons, then it comes back with four colors and tones—and on, and on.

How high does the computer go? The computer will play 50 games in a row. It has only 4 colors and 4 tones to work with. But it can piece these together into a string of up to 50 colors and tones. If you can remember 50 colors and tones in a row, the computer gives up. You are a genius!

The object of the game is to see how long you can follow the computer Pied Piper. It keeps stringing colors and tones together. And you try to match those colors and tones by pressing the color/tone buttons in the right order.

When the computer flashes the color red on the screen and plays a low C note, you press the **1** button.

When the computer flashes the color purple and plays a middle C note, you press the **2** button.

When the computer flashes the color blue and plays a high C note, you press the **3** button.

And when the computer flashes yellow and plays a high-high C, you press the **4** button.

What happens if the computer strings four colors and tones in a row? Here's an example. If the computer:

FLASHES	RED	RED	YELLOW	PURPLE
PLAYS	Low C	Low C	Hi-Hi-C	Middle C

You wait until the computer plays the four colors and notes. Then you press four buttons: **1 - 1 - 4 - 2**. The computer will print a happy face. You won!

If you missed one of the colors (say you typed **1 - 1 - 3 - 2**), the computer will print a sad face and then ask you if you want to start a new game.

The Game ...

Program Name: **PIPER**

```
50 REM *** PIED PIPER
60 PRINT " ↖ ":POKE 752,1
62 POSITION 7,8:PRINT "***   THE PIED PI
PER   ***"
63 FOR PAUSE=1 TO 1000:NEXT PAUSE
65 DIM KEY(31):KEY(31)=1:KEY(30)=2:KEY(2
6)=3:KEY(24)=4
70 DIM COMBO(50)
80 DIM HUE(4),LUM(4):HUE(1)=3:LUM(1)=5:H
UE(2)=6:LUM(2)=2:HUE(3)=11:LUM(3)=6:HUE(
4)=2:LUM(4)=15
90 DIM PICH(4):PICH(1)=243:PICH(2)=121:P
ICH(3)=60:PICH(4)=29
95 TOP=1
100 FOR TOTAL=1 TO TOP
105 POKE 752,1
110 LET COMBO(TOTAL)=INT(RND(0)*4)+1
120 FOR PLAY=1 TO TOTAL
130 PRINT " ↖ ":SETCOLOR 2,HUE(COMBO(PLAY
)),LUM(COMBO(PLAY)):SOUND 0,PICH(COMBO(P
LAY)),10,10
132 FOR T=1 TO 400:NEXT T:SOUND 0,0,0,0:
FOR T=1 TO 50:NEXT T
135 NEXT PLAY
140 GRAPHICS 0:POKE 752,1:POSITION 15,8:
PRINT "YOUR TURN"
145 FOR TRY=1 TO TOTAL
147 POKE 764,255
150 OPEN #1,4,0,"K:"
160 A=PEEK(764):IF A=255 THEN 160
175 IF KEY(A)<>COMBO(TRY) THEN GOSUB 201
```

```
0:CLOSE #1:POP :POP :GOTO 230
176 PRINT " \ ":POKE 752,1
177 SETCOLOR 2,HUE(COMBO(TRY)),LUM(COMBO
(TRY)):SOUND 0,PICH(COMBO(TRY)),10,10
185 A=PEEK(53775):IF A=251 THEN 185
190 CLOSE #1:SOUND 0,0,0,0
195 SETCOLOR 2,0,15
200 NEXT TRY
205 FOR PAUSE=1 TO 300:NEXT PAUSE
210 NEXT TOTAL
220 GOSUB 1010
230 OPEN #1,4,0,"K:"
234 FLAG=0
235 GRAPHICS 0:POKE 752,1:POSITION 11,8:
PRINT "AGAIN? (PRESS RETURN)"
237 A=PEEK(764):IF A=255 THEN 237
238 FLAG=FLAG+1:IF FLAG=1 THEN POKE 764,
255:GOTO 235
240 IF A=12 THEN CLOSE #1:POKE 764,255:T
OP=TOP+1:IF TOP<=50 THEN 100
260 CLOSE #1:POKE 752,0:PRINT " \ ":END
```

Highlights ...

This program runs almost completely inside of a single FOR-NEXT loop (from lines 100 to 210). The loop controls the number of colors and tones the computer will string in a row. In the first game the computer challenges you with a single color and tone (TOP=1). From then on, in each new game the computer adds a new color and tone (on line 240: TOP=TOP+1).

Within the major loop is a second loop (on lines 120 to 135). This loop displays the colors and plays the tones in a single round of a single game.

Then comes a third loop, also inside the main loop (on lines 145 to 200). This loop lets you select the colors and tones for a single round of a single game.

If you push the right button (**1**, **2**, **3**, **4**) to select the right color and tone, the computer will display the color and play the tone. It displays the color and plays the tone as long as you are pressing the button. It stops when you take your finger off the button. This is accomplished by line 185. As long as you are pressing the key, the number in memory location 53775 is 251. When you take your finger off the key, the number changes.

Also, take a look at line 238. The first time the computer displays the message "AGAIN? (PRESS RETURN)," it "burps!" The burp is in the form of a stray key. You haven't typed anything, but the key still appears. The key is not a **RETURN**, so the computer thinks you don't want to play. It stops the game.

To prevent this from happening, the FLAG variable on line 238 makes the computer disregard the first key and go back to line 235 and display the message again. This time the computer accepts your answer.

Variables ...

KEY	Array variable—used to translate the internal keyboard code to the numbers **1**, **2**, **3**, and **4**.
COMBO	Array variable—stores the number of the correct key (**1**, **2**, **3**, or **4**) for up to 50 rounds of colors and tones. The key-number is selected at random (see line 110).
HUE	Array variable—contains the color associated with each key (**1**, **2**, **3**, or **4**).
LUM	Array variable contains the color luminance associated with each key (**1**, **2**, **3**, or **4**).
PICH	Array variable—contains the tone associated with each key (**1**, **2**, **3**, or **4**).
TOTAL	FOR-NEXT loop counter—the number of colors and tones in the current round of the current game.
TOP	Counter—the total number of colors and tones in the current game.

PLAY	FOR-NEXT loop counter—the number of colors displayed and tones played by the computer in the current round of the current game.
T	FOR-NEXT loop counter—delay loop to play the computer's tone and display the computer's picture for a certain length of time.
TRY	FOR-NEXT loop counter—the number of colors and tones you have to select in the current round of the current game.
A	Contains the internal keyboard code of the number button you press.
PAUSE	FOR-NEXT loop counter—delay loop (line 205).
FLAG	Makes the computer disregard its own "burp" (an extraneous key transmitted to the screen editor).

Do-It-Yourself ...

You might want to change the TOP variable to something less miraculous than 50. For example, if you have a five-year-old playing the Pied Piper game, you might set TOP (see line 240) to go only up to 5 or 10. Then, if the child is able to follow the Pied Piper for 5 or 10 colors and tones, you can reward her. You can put some PRINT commands after line 240 congratulating her on doing so well.

Also, you can put some PRINT commands up front to introduce the Pied Piper and explain the rules of the game.

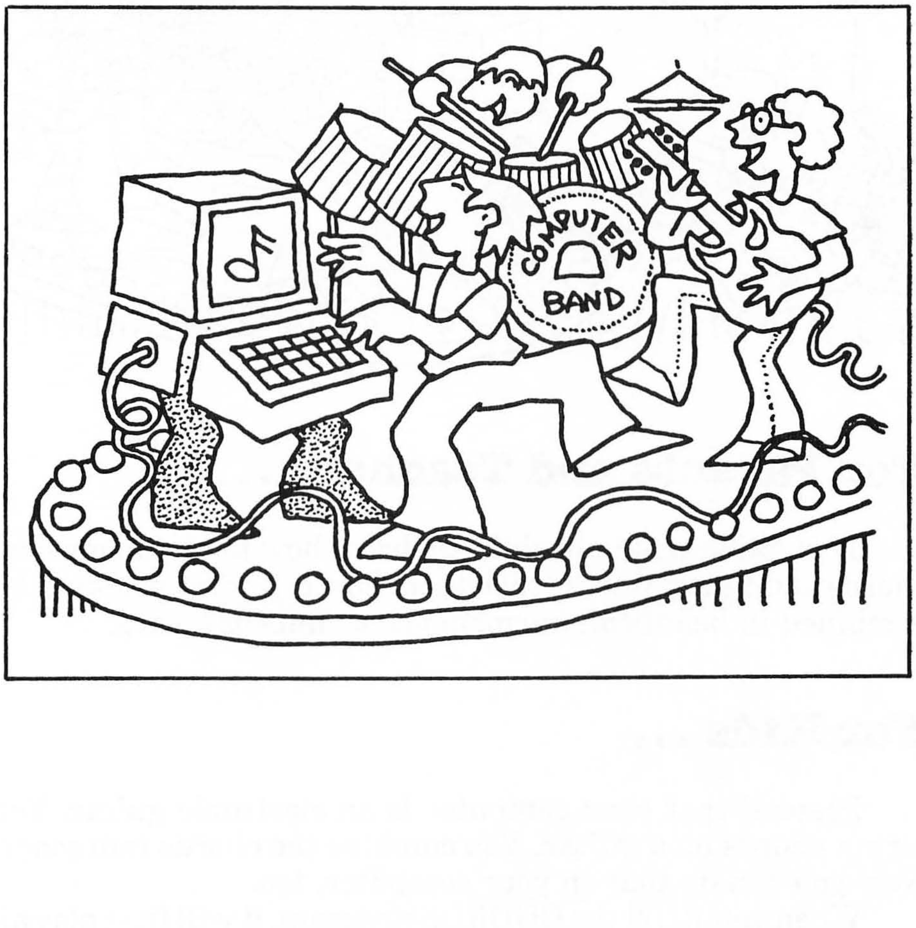
You can also make the game harder. You can create a Super Piper game by adding new tones and colors. Presently the game has only four colors and tones to work with. If you double this number, a child will have to keep track of eight sounds and eight tones strung together in combinations of up to 50 sounds and tones at a time.

You can also speed the game up. You can make the computer play the tones and display the colors in a flash by modifying the delay loop on line 132. For example, you can change the loop from FOR T=1 TO 400 to FOR T=1 TO 100.

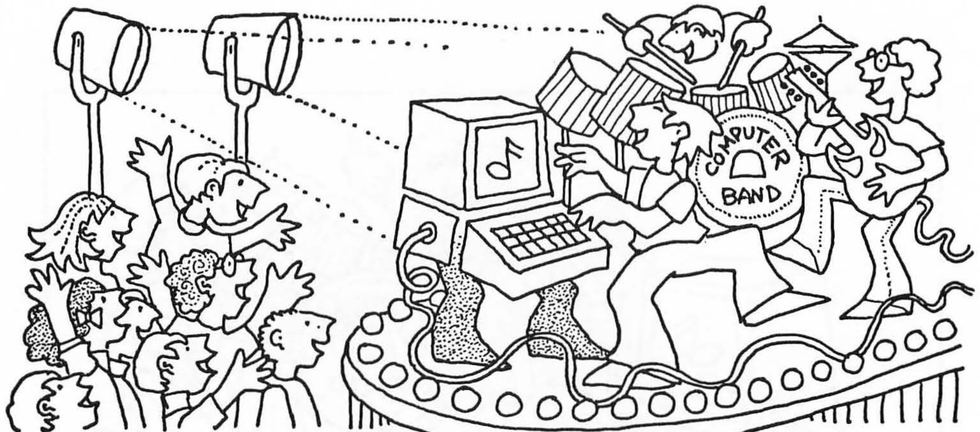
If anyone ever beat the computer at the Super Piper game, it would be a true feat of mental wizardry.

Perhaps you and your children are up to the challenge!

MUSIC



14 MAKE UP A SONG



For Parents and Teachers ...

This game will help children learn how to create musical chords and tunes. It will teach them how different notes can be combined in beautiful, harmonious, or unusual ways.

For Kids ...

Pretend that your computer is an electronic guitar. You strum chords on a guitar. You combine the chords into songs. Now you can do that on your computer, too.

When you RUN the CHORDS program, it will first play all the notes between middle C and high C (including the sharps). Listen carefully to these notes as the computer plays them.

Then the computer will ask you to "PICK A NOTE." Actually you get to pick out a chord consisting of three notes. A chord is a combination of two or more notes played together. To choose the notes in your chord, you need to press only one of two buttons—the **SPACE** bar or the **RETURN** button. You press the **RETURN** button to select a note to go in your chord. You press the **SPACE** bar to change the notes on the TV screen until the note displayed is the one you want to select.

You can create a song out of chords—one chord played right after the other. This game will let you create a song with up to 10 chords.

After you have created a single chord of three notes, the computer will ask you if you want to create a new chord. Answer "Y" or "YES" until you have created all the chords in your song.

When you answer "N" or "NO," the computer will play the chords in your song. First it will show how the chords are built out of the individual notes you selected. Then it will play the notes together as chords. It will play the chords one right after the other as a song.

After the computer has played all the chords, it will ask you if you want to hear the old chords again. If you type "N" or "NO," it will ask you if you want to make new chords. If you don't, the game will end. But if you do want to make new chords and answer "Y" or "YES," the computer will erase your old song and let you create a new song.

The Game ...

Program Name: **CHORDS**

```
50 REM *** PLAY CHORDS GAME
60 PRINT "↰":POKE 752,1
65 POSITION 5,8:PRINT "***   PLAY-A-CHOR
D GAME   ***"
67 FOR PAUSE=1 TO 1000:NEXT PAUSE
70 DIM A$(1)
75 DIM PLAY(2,10)
80 DIM KEY(13),SHARP(13)
90 DIM MUSIC(13)
92 GOSUB 3010
100 FIRST=1:LAST=13:HOP=1:GOSUB 4010
```

```
103 FOR PAUSE=1 TO 300:NEXT PAUSE
105 FIRST=13:LAST=1:HOP=-1:GOSUB 4010
106 FOR PAUSE=1 TO 300:NEXT PAUSE
115 OPEN #1,4,0,"K:"
117 FOR C=1 TO 10
118 GRAPHICS 2:POKE 752,1
119 GOSUB 3510
120 FOR O=0 TO 2
140 FOR J=1 TO 13
142 POSITION 8,5:PRINT #6;"    "
145 IF J<>13 THEN 150
147 POSITION 8,5:PRINT #6;"HI-";CHR$(KEY
(J))
148 GOTO 160
150 POSITION 9,5:PRINT #6;CHR$(KEY(J));C
HR$(SHARP(J))
160 SOUND 0,MUSIC(J),10,15
170 GET #1,K
180 IF K=32 THEN 230
190 IF K<>155 THEN 170
200 PLAY(O,C)=J
210 POP :GOTO 245
230 NEXT J
235 GOTO 140
245 NEXT O
247 IF C>=10 THEN POP :GOTO 300
250 GRAPHICS 0:POKE 752,1
260 POSITION 12,8:PRINT "NEW CHORD";:INP
UT A$
270 IF A$="Y" THEN 290
280 IF A$<>"N" THEN 250
285 POP :GOTO 300
290 NEXT C
300 GOSUB 4510
310 GOSUB 4710
320 GRAPHICS 0:POKE 752,1
330 POSITION 5,8:PRINT "PLAY OLD CHORDS 
AGAIN";:INPUT A$
335 IF A$="Y" THEN 310
337 IF A$<>"N" THEN 320
340 GRAPHICS 0:POKE 752,1
343 POSITION 8,8:PRINT "PLAY NEW CHORDS"
;:INPUT A$
345 IF A$="Y" THEN 117
```

```
347 IF A$<>"N" THEN 340
350 CLOSE #1:GRAPHICS 0:POKE 752,0:END
3000 REM *** LOAD KEYS
3010 FOR I=1 TO 13
3020 READ K,M,S
3030 MUSIC(I)=K
3035 KEY(I)=M
3038 SHARP(I)=S
3040 NEXT I
3050 RETURN
3500 REM *** SCREEN MESSAGE
3510 PRINT "     ***    PICK A NOTE    ***"

3520 RETURN
4000 REM *** PLAY NOTES
4010 GRAPHICS 2:POKE 752,1
4012 PRINT "    *** LISTEN TO THE NOTES *
**"
4015 FOR J=FIRST TO LAST STEP HOP
4020 SOUND 0,MUSIC(J),10,15
4040 POSITION 8,5:PRINT #6;"     "
4050 IF J<>13 THEN 4080
4060 POSITION 8,5:PRINT #6;"HI-";CHR$(KE
Y(J))
4070 GOTO 4090
4080 POSITION 9,5:PRINT #6;CHR$(KEY(J));
CHR$(SHARP(J))
4090 FOR PAUSE=1 TO 100:NEXT PAUSE
4100 NEXT J
4105 SOUND 0,0,0,0
4110 RETURN
4500 REM *** PLAY NOTES IN CHORDS
4510 GRAPHICS 2:POKE 752,1
4512 PRINT "    *** NOTES IN THE CHORDS *
**"
4513 FOR F=1 TO C
4515 FOR E=0 TO 2
4520 SOUND 0,MUSIC(PLAY(E,F)),10,15
4550 IF PLAY(E,F)<>13 THEN 4580
4560 POSITION 8,5-2*E:PRINT #6;"HI-";CHR
$(KEY(PLAY(E,F)))
4570 GOTO 4590
4580 POSITION 9,5-2*E:PRINT #6;CHR$(KEY(
PLAY(E,F)));CHR$(SHARP(PLAY(E,F)))
```

```
4590 FOR PAUSE=1 TO 50:NEXT PAUSE
4595 NEXT E
4596 SOUND 0,0,0,0
4600 FOR PAUSE=1 TO 200:NEXT PAUSE
4603 GOSUB 4650
4605 NEXT F
4610 RETURN
4650 REM *** ERASE SCREEN SUBROUTINE
4655 FOR I=5 TO 1 STEP -2
4660 POSITION 8,I:PRINT #6;"      "
4670 NEXT I
4680 RETURN
4700 REM *** PLAY CHORDS
4710 GRAPHICS 2:POKE 752,1
4712 PRINT "   *** LISTEN TO THE CHORDS
***"
4713 FOR A=1 TO 3
4715 FOR F=1 TO C
4730 FOR E=0 TO 2
4750 IF PLAY(E,F)<>13 THEN 4780
4760 POSITION 8,5-2*E:PRINT #6;"HI-";CHR
$(KEY(PLAY(E,F)))
4770 GOTO 4795
4780 POSITION 9,5-2*E:PRINT #6;CHR$(KEY(
PLAY(E,F)));CHR$(SHARP(PLAY(E,F)))
4795 NEXT E
4796 SOUND 0,MUSIC(PLAY(0,F)),10,15
4797 SOUND 1,MUSIC(PLAY(1,F)),10,15
4798 SOUND 2,MUSIC(PLAY(2,F)),10,15
4800 FOR PAUSE=1 TO 200:NEXT PAUSE
4803 SOUND 0,0,0,0:SOUND 1,0,0,0:SOUND 2
,0,0,0
4804 GOSUB 4850
4805 NEXT F
4806 FOR PAUSE=1 TO 100:NEXT PAUSE
4807 NEXT A
4810 RETURN
4850 REM *** ERASE SCREEN SUBROUTINE
4855 FOR I=5 TO 1 STEP -2
4860 POSITION 8,I:PRINT #6;"      "
4870 NEXT I
4880 RETURN
5000 DATA 121,67,32
5010 DATA 114,67,35
```

```
5020 DATA 108,68,32
5030 DATA 102,68,35
5040 DATA 96,69,32
5050 DATA 91,70,32
5060 DATA 85,70,35
5070 DATA 81,71,32
5080 DATA 76,71,35
5090 DATA 72,65,32
5100 DATA 68,65,35
5110 DATA 64,66,32
5120 DATA 60,67,32
```

Highlights ...

Look at the DATA commands on lines 5000 to 5120. Each DATA command has three values: a musical tone, a code for the button that selects the note (a "C," a "D," an "E," etc.), and the Atari ASCII code for a space or a "#" symbol. The "#" symbol signifies that the note is a **sharp**. If the third value is not a "sharp" (an ASCII 35), it is a space (an ASCII 32). This causes an empty space to be printed after the note for those notes that are not sharps.

Each time you select a note, the value of the note (a pointer to the actual note values stored in the MUSIC array and the note-name values stored in the KEY array) is stored in a two-dimensional (rows-and-columns) array called PLAY. There are three rows in the PLAY array to store the three notes in each chord. There are 10 columns in the PLAY array to store the (up to) 10 chords you have selected for your song.

The two new chief subroutines are the Play Notes in Chords subroutine beginning on line 4500 and the Play Chords subroutine beginning on line 4700. These subroutines look at the PLAY array and select the chords, one at a time. The Play Notes subroutine plays the individual notes in each chord. The Play Chords subroutine plays all three notes together to make a single chord sound. It also plays all the chords in the song.

Variables ...

PLAY	Two-dimensional array—the values in PLAY act as pointers to the musical note values stored in the MUSIC array and the note-name values stored in the KEY array.
SHARP	Stores either a space or a sharp value (#) to go with each musical note displayed on the TV screen.
C	Loop counter—major loop (controls number of chords chosen—you may choose up to 10 chords).
O	Loop counter—controls the three notes you choose for each chord.
J	Loop counter—controls the 13 notes that you have to choose from (displays the notes on the TV screen).
F	Loop counter in Play Notes in Chords subroutine—controls playing of each of up to 10 chords you have selected.
E	Loop counter in Play Notes in Chords subroutine—controls playing of the three notes in each chord.
A	Loop counter in Play Chords subroutine—controls number of times your song is played (three times).
N	Stores the random note (any note from middle C to high C).
O	Loop counter—major program loop (controls the number of notes you will be challenged with).
J	Loop counter—in subroutine beginning at 4000 (controls playing all the notes in the octave).

FIRST	First note to be played in the loop.
LAST	Last note played in the loop.
HOP	Increment by which the loop is increased or decreased. The first time around, HOP is +1, so the computer plays the octave forward from middle C to high C. The second time, HOP is -1, so the computer plays the octave **backwards** from high C to middle C.

Do-It-Yourself ...

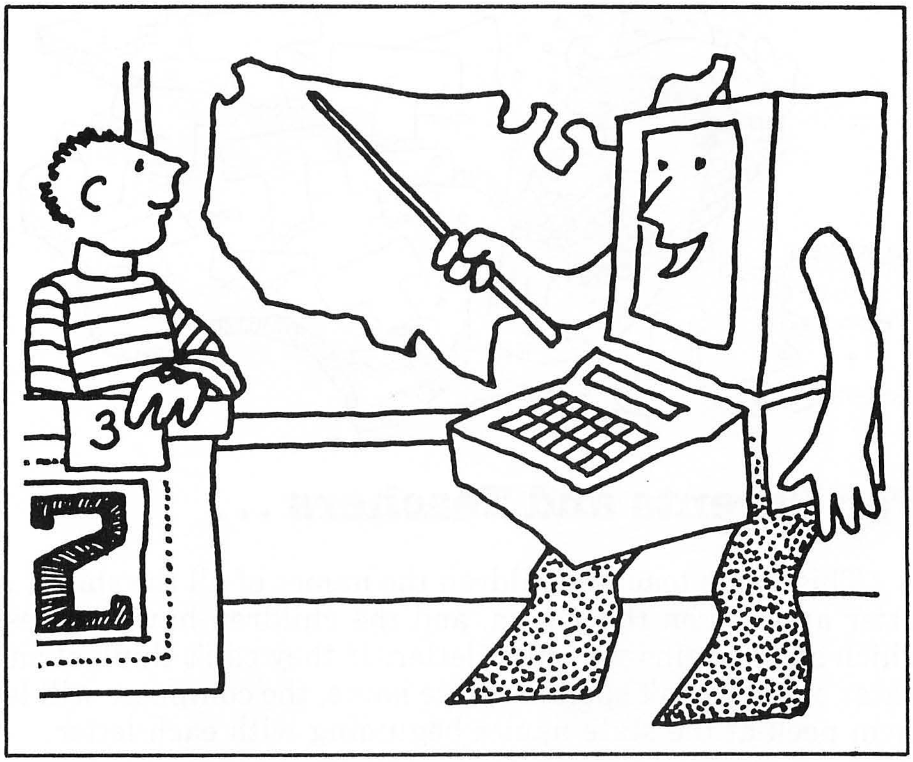
This program can be simplified or made even more elaborate. For example, you might want to create a simpler game in which children compose a song out of notes instead of chords. Or you can load in a larger number of notes (see the table of Pitch Values for Musical Notes), so that children can create chords from up to three octaves on the musical keyboard. Also, since the Atari gives you up to four voices (SOUND 0, SOUND 1, SOUND 2, and SOUND 3), you can create chords having up to four notes.

Finally, you might also consider giving children the chance to select the tempo of the song and other components of the musical score such as note duration. For example, children could select eighth-notes, quarter-notes, half-notes, or whole-notes.

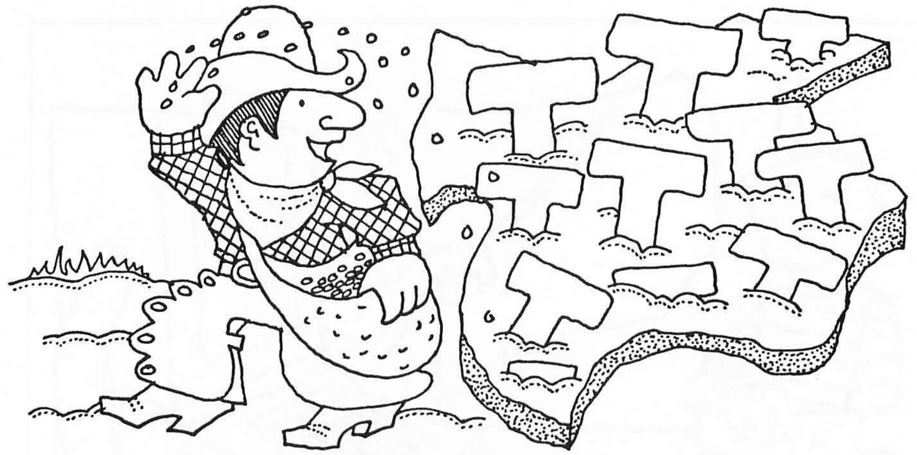
PITCH VALUES FOR MUSICAL NOTES

HIGH NOTES		MIDDLE C		LOW NOTES	
C	29	C	121	D	193
B	31	B	128	D#, E♭	204
A#, B♭	33	A#, B♭	136	D	217
A	35	A	144	C#, D♭	230
G#, A♭	37	G#, A♭	153	C	243
G	40	G	162		
F#, G♭	42	F#, G♭	173		
F	45	F	182		
E	47				
D#, E	50				
D	53				
C#, D♭	57				
C	60				
B	64				
A#, B	68				
A	72				
G#, A♭	76				
G	81				
F#, G♭	85				
F	91				
E	96				
D#, E♭	102				
D	108				
C#, D♭	114				

KNOWLEDGE



15
AL'S TOUR OF THE STATES



For Parents and Teachers ...

This game teaches children the names of all the states. A letter appears on the screen, and the children have to guess which state begins with that letter. If they can't think of any states or they can't spell the state name, the computer will let them peek at the state names beginning with each letter.

For Kids ...

Here's Alphabet Al sowing "alphabet" seeds all over the country. When Al visits a state he finds out the first letter in the state's name and then plants only that letter. For example, Al plants T's all over Texas, A's all over Alaska, and N's all over New Jersey.

Al doesn't plant any B's, E's, J's, Q's, X's, Y's, or Z's, since there are no states beginning with those letters.

After Al enters a new state and plants a letter, he stops and asks you to guess what state he is in. If you guess a state that begins with the letter Al has just planted, you win. If not, Al lets you try again.

If you can't think of a state beginning with the right letter, just press the **RETURN** button. Al will flash all the states beginning with that letter on the TV screen. When you're through studying the state names, just press **RETURN** again (or any other key). Then Al gives you another chance.

The Game

Program Name: **STATES**

```
10 REM *** AL'S TOUR OF THE STATES
20 DIM A$(25),B$(14)
50 GRAPHICS 0
60 POKE 752,1
65 POSITION 2,8:PRINT "****  AL'S TOUR O
F THE STATES  ****"
66 FOR PAUSE=1 TO 1000:NEXT PAUSE
70 FOR X=65 TO 90
75 IF X=66 OR X=69 OR X=74 OR X=81 OR X>
87 THEN 168
80 PRINT "↖"
100 COLOR X
110 POKE 752,1
120 PRINT "↖"
125 FOR I=1 TO 25
130 DRAWTO INT(RND(1)*39),INT(RND(1)*24)

140 SOUND 0,INT(RND(1)*220)+10,10,10
142 NEXT I
144 SOUND 0,0,0,0
145 FOR PAUSE=1 TO 250:NEXT PAUSE
146 PRINT "↖"
150 POSITION 5,5:PRINT "WHAT STATE IS AL
 IN";:INPUT A$
```

```
154 IF A$="" THEN GOSUB 3010:GOTO 120
156 RESTORE
157 FOR STATE=1 TO 50:READ B$:IF ASC(A$(
1))<ASC(B$(1)) THEN GOSUB 2010:POP :GOTO
 120
158 IF B$=A$ AND ASC(A$(1))=X THEN GOSUB
 1010:POP :GOTO 168
163 NEXT STATE
164 GOSUB 2010:GOTO 120
168 NEXT X
170 POKE 752,0
180 END
3000 REM ***DISPLAY STATES
3010 POSITION 3,8:PRINT "STATES BEGINNIN
G WITH THE LETTER ";CHR$(X)
3020 POSITION 3,9:PRINT "_______________
___________________"
3025 RESTORE :FOR NUM=1 TO 50:READ B$:IF
 ASC(B$(1))=X THEN NUM=50
3028 NEXT NUM
3030 POSITION 9,11:PRINT B$
3040 FOR ROW=12 TO 20
3050 READ B$
3060 IF ASC(B$(1))>X THEN POP :GOTO 3100

3070 POSITION 9,ROW:PRINT B$
3090 NEXT ROW
3100 OPEN #1,4,0,"K:"
3110 A=PEEK(764):IF A=255 THEN 3110
3120 POKE 764,255
3130 CLOSE #1
3140 RETURN
4000 DATA ALABAMA,ALASKA,ARIZONA,ARKANSA
S,CALIFORNIA,COLORADO,CONNECTICUT,DELAWA
RE,FLORIDA,GEORGIA,HAWAII,IDAHO
4010 DATA ILLINOIS,INDIANA,IOWA,KANSAS,K
ENTUCKY,LOUISIANA,MAINE,MARYLAND,MASSACH
USETTS,MICHIGAN,MINNESOTA,MISSISSI
4020 DATA MISSOURI,MONTANA,NEBRASKA,NEVA
DA,NEW HAMPSHIRE,NEW JERSEY,NEW MEXICO,N
EW YORK,NORTH CAROLINA,NORTH DAKOT
4030 DATA OHIO,OKLAHOMA,OREGON,PENNSYLVA
NIA,RHODE ISLAND,SOUTH CAROLINA,SOUTH DA
```

```
KOTA,TENNESSEE,TEXAS,UTAH,VERMONT
4040 DATA VIRGINIA,WASHINGTON,WEST VIRGI
NIA,WISCONSIN,WYOMING
```

Highlights ...

On line 75, the program checks for letters that do not begin state names: B's, E's, J's, Q's, X's, Y's, or Z's. If X equals the ASCII code for that number, the program jumps to line 168 and a new code (the next letter in the alphabet) is chosen.

On line 153 the program asks what state Al is in. If you just press **RETURN**, then the program jumps to the "Display States" subroutine on line 3010.

On lines 3010-3020, the program prints out a title. Next (on lines 3025-3028) the program searches through the state names listed in the DATA statements on lines 4000-4040. The names are in alphabetical order. When the program arrives at the first state name beginning with Alphabet Al's current letter, it stops searching, and (on line 3030) prints that name.

On lines 3040-3090, the program continues searching through the list of state names. If it finds more names that begin with Al's letter, it prints those out, too.

The list of names stays on the screen until you press **RETURN** (or any other button). The program takes care of this on lines 3100-3130. On line 3110, it looks at memory position 764. If you haven't pressed a button yet, the number 255 should be there. After you press any button, the number will change.

Line 3110 is a tiny loop. The computer keeps checking memory position 764 until the number stored in it changes (you press a button).

On line 3120, the program puts a 255 back into memory position 764 for use the next time you want to see the state names printed out.

Line 3100 directs the computer's attention to the keyboard for one-button messages from you. Line 3130 turns the computer's attention away from the keyboard and back to the program.

When you enter the name of a state, it gets stored in A$. The computer checks its list of state names (DATA statements 4000 to 4040) on lines 157 and 158.

If there is a match between your answer (A$) and the state name the computer pulls from the list (B$), then (on line 158) the computer jumps to the Happy Face subroutine on line 1010. Remember to ENTER the Happy Face subroutine after you have typed in this program.

If the computer can't find a match (on lines 157 or 164), it jumps to the Sad Face subroutine. Remember to ENTER the Sad Face subroutine after you have typed in this program.

Line numbers 1000 to 2999 have been left empty so you can slip the Happy and Sad Face subroutines into this program.

Variables . . .

A$	Your Answer—what state Al is in.
B$	A state from the computer's list.
X	Counter—the ASCII code for the letters in the alphabet.
I	Counter—Al plants 25 letters in each state.
STATE	Counter—Computer searches through list of 50 states to find a match with your answer.
NUM	Counter—Computer searches through list of 50 states to display states beginning with certain letter.
ROW	Counter—Computer prints out state names on column 9 and row# ROW.
A	The number the computer finds in memory position 764.

Do-It-Yourself . . .

Al runs through the alphabet from A to Z. You can make him run through the alphabet backwards, or you can have him hop around the alphabet.

Hint #1: Look at the FOR-NEXT loop on line 70. Hint #2: A FOR-NEXT loop can go backwards. In the FOR command, remember to include STEP -1. This makes the counter go backwards from the higher number to the lower number, one number at a time.

How would you make Al print letters at random?

Hint #1: You need to change lines 70 and 160.

Hint #2: To compute a number representing a random letter of the alphabet, use the command INT(RND(1)*26)+65. This gives you a random number between 65 and 90 (the Atari ASCII codes representing the upper-case letters).

If Al's pace is too frantic and you'd like to slow him down a bit, you can add a delay loop at line 145 by typing between lines 140 and 150 this line:

```
145 FOR D=1 TO 200:NEXT D
```

You can also increase the amount of time the letters are on the screen before Al erases them and pops his question. Just increase the PAUSE loop on line 152 from 200 to 400 (or even higher).

Here is another idea. Why not have Al plant things other than letters? Letters would still fly around the screen, but they would stand for foreign countries, fruits, sports cars, rock stars, or whatever.

Hint: you will need to change the title on lines 10 and 65. You will need to change the question on line 153. You will need to change the titles on lines 3000 and 3010. You will need to replace the DATA statements on lines 4000 to 4040. You will also need to change the FOR statements on lines 157 and 3025 to the number of pizza toppings, animals, or whatever you are having Al plant (the old number is 50 for the 50 states listed in the DATA statements).

16 WHERE DO YOU LIVE?

For Parents and Teachers ...

This game helps children learn their address, including their street, apartment, city, state, and zip code. It teaches them how to format their address on an envelope.

For Kids ...

This game helps you learn your address and how to write your address on an envelope when you send a letter.

The computer asks you for your first name. Next it asks for your last name. Then it prints out your name and address, just the way you would write them on an envelope to mail to your grandparents or to a friend.

Then the computer asks you to type in your address, one part at a time. It asks you for your street, your apartment (if you live in an apartment), your city, your state, and your zip code.

If you can't remember part of your address, don't worry. Try guessing. The computer will give you three chances, then it will let you peek at the part of the address you can't remember. Then it will hide it and ask you to type it in yourself.

Maybe some parts will take dozens of tries to get just right. But the computer won't lose its temper or grumble. It never gets upset. It just keeps playing the game until you know your whole address.

The Game ...

Program Name: **ADDRESS**

```
5 REM *** LEARN YOUR ADDRESS
10 REM THIS PROGRAM HELPS YOUNGER     CH
ILDREN TO LEARN THEIR ADDRESS
20 REM MELISSA PERDUE
50 GRAPHICS 0:POKE 752,1
55 POSITION 5,8:PRINT "***  WHERE DO YOU
 LIVE?  ***"
56 FOR PAUSE=1 TO 1000:NEXT PAUSE
60 REM N$ STORES PLAYER'S NAME
65 DIM N$(30),N1$(30),N2$(30)
87 DIM A$(1)
88 DIM GAD$(25),ADN$(25),RAD$(25)
90 DIM RD$(25),AP$(25),CT$(25),ST$(25),Z
P$(25)
91 RESTORE 6001
92 READ RD$:READ AP$:READ CT$:READ ST$:R
EAD ZP$
120 GRAPHICS 0:POKE 752,1
130 POSITION 3,7
140 PRINT "HELLO!"
145 POSITION 3,10
150 PRINT "WHAT IS YOUR"
155 POSITION 3,12:PRINT "FIRST NAME";:IN
PUT N1$
158 POSITION 3,12:PRINT "
          "
```

```
» 159 POSITION 3,12:PRINT "LAST NAME";:INP
  UT N2$
  160 N$=N1$
  161 N$(LEN(N$)+1)=" "
  162 N$(LEN(N$)+1)=N2$
  164 PRINT "↰"
  165 POSITION 8,4
  167 PRINT "HI,THERE, ";N$;"!"
  168 GOSUB 3010:REM * DISPLAY ADDRESS
  170 PRINT "↰"
  172 FOR J=1 TO 5
  173 RESTORE 5000+J:READ ADN$
  175 RESTORE 6000+J:READ RAD$
  180 IF RAD$="" THEN 240
  187 PRINT "↰"
  188 POSITION 8,8
  190 PRINT "WHAT IS YOUR"
  192 POSITION 8,11
  194 PRINT ADN$;:INPUT GAD$
  200 IF GAD$=RAD$ THEN PRINT "↰":GOSUB 1
  010:GOTO 240
  210 COUNT=COUNT+1:PRINT "↰":GOSUB 2010:
  IF COUNT<3 THEN 187
  215 PRINT "↰"
  220 POSITION 8,8:PRINT "YOUR ";ADN$;" IS
  "
  225 POSITION 8,11:PRINT RAD$
  230 FOR PAUSE=1 TO 1500:NEXT PAUSE
  232 COUNT=0
  235 GOTO 187
  240 NEXT J
  250 GOSUB 3010:REM * DISPLAY ADDRESS
  260 PRINT "↰"
  270 POSITION 13,8:PRINT "PLAY AGAIN";:IN
  PUT A$
  280 IF A$="Y" THEN 170
  290 IF A$<>"N" THEN 260
  300 PRINT "↰":POKE 752,0:END
  2150 SOUND 0,0,0,0
  3000 REM *** DISPLAY ADDRESS
  3010 PRINT "↰"
  3020 POSITION 6,6:PRINT "YOUR ADDRESS"
  3021 POSITION 6,7:PRINT "------------"
  3025 POSITION 6,10:PRINT N$
  3030 Y=12
```

```
3035 POSITION 6,Y:PRINT RD$:Y=Y+2
3037 IF AP$="" THEN 3050
3040 POSITION 6,Y:PRINT AP$:Y=Y+2
3050 POSITION 6,Y:PRINT CT$;", ";ST$;" "
;ZP$
3080 FOR PAUSE=1 TO 2000:NEXT PAUSE
3090 RETURN
5001 DATA STREET OR ROAD
5002 DATA APARTMENT
5003 DATA CITY
5004 DATA STATE
5005 DATA ZIP
6001 DATA 2117 CARTER ROAD SW
6002 DATA
6003 DATA ROANOKE
6004 DATA VIRGINIA
6005 DATA 24015
```

Typing Hints ...

➧ To make the **first** name and **last** name messages display in reverse video (see lines 155 and 159), you need to press the Atari-symbol button ( ). This turns the reverse video on. To turn it off, just press the button again.

Highlights ...

The "Where Do You Live" address game was developed by Melissa Perdue, a student at Patrick Henry High School in Roanoke, Virginia.

The address game program has a sample (Roanoke) address entered on lines 6001 to 6005. You will need to replace this with your address. Your street goes on line 6001. Your apartment goes on 6002. If you don't live in an apartment, just leave the empty DATA command at 6002. Your city goes on line 6003. Your state at 6004. Your zip code at 6005.

The address categories are stored in DATA statements on lines 5001 to 5005 (i.e., street or road, apartment, city, state, and zip).

The main action in the program takes place inside the loop from lines 172 to 240. Using RESTORE commands, the program READs in the address categories and the actual address on lines 173 and 175. If a category is empty (for example, your apartment address), the program skips to the next category.

Each time the child gets a wrong answer, the wrong-answer counter, COUNT, is incremented by 1. If COUNT is less than 3, the computer keeps asking for the same part of his or her address. When COUNT reaches 3, the computer displays the correct answer, then erases it and lets the child try again.

At the beginning and end of the game, the program calls the subroutine beginning on line 3000. This subroutine displays the child's name and address in the format in which it would appear on an envelope.

Variables ...

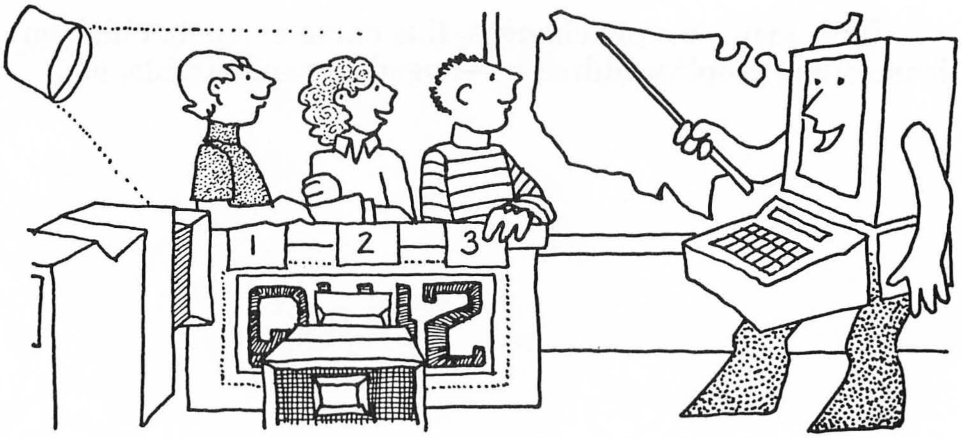
PAUSE	Delay-loop counter.
N$	Your name.
N1$	First Name.
N2$	Last Name.
A$	Answer to question "PLAY AGAIN?"
GAD$	Address guess.
ADN$	Address category (street, city, etc.).
RAD$	Address (your street name, city name, etc.).
RD$	Your road.
AP$	Your apartment.
CT$	Your city.
ST$	Your state.
ZP$	Your zip code.
J	Loop counter—main loop of game.

COUNT	Wrong-answer counter.
Y	Row where part of address is displayed.

Do-It-Yourself ...

With just a couple changes, this game can help children learn other people's addresses—friends, grandparents, etc.

17 QUIZ SHOW



For Parents and Teachers ...

This game helps children learn some interesting facts. The questions in this game can be easily replaced by questions focusing on a particular subject: history, geography, current events, language arts, etc. Also, many new questions can be added.

The game is self grading. It keeps track of the children's correct and incorrect answers. The children's score is printed at the end of the quiz.

For Kids ...

Pretend that you get a letter in the mail. You have been invited to be on a TV quiz show!

You go to the library, and take out dozens of books. You read the newspaper every day. You study your encyclopedia.

You feel so full of facts and figures that they might soon pop out of your ears!

The big day arrives. You go to the TV studio. The show begins. They put makeup on your face, so you don't look pale under the cameras. The stage is hot. The lights are bright. You feel nervous. Here comes the first question: Who is the father of our country? George Washington? John Adams? Thomas Jefferson?

For a moment, you throw a blank. None of the names sounds right. Then you recover. "George Washington?" you say.

The quiz show announcer's happy face appears. "Right!" he says. Happy music floods the studio. The other kids look at you enviously. How could you be so sharp, so cool under pressure?

You relax. This is going to be easy.

The Game ...

Program Name: **QUIZ**

```
50 REM REM *** QUIZ SHOW
55 REM DESIGNED BY SCOTT RAINEY AND
    BRIAN FRANCOIS
60 POKE 752,1:PRINT " \ "
65 POSITION 10,8:PRINT "***  QUIZ SHOW
***"
67 FOR PAUSE=1 TO 1000:NEXT PAUSE
90 DIM A$(1)
110 POKE 82,8
120 ? CHR$(125):POSITION 8,6
130 ? "WHO IS THE FATHER OF OUR
    COUNTRY?":? ""
140 PRINT "A) GEORGE WASHINGTON
   B) JOHN ADAMS                    C) TH
OMAS JEFFERSON"
150 INPUT A$
151 IF A$="A" THEN RC=RC+1:PRINT " \ ":GO
SUB 1010:GOTO 155
152 WC=WC+1:PRINT " \ ":GOSUB 2010
155 ? CHR$(125):POSITION 8,6
```

```
160 ? "WHAT IS THE CAPITAL OF THE
     UNITED STATES?":? ""
170 ? "A) NEW YORK                  B
) WASHINGTON D.C.           C) PHILAD
ELPHIA"
180 INPUT A$
181 IF A$="B" THEN RC=RC+1:POKE 752,1:PR
INT " ↰ ":GOSUB 1010:GOTO 185
182 WC=WC+1:PRINT " ↰ ":GOSUB 2010
185 ? CHR$(125):POSITION 8,6
190 ? "WHO IS THE CURRENT PRESIDENT
     OF THE UNITED STATES?":? ""
200 ? "A) JIMMY CARTER              B
) RICHARD NIXON             C) RONALD
 REAGAN"
210 INPUT A$
211 IF A$="C" THEN RC=RC+1:POKE 752,1:PR
INT " ↰ ":GOSUB 1010:GOTO 215
212 WC=WC+1:PRINT " ↰ ":GOSUB 2010
215 ? CHR$(125):POSITION 8,6
220 ? "WHICH COUNTRY GIVES ITS      P
EOPLE THE MOST FREEDOM?":? ""
230 ? "A) RUSSIA                    B
) POLAND                    C) UNITED
 STATES"
240 INPUT A$
241 IF A$="C" THEN RC=RC+1:POKE 752,1:PR
INT " ↰ ":GOSUB 1010:GOTO 245
242 WC=WC+1:PRINT " ↰ ":GOSUB 2010
245 ? CHR$(125):POSITION 8,6
250 ? "WHICH COUNTRY WAS THE FIRST
     TO PUT A MAN IN SPACE?":? ""
260 ? "A) RUSSIA                    B
) UNITED STATES             C) CHINA"

270 INPUT A$
271 IF A$="B" THEN RC=RC+1:POKE 752,1:PR
INT " ↰ ":GOSUB 1010:GOTO 275
272 WC=WC+1:PRINT " ↰ ":GOSUB 2010
275 ? CHR$(125):POSITION 8,6
280 ? "WHO DISCOVERED ELECTRICITY?":? ""

290 ? "A) BENJAMIN FRANKLIN         B
) ISAAC NEWTON              C) THOMAS
 JEFFERSON"
```

```
300 INPUT A$
301 IF A$="A" THEN RC=RC+1:POKE 752,1:PR
INT " \ ":GOSUB 1010:GOTO 305
302 WC=WC+1:PRINT " \ ":GOSUB 2010
305 ? CHR$(125):POSITION 8,6
310 ? "WHO DISCOVERED AMERICA IN
     1492?":? ""
320 ? "A) AMERIGO VESPUCCI              B
) VIKINGS                       C) CHRIST
OPHER COLUMBUS"
330 INPUT A$
331 IF A$="C" THEN RC=RC+1:POKE 752,1:PR
INT " \ ":GOSUB 1010:GOTO 335
332 WC=WC+1:PRINT " \ ":GOSUB 2010
335 ? CHR$(125):POSITION 8,6
340 ? "WHAT COUNTRY WAS THE FIRST
     TO PUT A MAN ON THE MOON?":? ""
350 ? "A) RUSSIA                        B
) UNITED STATES                 C) BRITAI
N"
360 INPUT A$
361 IF A$="B" THEN RC=RC+1:POKE 752,1:PR
INT " \ ":GOSUB 1010:GOTO 365
362 WC=WC+1:PRINT " \ ":GOSUB 2010
365 ? CHR$(125):POSITION 8,6
370 ? "WHAT WAS THE FIRST COUNTRY
     TO ENTER THE NUCLEAR AGE?":? ""
380 ? "A) UNITED STATES                 B
) CHINA                         C) RUSSIA
"
390 INPUT A$
391 IF A$="A" THEN RC=RC+1:POKE 752,1:PR
INT " \ ":GOSUB 1010:GOTO 395
392 WC=WC+1:PRINT " \ ":GOSUB 2010
395 ? CHR$(125):POSITION 8,6
400 ? "WHAT WAR WON THE UNITED STATES
     HER FREEDOM?":? ""
410 ? "A) WAR OF 1812                   B
) REVOLUTION                    C) CIVIL
WAR"
420 INPUT A$
421 IF A$="B" THEN RC=RC+1:POKE 752,1:PR
INT " \ ":GOSUB 1010:GOTO 425
422 WC=WC+1:PRINT " \ ":GOSUB 2010
425 ? CHR$(125):POSITION 8,6
```

```
430 ? "WHO INVENTED THE TELEPHONE?":? ""

440 ? "A) THOMAS EDISON                B
) ALEXANDER GRAHAM BELL          C) BENJAM
IN FRANKLIN"
450 INPUT A$
451 IF A$="B" THEN RC=RC+1:POKE 752,1:PR
INT " \ ":GOSUB 1010:GOTO 455
452 WC=WC+1:PRINT " \ ":GOSUB 2010
455 ? CHR$(125):POSITION 8,6
460 ? "WHEN IS THE AMERICAN
     INDEPENDENCE DAY?":? ""
470 ? "A) NOVEMBER 25                  B
) OCTOBER 31                     C) JULY 4
"
480 INPUT A$
481 IF A$="C" THEN RC=RC+1:POKE 752,1:PR
INT " \ ":GOSUB 1010:GOTO 485
482 WC=WC+1:PRINT " \ ":GOSUB 2010
485 ? CHR$(125):POSITION 8,6
490 ? "WHO WAS THE PRESIDENT OF THE    U
NITED STATES AT THE BEGINNING  OF THE CI
VIL WAR?":? ""
500 ? "A) ABRAHAM LINCOLN              B
) ANDREW JACKSON                 C) U.S. G
RANT"
510 INPUT A$
511 IF A$="A" THEN RC=RC+1:POKE 752,1:PR
INT " \ ":GOSUB 1010:GOTO 515
512 WC=WC+1:PRINT " \ ":GOSUB 2010
515 ? CHR$(125):POSITION 8,6
520 ? "WHICH OF THE FOLLOWING IS A
     NUMERIC VARIABLE?":? ""
530 ? "A) AX$                          B
) Z                              C) 45"
540 INPUT A$
541 IF A$="B" THEN RC=RC+1:POKE 752,1:PR
INT " \ ":GOSUB 1010:GOTO 545
542 WC=WC+1:PRINT " \ ":GOSUB 2010
545 ? CHR$(125):POSITION 8,6
550 ? "WHAT TYPE OF COMPUTER
     IS THIS?":? ""
560 ? "A) APPLE                        B
) ATARI                          C) TIMEX"
```

```
570 INPUT A$
571 IF A$="B" THEN RC=RC+1:POKE 752,1:PR
INT " ↰ ":GOSUB 1010:GOTO 575
572 WC=WC+1:PRINT " ↰ ":GOSUB 2010
575 ? CHR$(125):POSITION 8,6
580 ? "YOUR SCORE IS:"
590 PRINT :PRINT
600 PRINT "    ";RC;:POKE 85,15:PRINT "RI
GHT"
610 PRINT "    ";WC;:POKE 85,15:PRINT "WR
ONG"
620 PRINT :PRINT
630 IF RC=15 THEN PRINT "YOU'RE A GENIUS
!":GOTO 700
640 IF RC>=11 THEN PRINT "YOU ARE PRETTY
 SMART!":GOTO 700
650 IF RC>=8 THEN PRINT "YOU DID OKAY.":
GOTO 700
660 IF RC>=4 THEN PRINT "YOU NEED A LITT
LE WORK.":GOTO 700
670 IF RC>=1 THEN PRINT "YOU NEED TO STU
DY MORE!":GOTO 700
680 PRINT "OH! OH!  OH!!"
700 FOR PAUSE=1 TO 2000:NEXT PAUSE
710 POKE 82,2:PRINT " ↰ ":POKE 752,0:END
```

Background ...

The Quiz Show game was designed and programmed by Scott Rainey and Brian François, students at Patrick Henry High School in Roanoke, Virginia.

The boys had only two weeks to put this program together. Also, they had to use a faulty program recorder. At the end of each day they would save the current version of their program. The next morning they would try to load the program back into the computer. But the program had disappeared. The recorder had “eaten” it!

In order to complete the program on time, the boys had to get permission to skip some of their classes at school. They finally completed the program on the day it was due.

They write: "In order to complete this program, we had to overcome many obstacles. We hope that your child will learn and benefit from our work."

Highlights ...

On line 110 there is a POKE 82,8 command to center the questions on the TV screen. The left margin of the screen is stored in location 82.

The PRINT CHR$(125) at the beginning of each question is the same as a PRINT " ↰ ". It causes the computer to clear the screen.

There are two counters. The RC counter keeps track of correct answers. The WC counter keeps track of the wrong answers. At the end of the quiz, your score is printed along with a rating. The rating goes from "YOU'RE A GENIUS!" (for all 15 questions correct) to "OH! OH! OH!" (for all 15 questions wrong).

Variables ...

PAUSE	Delay-loop counter.
A$	Accepts your one-letter answer to quiz show questions.
RC	Counter for right answers.
WC	Counter for wrong answers.

Do-It-Yourself ...

These questions are just a sample of the types of questions you can think up for the quiz show game. You can set up a quiz show based on your favorite facts or trivia. Do you like movies? Sports? Astronomy? Adventure games? Mysteries? Rock music? You can set up a quiz show on any of these subjects, or on anything you want.

HAND-EYE

18
HOW FAST ARE YOU?

For Parents and Teachers ...

This game helps children develop hand-eye coordination and concentration.

The computer draws a line from left to right across the screen. As soon as the child sees the line appear on the lefthand side of the screen, he or she presses the **SPACE** bar. When the **SPACE** bar is pressed, the line stops. The faster the child reacts, the better the score.

For Kids ...

How fast are you?

To find out, step right up and have a duel with the computer.

The computer will print a gold block on the TV screen. It will tell you "GET READY!" Then a little black line will shoot across the screen, from left to right. As soon as you see the line appear on the left side of the screen, press the **SPACE** bar. The line will stop, and the computer will tell you how fast you are.

The slower you are, the longer the line on the screen. The faster you are, the shorter the line.

The computer will also tell you how much time passed between the time when the line first appeared and when you pressed the **SPACE** bar and stopped the line.

Keep playing the game to see if you can improve your time.

If the game gets simpler, change the rules. If you are righthanded, start pressing the **SPACE** bar with your left hand. If you are lefthanded, start using your right hand. Or use your nose. Or your elbow. Or turn around and use a mirror.

Let your whole family try the game. Have a family championship. Maybe you can even teach your dog or cat how to play!

The Game . . .

Program Name: **FAST**

```
50 REM *** HOW FAST ARE YOU?
60 GRAPHICS 0:POKE 752,1
65 POSITION 6,8:PRINT "***  HOW FAST ARE
 YOU?  ***"
67 FOR PAUSE=1 TO 1000:NEXT PAUSE
70 DIM A$(35)
80 DIM B$(22)
90 B$="                      "
100 A$=""
102 GRAPHICS 7
105 POKE 752,1
110 PLOT 159,40:DRAWTO 159,20:DRAWTO 0,2
0:POSITION 0,40:POKE 765,1:XIO 18,#6,0,0
,"S:"
140 POKE 656,1:POKE 657,9:PRINT "***  GE
T READY!!  ***"
150 OPEN #1,4,0,"K:"
200 FOR PAUSE=1 TO 1000:NEXT PAUSE
205 COLOR 0
```

```
210 FOR X=0 TO 159:IF PEEK(764)=255 THEN
 PLOT X,30:NEXT X
220 CLOSE #1
230 POKE 656,1:POKE 657,8:PRINT B$
231 IF X=0 THEN POKE 656,1:POKE 657,10:P
RINT "TOO SOON!  TRY AGAIN!":GOTO 250
232 RESTORE 5000+INT(X/10):READ A$
235 GOSUB 3010:REM * CENTER LINE
240 POKE 656,1:POKE 657,INT((40-LEN(A$))
/2):PRINT A$
250 FOR PAUSE=1 TO 1000:NEXT PAUSE
255 POKE 764,255
260 GOTO 100
3000 REM *** ADD TIME TO MESSAGE
3010 A$(LEN(A$)+1)="  ("
3020 A$(LEN(A$)+1)=STR$(X/100)
3030 A$(LEN(A$)+1)=" SECONDS)"
3040 RETURN
5000 DATA ACE!!
5001 DATA PRO!
5002 DATA VERY FAST!
5003 DATA FAST
5004 DATA QUICK
5005 DATA AVERAGE
5006 DATA OKAY
5007 DATA SLOW
5008 DATA VERY SLOW
5009 DATA SLOW AS A TURTLE
5010 DATA SLOW AS A SNAIL
5011 DATA SLOW AS A SLOTH
5012 DATA WAKE UP!
5013 DATA WAKE UP!
5014 DATA YOU NEED HELP!
5015 DATA YOU NEED HELP!
5016 DATA IS ANYBODY THERE?
```

Highlights ...

This game uses Graphics Mode 7. The commands on line 110 draw the gold playing field on the TV screen. The DRAWTO and PLOT commands draw the edges of the field.

The XIO command at the end colors the playing field gold.

The POKE commands on line 140 set the text-area line (POKE 656,1) and the column position (POKE 657,9) for the "GET READY" message.

The PLOT command on line 210 draws the computer's line that sprints across the screen. The computer checks for keyboard input before plotting each point on the line.

If you press the **SPACE** bar before the computer starts drawing the sprint line, it tells you (on line 231): "TOO SOON! TRY AGAIN!" Then it starts all over.

The current column of the computer's sprinting line is represented by X. On line 232 the computer uses X (divided by 10) to point it to the right message to display on the screen. Line 240 centers the message and then prints it. The subroutine on line 3000 adds the child's time onto the message.

The STR$ function on line 3020 converts the value of X/100 into a string, so that it can be added to the message from the computer. X/100 represents the time in seconds that it took for you to press the **SPACE** bar.

Variables ...

PAUSE	Delay-loop counter.
A$	Message—computer tells you how well you did.
B$	Erases text-area messages (filled with blanks).
X	Current column position of the computer's sprint line.

Do-It-Yourself ...

The program always starts the race after it counts to 1000 on line 200. This makes it possible to anticipate the computer and get a better score than would otherwise be possible. To change this, you can put in an RND function and have the computer start the race randomly and unpredictably.

Also, it might make the race more interesting if you plotted a little figure (a horse, person, hare, or tortoise) to race across the golden track. You can move the figure by plotting it and erasing it at different positions across the screen.

You can also make the game more challenging by randomly starting the sprinting line (or figure) on the right or left side of the screen.

FOREIGN LANGUAGE

19
UNO! DOS! TRES!

For Parents and Teachers . . .

This game helps children learn how to count from zero to 10 in Spanish.

For Kids . . .

When you RUN this program, the screen turns blank. At the bottom, the computer asks "NUMBER (0-10)?" You can type any number from **0** to **10**. Let's say you type a **5**. The computer puts a 5 on the screen and underneath it the word for 5 in Spanish:

```
        5
      CINCO
```

When the computer prints the number, it plays a special musical tone. The computer asks you to say the name of the number, first in English and then in Spanish.

If you keep practicing the numbers in Spanish you will soon be able to say them all without any help from the computer:
"SERO! UNO! DOS! TRES! CUATRO! CINCO! SEIS! SIETE! OCHO! NUEVE! DIEZ!"

The Game ...

Program Name: **SPANISH**

```
50 REM *** COUNT IN SPANISH
60 PRINT " ↖":POKE 752,1
65 POSITION 3,8:PRINT "***  COUNT TO 10
IN SPANISH  ***"
67 FOR PAUSE=1 TO 1000:NEXT PAUSE
70 DIM PAD(5):PAD(2)=3:PAD(3)=2:PAD(4)=2
:PAD(5)=1
75 DIM B$(35)
76 B$="
  "
80 DIM M(10)
82 FOR I=0 TO 10
84 READ N
86 M(I)=N
88 NEXT I
90 DIM N$(6)
95 DIM O$(2)
97 DIM A$(1)
100 GRAPHICS 2:POKE 752,1
105 TRAP 100
110 FOR I=1 TO 11
115 POKE 656,0:PRINT B$
120 POKE 656,0:PRINT CHR$(125):PRINT "
        NUMBER (0-10)";:INPUT K
130 IF K<0 OR K>10 THEN 120
135 RESTORE 5010
140 FOR J=0 TO K
145 READ N$
```

```
   150 IF J=K THEN POP :GOTO 165
   160 NEXT J
➧  165 POKE 656,0:PRINT "  SAY IN ENGLISH
    AND  SPANISH"
   170 POSITION 9,3:PRINT #6;J
   175 GOSUB 3010:REM * CENTER NUMBER
   180 POSITION 6,6:PRINT #6;O$;N$
   190 SOUND 0,M(J),10,15
   200 FOR PAUSE=1 TO 800:NEXT PAUSE
   210 SOUND 0,0,0,0
   220 POSITION 9,3:PRINT #6;"   "
   230 POSITION 6,6:PRINT #6;"         "
   240 NEXT I
   250 GRAPHICS 0:POKE 752,1
   255 POSITION 15,8:PRINT "AGAIN";:INPUT A
   $
   257 IF A$="Y" THEN 100
   260 IF A$<>"N" THEN 250
   270 GRAPHICS 0:POKE 752,1:END
   3000 REM *** CENTER NUMBER SUBROUTINE
   3010 O$=""
   3020 IF LEN(N$)+LEN(O$)>5 THEN 3050
   3025 FOR P=1 TO PAD(LEN(N$))
   3030 O$(LEN(O$)+1)=" "
   3040 NEXT P
   3050 RETURN
   5000 DATA 121,108,96,91,81,72,64,60,53,4
   7,45
   5010 DATA SERO,UNO,DOS,TRES,CUATRO,CINCO
   ,SEIS,SIETE,OCHO,NUEVE,DIEZ
```

Typing Hints ...

➧ To get the reverse video for the words English and Spanish on line 165, you need to press the Atari-symbol button (⏃). To turn off the reverse video, press the button again.

Highlights ...

The musical tones that accompany each number are in the DATA command on line 5000. The Spanish names for the numbers 0 through 10 are in the DATA command on line 5010. The musical tones are read into the M array on lines 82 to 88.

The program lets you choose any 11 numbers from 0 to 10. You can count from 0 to 10. Or you can repeat the number 5 (CINCO) 11 times. The major program loop begins on line 110 and ends on line 240.

The program INPUTs your chosen number into the variable K. Then, using a RESTORE 5010 command and a READ command (lines 135 and 145), the program loops through the Spanish words K+1 times and fetches the Spanish name for the number you have chosen.

The Spanish names are different lengths. The subroutine beginning on line 3000 pads the smaller names with blanks to center them (roughly) under the number.

The TRAP command is used on line 105. If you type something other than a number in response to the question on line 120, the TRAP command takes over. It causes the computer to jump to line 100, initialize the screen, and ask you the same question over again. When the computer does this, it executes the TRAP command again. This readies the computer for any new errors.

Variables ...

PAUSE	Delay-loop counter.
PAD	Array—stores number of spaces to pad the beginning of the smaller Spanish number names.
B$	Blanks—empty spaces to erase old numbers and names.
M	Array—stores musical tones for each Spanish number name.
I	Loop counter.
N$	Stores the Spanish name for the number.
O$	Stores the padding spaces that precede the Spanish name for the number.

A$	Stores your answer to "PLAY AGAIN?"
J	Loop counter.
K	The number you choose.
P	Loop counter for subroutine (beginning on line 3000).

Do-It-Yourself . . .

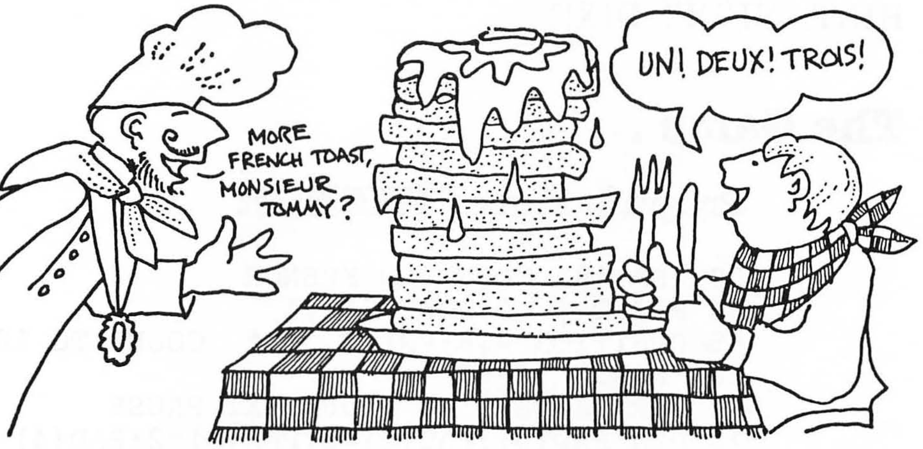
The program only accepts numbers (0, 1, 2, etc.) to indicate which Spanish number name you want to practice. You might modify the program to accept English number names—"zero," "one," "two," and so on. Or, you could have the program accept Spanish number names—"uno," "dos," "tres," etc. This would help the children learn how to spell the names.

A neat feature for the program to have would be automatic counting. At the beginning and end of the game, the program could count from 0 to 10 in Spanish. It would display all the numbers and play all the musical tones.

Also, the program could be expanded to Spanish words and Spanish phrases. It could also display the phonetic pronunciation of each of the numbers, words, and phrases to help the children say the words correctly.

20

UN! DEUX! TROIS!



For Parents and Teachers ...

This game helps children learn how to count from zero to 10 in French.

For Kids ...

When you RUN this program, the screen turns blank. At the bottom, the computer asks "NUMBER (0-10)?" You can type any number from zero to 10. Let's say you type a 5. The computer puts a 5 on the screen and underneath it the word for 5 in French:

```
      5
    CINQ
```

When the computer prints the number, it plays a special musical tone. The computer asks you to say the name of the number, first in English and then in French.

If you keep practicing the numbers in French you will soon be able to say them all without any help from the computer: "NUL! UN! DEUX! TROIS! QUATRE! CINQ! SIX! SEPT! HUIT! NEUF! DIX!"

The Game ...

Program Name: **FRENCH**

```
50 REM *** COUNT IN FRENCH
60 PRINT "\":POKE 752,1
65 POSITION 4,8:PRINT "***  COUNT TO 10
IN FRENCH  ***"
67 FOR PAUSE=1 TO 1000:NEXT PAUSE
70 DIM PAD(5):PAD(2)=3:PAD(3)=2:PAD(4)=2
:PAD(5)=1
75 DIM B$(35)
76 B$="
  "
80 DIM M(10)
82 FOR I=0 TO 10
84 READ N
86 M(I)=N
88 NEXT I
90 DIM N$(6)
95 DIM O$(3)
97 DIM A$(1)
100 GRAPHICS 2:POKE 752,1
105 TRAP 100
110 FOR I=1 TO 11
115 POKE 656,0:PRINT B$
120 POKE 656,0:PRINT CHR$(125):PRINT "
        NUMBER (0-10)";:INPUT K
130 IF K<0 OR K>10 THEN 120
135 RESTORE 5010
140 FOR J=0 TO K
145 READ N$
150 IF J=K THEN POP :GOTO 165
160 NEXT J
```

```
» 165 POKE 656,0:PRINT "  SAY IN ENGLISH
   AND FRENCH"
  170 POSITION 9,3:PRINT #6;J
  175 GOSUB 3010:REM * CENTER NUMBER
  180 POSITION 6,6:PRINT #6;O$;N$
  190 SOUND 0,M(J),10,15
  200 FOR PAUSE=1 TO 800:NEXT PAUSE
  210 SOUND 0,0,0,0
  220 POSITION 9,3:PRINT #6;"  "
  230 POSITION 6,6:PRINT #6;"      "
  240 NEXT I
  250 GRAPHICS 0:POKE 752,1
  255 POSITION 15,8:PRINT "AGAIN";:INPUT A
  $
  257 IF A$="Y" THEN 100
  260 IF A$<>"N" THEN 250
  270 GRAPHICS 0:POKE 752,1:END
  3000 REM *** CENTER NUMBER SUBROUTINE
  3010 O$=""
  3020 IF LEN(N$)+LEN(O$)>5 THEN 3050
  3025 FOR P=1 TO PAD(LEN(N$))
  3030 O$(LEN(O$)+1)=" "
  3040 NEXT P
  3045 IF N$="DIX" THEN O$(LEN(O$)+1)=" "
  3050 RETURN
  5000 DATA 121,108,96,91,81,72,64,60,53,4
  7,45
  5010 DATA NUL,UN,DEUX,TROIS,QUATRE,CINQ,
  SIX,SEPT,HUIT,NEUF,DIX
```

Typing Hints ...

» To get the reverse video for the words English and French on line 165, you need to press the Atari-symbol button ( ). To turn off the reverse video, press the button again.

Highlights ...

The musical tones that accompany each number are in the DATA command on line 5000. The French names for the numbers 0 through 10 are in the DATA command on line 5010. The musical tones are read into the M array on lines 82 to 88.

The program lets you choose any 11 numbers from 0 to 10. You can count from 0 to 10. Or you can repeat the number 5 (CINQ) 11 times. The major program loop begins on line 110 and ends on line 240.

The program INPUTs your chosen number into the variable K. Then, using a RESTORE 5010 command and a READ command (lines 135 and 145), the program loops through the French words K+1 times and fetches the French name for the number you have chosen.

The French names are different lengths. The subroutine beginning on line 3000 pads the smaller names with blanks to center them (roughly) under the number.

The TRAP command is used on line 105. If you type something other than a number in response to the question on line 120, the TRAP command takes over. It causes the computer to jump to line 100, initialize the screen, and ask you the same question over again. When the computer does this, it executes the TRAP command again. This readies the computer for any new errors.

Variables . . .

PAUSE	Delay-loop counter.
PAD	Array—stores number of spaces to pad the beginning of the smaller French number names.
B$	Blanks—empty spaces to erase old numbers and names.
M	Array—stores musical tones for each French number name.
I	Loop counter.
N$	Stores the French name for the number.
O$	Stores the padding spaces that precede the French name for the number.

A$	Stores your answer to "PLAY AGAIN?"
J	Loop counter.
K	The number you choose.
P	Loop counter for subroutine (beginning on line 3000).

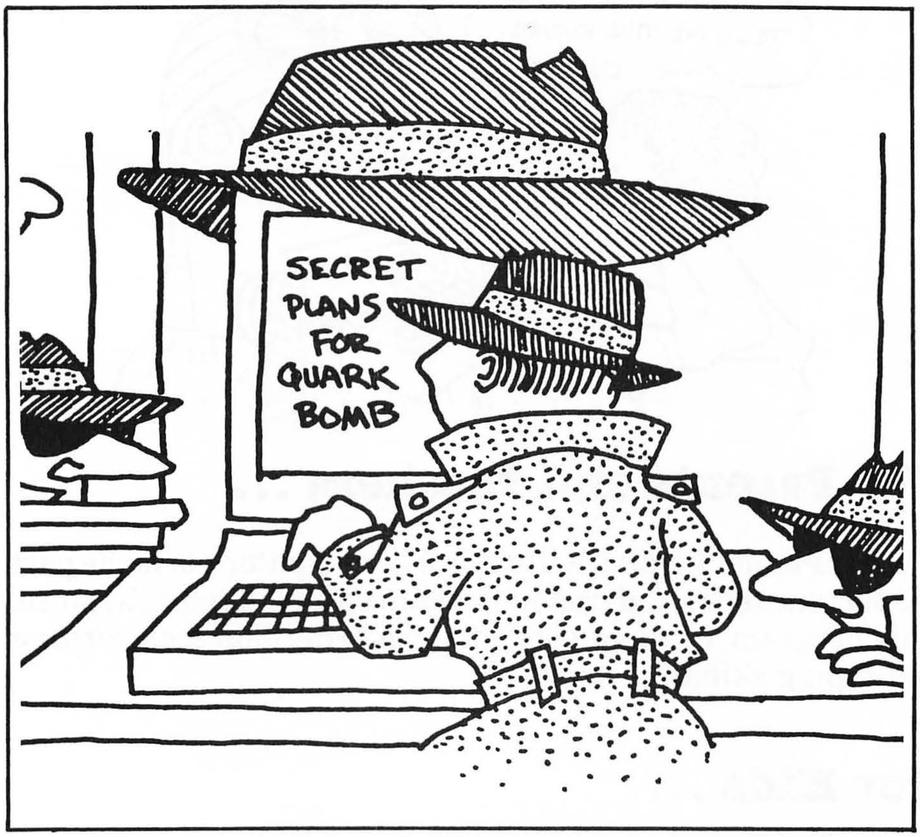
Do-It-Yourself . . .

The program only accepts numbers (0, 1, 2, etc.) to indicate which French number name you want to practice. You might modify the program to accept English number names—"zero," "one," "two," and so on. Or, you can have the program display only the English name and the digit. Then the child has to type in the correct French name—"un," "deux," "trois," etc.

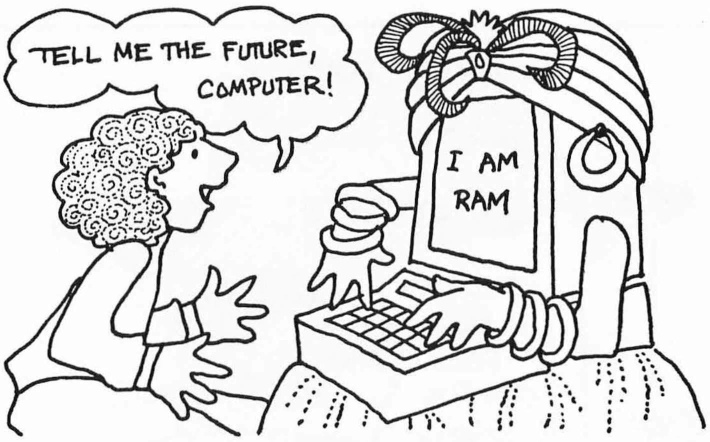
It would also be neat if the program could automatically count and display the numbers. At the beginning and end of the game, the program would count from zero to 10 in French. It would display all the numbers and play all the musical tones.

Also, the program could be expanded to French words and French phrases. It could also display the phonetic pronunciation of each of the numbers, words, and phrases to help the chidren say the words correctly.

IMAGINATION



21 THE FORTUNE TELLER



For Parents and Teachers ...

This is an imagination game. The computer plays the part of a fortune teller and pretends it can see into the child's future. This program will motivate children to develop their writing and typing skills.

For Kids ...

Imagine that it's late at night and you are asleep.

A crash of thunder wakes you. You hear eerie music coming up the stairs. You sneak downstairs to see what is making the music.

It's the computer. Strange messages appear on the computer's TV screen:

```
I AM RAM. I AM A WIZARD.
I AM 10,000 YEARS OLD.
I LIVE INSIDE YOUR COMPUTER.
```

This isn't real. You pinch yourself to see if you are dreaming. Ouch! Nope.

New messages appear:

```
WHEN YOU TURN ON YOUR COMPUTER,
I WAKE UP.
THEN I CAN SEE THE FUTURE.
I CAN SEE YOUR FUTURE!
```

All of a sudden, you start to shiver.

The messages continue:

```
ASK ME ANY QUESTION WITH A
YES OR NO ANSWER.
WHAT IS YOUR QUESTION?
```

Wow! You could use this computer wizard to predict the outcome of elections, or bet on horse races and make a million dollars. Or you could become a superstar reporter and find out the news even before it happens!

What should be your first question? You have it! You type madly.

The wizard flashes a new message:

```
I AM GAZING AT MY CRYSTAL BALL ...
                   GAZING ...
                   GAZING ...
                   GAZING ...
I SEE YOUR FUTURE!
```

YOUR QUESTION:

WILL I EVER SCORE A MILLION POINTS
IN THE NEW ARCADE GAME
MEATBALL WARS?

THE ANSWER IS ...

The wizard pauses for dramatic suspense.

You press your nose against the glass of the TV screen. You are so close the words fuzz up. Your eyes start to cross. "What's the answer?" you cry.

The wizard finally answers:

YES!!

"Hurray!" you cheer.

You run off to tell the other members of your family about the fortune teller who lives inside your computer.

The Game ...

Program Name: **FORTUNE**

```
50 REM *** FORTUNE TELLER
60 POKE 752,1:PRINT "↖"
65 POSITION 5,8:PRINT "***  THE FORTUNE
TELLER  ***"
66 FOR PAUSE=1 TO 1000:NEXT PAUSE
70 DIM A$(1)
80 DIM N$(15)
90 DIM M$(110)
100 PRINT "↖"
110 PRINT "I AM RAM.  I AM A WIZARD."
120 PRINT
130 PRINT "I AM 10,000 YEARS OLD."
140 PRINT
150 PRINT "I LIVE INSIDE YOUR COMPUTER."

160 FOR PAUSE=1 TO 1500:NEXT PAUSE
170 PRINT "↖"
```

```
   180 PRINT "WHEN YOU TURN ON YOUR COMPUTE
   R,"
   185 PRINT
   190 PRINT "I WAKE UP."
   200 PRINT
   210 PRINT "THEN I CAN SEE THE FUTURE."
   220 PRINT
➧ 230 PRINT "I CAN SEE YOUR FUTURE!"
   235 PRINT "           ----"
   240 FOR PAUSE=1 TO 1500:NEXT PAUSE
   250 PRINT "↰"
   255 M$=""
   260 PRINT "ASK ME ANY QUESTION WITH A"
   265 PRINT
➧ 270 PRINT "YES OR NO ANSWER."
   280 PRINT
   290 PRINT "WHAT IS YOUR QUESTION";:INPUT
    M$
   300 PRINT "↰"
   304 PRINT "I AM GAZING AT MY CRYSTAL BAL
   L ... "
   305 PRINT
   306 PRINT
   307 FOR PAUSE=1 TO 500:NEXT PAUSE
   310 FOR I=1 TO 3
➧ 320 POKE 85,8:PRINT "GAZING ..."
   325 FOR PAUSE=1 TO 200:NEXT PAUSE
   330 PRINT
   340 NEXT I
   345 PRINT
   350 PRINT "I SEE YOUR FUTURE!"
   360 PRINT
   370 FOR PAUSE=1 TO 400:NEXT PAUSE
   372 PRINT
   373 PRINT "YOUR QUESTION:"
   374 PRINT :PRINT M$
   375 PRINT :PRINT
   380 PRINT "THE ANSWER IS ... ";
   390 FOR PAUSE=1 TO 1000:NEXT PAUSE
➧ 400 IF INT(RND(1)*2)+1=1 THEN PRINT "YES
   !!":GOTO 430
➧ 410 PRINT "NO!!"
   430 FOR PAUSE=1 TO 500:NEXT PAUSE
   440 PRINT "↰"
```

```
450 PRINT "WANT ME TO TELL YOUR"
460 PRINT "FUTURE AGAIN";:INPUT A$
470 IF A$="Y" THEN 250
480 IF A$<>"N" THEN 440
490 POKE 752,0:PRINT " ↰ ":END
```

Typing Hints ...

➧ You can get the letters to appear in reverse video on lines 230, 270, 320, 400, and 410 by pressing the Atari-symbol button (◼). To turn the reverse video effect off, press the button again.

Highlights ...

This program is long, but it consists almost entirely of PRINT statements.

The key to the program is on line 400. There the computer swami "flips a coin" to decide your future. The RND function makes the computer choose either a 1 or a 2. If the computer chooses a 1, it answers your question "YES!!" If the computer chooses a 2, it answers your question "NO!!"

What makes this program a success, of course, is obviously not programming knowhow. Instead it is atmosphere and imagination. Sit down and try asking the computer to predict **your** future. Before you know it, you will be asking the computer some pretty serious questions. It's easier than you think to come under the fortune teller's spell.

Variables ...

PAUSE	Delay-loop counter.
A$	Accepts your answer to "TELL YOUR FUTURE AGAIN?"

N$	Accepts child's name.
M$	Accepts child's question.
I	Loop counter—prints "GAZING . . ." three times.

Do-It-Yourself ...

You can add all sorts of bells and whistles to this program to heighten the illusion that a real 10,000-year-old wizard lives inside the computer. For example, the wizard can ask the child his or her name. (The name-variable N$ has already been dimensioned on line 80.) Information in DATA statements can correlate with a particular name and the wizard can impress the child with how much he knows about him or her.

You can also add SOUND commands that make sound effects and eerie noises. You might even consider adding a crystal ball on the screen or a glimpse of the wizard's ancient face.

This program is an example of how a good game can be 90% imagination and only 10% perspiration.

22

SECRET AGENT

For Parents and Teachers ...

This is an imagination game. It takes words and sentences and turns them into secret codes for children to pass around and try to figure out. This game might act as an incentive to encourage your children to practice writing and typing on the computer.

For Kids ...

Pretend you are agent Triple-Nine. You have a secret message that you have to deliver to the president of a small country nestled in the Andes Mountains in South America. The message contains the plans to a powerful, new, top-secret Quark bomb. The president needs this bomb to defend his country against an imminent attack of robot guerillas from a neighboring country.

You are almost ready to board your private jet and fly to the president's country. But first, in case you meet with foul play, you need to translate the bomb plans into a secret code. Then, even if the plans fall into the enemy's hands, they will be useless.

You can invent a secret code. Then you can take each letter of each word in the bomb plans and translate it.

But this would take forever. Anyway, you already have a coding machine. It's your computer.

You turn on the computer, load the Secret-Agent program, and type **RUN**. The computer asks you for your code name. You type **999**. "GOOD NAME!" says the computer. "I LIKE THAT!"

You type in the secret plans, one line at a time. A message flashes on the TV screen: "CODING MACHINE NOW WORKING." Moments later the coded bomb plans appear on the screen. You copy them down and destroy the original plans. You board your airplane and head for South America.

The Game ...

Program Name: **SECRET**

```
50 REM *** SECRET AGENT
60 POKE 752,1:PRINT "↖"
65 POSITION 6,8:PRINT "***  SECRET AGENT
 GAME  ***"
66 FOR PAUSE=1 TO 1000:NEXT PAUSE
70 DIM A$(1)
80 DIM N$(15)
90 DIM M$(110),M2$(110)
100 PRINT "↖"
110 PRINT "WHAT IS YOUR CODE NAME";:INPU
T N$
120 PRINT "↖"
130 PRINT "GOOD NAME!  I LIKE IT!"
140 PRINT
150 PRINT N$;", WHAT IS YOUR SECRET"
155 PRINT "MESSAGE";:INPUT M$
160 PRINT "↖":M2$=""
170 PRINT "CODING MACHINE NOW WORKING"
```

```
180 FOR PAUSE=1 TO 1500:NEXT PAUSE
190 FOR I=1 TO LEN(M$)
195 IF M$(I,I)=" " THEN M2$(I,I)=" ":GOT
O 210
200 M2$(I,I)=CHR$(ASC(M$(I,I))+1)
210 NEXT I
220 PRINT "↰"
230 PRINT "ORIGINAL MESSAGE:"
240 PRINT
250 PRINT M$
260 PRINT :PRINT :PRINT
270 PRINT "CODED MESSAGE:"
280 PRINT
290 PRINT M2$
300 PRINT :PRINT :PRINT
310 PRINT "PLAY AGAIN, ";N$;:INPUT A$
320 IF A$="Y" THEN PRINT "↰":GOTO 150
330 IF A$<>"N" THEN 310
340 POKE 752,0:PRINT "↰":END
```

Typing Hints . . .

» You can make the reverse video message on line 170 by pressing the Atari-symbol button ( ). Then type the message and press the Atari-symbol button once more.

Highlights . . .

This is an extremely simple program. It begins and ends with some PRINT commands. In between is a small loop on lines 190 to 210 that translates your message from English to secret code.

Line 200 is the key line. On line 200 the computer takes a letter or number in your message, converts it to its Atari ASCII code value, then adds one. Next, it converts this number back into some new letter, character, or punctuation symbol. The message in its original form is stored in M$. The new, coded message is stored in M2$.

Variables ...

PAUSE	Delay-loop counter.
A$	Your answer to question "NEW MESSAGE?"
N$	Your secret-agent name.
M$	Original message.
M2$	Coded message.
I	Counter for coding loop.

Do-It-Yourself ...

This game is good at coding secret messages but not at decoding them. You can modify the program to make it into a new, secret-message decoder. To do that you need to change line 200. Instead of adding 1 to the Atari ASCII value of M$(I,I), you have the program subtract 1.

You might also modify the program so it stores your child's secret messages on tape or disk. That way the messages don't have to be copied by hand back into the computer each time the child wants to code or decode them.

5771-7

Atari® in Wonderland

Fred D'Ignazio

A simple, entertaining introduction to the world of the Atari for youngsters. Learn how to write a book report, learn angle measure while "riding" a 3-D roller coaster, create songs, test reflexes, appear on a Quiz Show, and learn to count in French and Spanish. Twenty-two exciting short stories and original programs teach children word and number skills in an easygoing, friendly style that children love.

Contains instructions for using the Atari graphics keys. Suggestions in each chapter explain how to modify the programs to make the adventures even more exciting. The programs are listed in the book and can be easily typed into the Atari. A cassette tape is also available containing all the programs ready to run.

HAYDEN BOOK COMPANY, INC.
Hasbrouck Heights, New Jersey

ISBN 0-8104-5771-7