

COMMODORE 64TM IN Wonderland

23 programs
for learning and fun!
Fred D'Ignazio



HAYDEN



COMMODORE 64™
IN Wonderland



COMMODORE 64™
IN Wonderland

Fred D'Ignazio



HAYDEN BOOK COMPANY
a division of Hayden Publishing 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: LES PURIFICACIÓN
Production Editor: LORI WILLIAMS
Developmental Editor: KAREN PASTUZYN
Conversion by: BARRY GILBERT
Book Design: 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 GRAPHICS CO.:
FAIRFIELD GRAPHICS DIVISION

Library of Congress Cataloging in Publication Data

D'Ignazio, Fred.

Commodore 64 in Wonderland.

1. Commodore 64 (Computer)—Programming—Juvenile literature.
2. Computer games—Juvenile literature. 3. BASIC (Computer program language)

I. Title. II. Title: Commodore sixty-four in Wonderland.

QA76.8.C64D54 1984 001.64'25 84-4681

ISBN 0-8104-6308-3

Commodore 64 is a trademark of Commodore Business Machines, Inc., which is not affiliated with Hayden Book Company.

Copyright © 1984 by Fred D'Ignazio. All rights reserved. No part of this book may be reprinted, or reproduced, or utilized in any form or by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying and recording, or in any information storage and retrieval system, without permission in writing from the Publisher.

Printed in the United States of America

1	2	3	4	5	6	7	8	9	PRINTING
84	85	86	87	88	89	90	91	92	YEAR

PREFACE

This is a book of learning games you can type in on your Commodore 64 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-three 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.

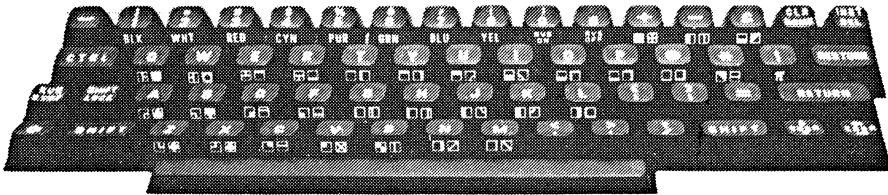
Next is a **For Kids** section that weaves a story around each game and encourages children to use their imaginations when they play the game.

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, and END. References to buttons on the keyboard are written in bold capital letters: for example, CTRL, SHIFT, and RETURN.

Some 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 keyboard is shown here. When you use these special characters, remember to hold down the **SHIFT** or **⌘** key while pressing the selected graphics key.



Note on Program Listings ...

The program listings contain bracketed codes for colors and cursor movements. Refer to the Appendix to identify the keys to press to obtain the bracketed codes.

Thanks to My Young Assistants ...

Young people designed 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 designing 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 original programs for this book. David spent several weeks carrying their one 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, probably due to a defective program recorder.

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:

- A Commodore 64
- A black and white or, preferably, color monitor (TV)
- A Commodore Datassette or a Commodore disk drive

CONTENTS

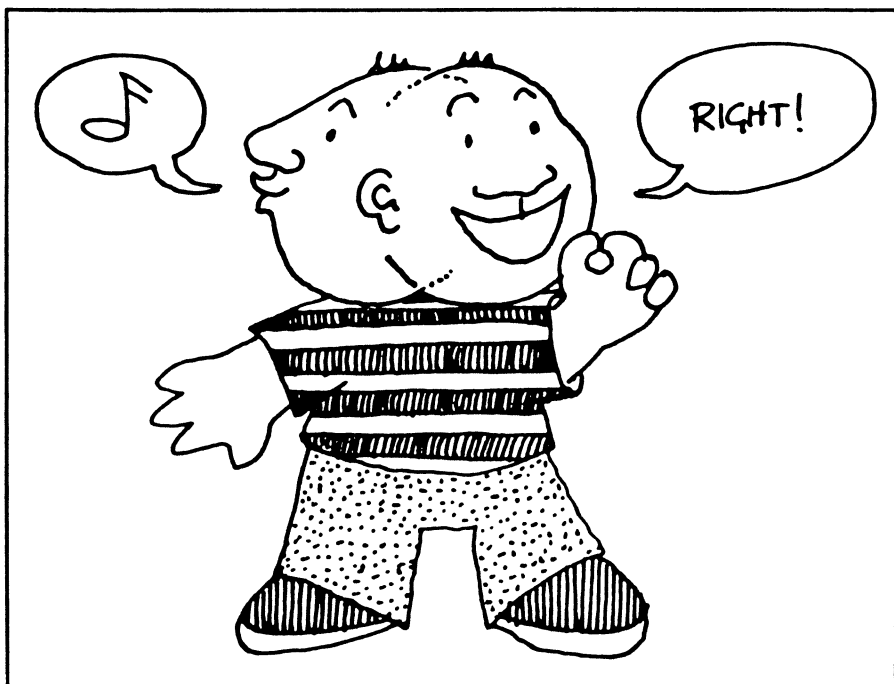
FACES ...	1
1. Happy Face	2
2. Sad Face	7
3. Happy Face and Sad Face	11
WORDS ...	15
4. Scrambled Bees	16
5. Backwords	20
6. Fat Cat	23
7. Book Report	29
NUMBERS ...	37
8. The Arithmetic Game	38
9. Greater than What?	46
10. The Hamburger Contest	54
11. Pepperoni, Please!	61
12. The Roller Coaster	72
13. Sleds!	78
COLORS ...	83
14. The Pied Piper	84
MUSIC ...	91
15. Make Up a Song	92

KNOWLEDGE ...	101
16. Al's Tour of the States	102
17. Where Do You Live?	108
18. Quiz Show	114
HAND-EYE ...	123
19. How Fast Are You?	124
FOREIGN LANGUAGE ...	131
20. Uno! Dos! Tres!	132
21. Un! Deux! Trois!	138
IMAGINATION ...	145
22. The Fortune Teller	146
23. Secret Agent	152
APPENDIX ...	157

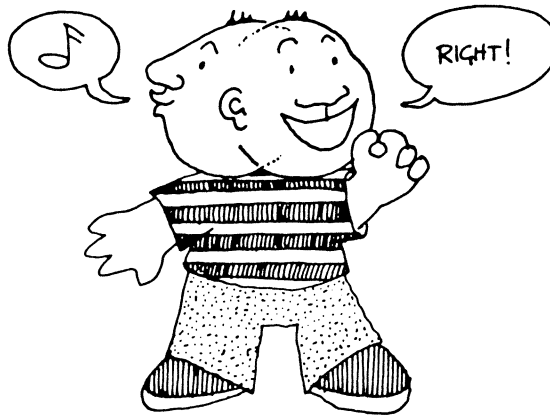
COMMODORE 64™
IN Wonderland



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 AND SOUND
1010 PRINT"{HM}{CD}{CD}{CD}{CD}{CD}{CD}{CD}
      ){CD}{CD}{CD}"
1020 PRINTTAB(17)"      "
1030 PRINTTAB(17)"  /      \"
1040 PRINTTAB(17)" | *      * |"
1050 PRINTTAB(17)" |   *   |"
1060 PRINTTAB(17)" |   v   |"
1070 PRINTTAB(17)" |   ^   |"
1080 PRINTTAB(17)" |   v   |"
1090 PRINTTAB(17)"  \      /"
1100 PRINTTAB(17)"      "
1110 PRINTTAB(17)" RIGHT!"
1120 FOR L=54272 TO 54296:POKEL,0:NEXT
1130 POKE 54296,15 :REM MAX. VOLUME
1140 POKE 54277,92 :REM ATTACK/DECAY
1150 POKE 54278,136:REM SUSTAIN/RELEASE
1160 FOR L=1 TO 9
1170 POKE 54273, 59:POKE 54272,190 :REM VO
ICE 1 - HIGH & LOW FREQUENCY
1180 POKE 54276,17 :REM TRIANGLE WAVEFORM
1190 FOR M=1 TO 30
1200 NEXT M
1210 POKE 54276,16 :REM VOICE 1 OFF
1220 POKE 54273,29:POKE 54272,223 :REM VOI
CE 1 - HIGH & LOW FREQUENCY
1230 POKE 54276,17 :REM TRIANGLE WAVEFORM
1240 FOR M=1 TO 60
1250 NEXT M
1260 NEXT L
1270 POKE 54276,0:REM VOICE 1 OFF
1280 POKE 54296,0:REM MIN. VOLUME
1290 PRINT"{HM}{CD}{CD}{CD}{CD}{CD}{CD}{CD}
      ){CD}{CD}{CD}"
1300 FOR I=1 TO 10
1310 PRINT TAB(17)"      "
1320 NEXT I
1330 RETURN

```

Typing Hints ...




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

Each button produces two graphics characters, depending on whether you use the  or the **SHIFT** key with it.

1. Hold the Commodore () key down and push the key with the character you want to type. For example, hold down  and press the **M** key and you have a |.

2. Hold the **SHIFT** key down and push the button with the character you want to type. For example, hold down **SHIFT** and press **M** and you have a \.

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

In line Number ...	To get ...	Which is (are) ...	Press ...
1020, 1070, 1090	—	top and bottom of face and mouth	 @
1030	/ and \	left and right upper corners of face	SHIFT N and SHIFT M
1040-1080	and	left and right sides of face	 G and  M
1040	♥	eyes	SHIFT S
1050	•	nose	SHIFT Q
1070, 1090	\ and /	left and right corners of mouth and lower left and right corners of face	SHIFT M and SHIFT N

Highlights ...

This is one of the simplest programs in the book. On lines 1020 to 1100, it prints out the happy face. On line 1110, it prints the message "RIGHT!" On lines 1130 to 1280 it makes the whistle sound.

The program uses specific POKE commands to make the whistle. POKE 54296,15 sets the speaker to maximum volume. POKE 54273,59 and POKE 54272,190 control one of the sounds. POKE 54273,29 and POKE 54272,223 control the second sound. Line 1120 clears the Commodore 64 sound registers.

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 L = 1 TO 9 on line 1160. This tells the computer to set the loop counter L to 1. Each time the computer races around the loop it adds one to the loop counter (L). The computer does nine loops (from L = 1 to L = 9). Then it goes on and does something else.

Also, note that the program starts on line 1000 and ends on line 1330 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 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!"

You should continue to the next section and type in the Sad Face subroutine. Follow the directions in the Highlights section of Sad Face.

Variables ...

- L Whistle counter—9 whistles.
- M 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 AND SOUND
2010 PRINT"{HM}{CD}{CD}{CD}{CD}{CD}{CD}{CD}
}{CD}{CD}{CD}"
2020 PRINTTAB(17) " _____ "
2030 PRINTTAB(17) " /         \ "
2040 PRINTTAB(17) "| +     + |"
2050 PRINTTAB(17) "|   ●   |"
2060 PRINTTAB(17) "|         |"
2070 PRINTTAB(17) "|  ^   ^  |"
2080 PRINTTAB(17) "|         |"
2090 PRINTTAB(17) " \_____/ "
2100 PRINTTAB(17) " "
2110 PRINTTAB(11) "SORRY ... TRY AGAIN"
2120 FOR L=54272 TO 54296:POKEL,0:NEXT
2130 POKE 54296,15 :REM VOICE 1 VOLUME
2140 POKE 54277,92 :REM ATTACK/DECAY
2150 POKE 54278,16 :REM SUSTAIN/RELEASE
2160 POKE 54273,5 :POKE 54272,152 :REM VOI
CE 1 - HI & LOW FREQUENCY
2170 POKE 54275,12:POKE54274,15 :REM HI &
LOW PULSE WIDTH
2180 POKE 54276,65 :REM PULSE WAVEFORM
2190 FOR M=1 TO 700
2200 NEXT M
2210 POKE 54276,64 :REM VOICE 1 OFF
2220 POKE 54273,3 :POKE 54272,134:REM VOIC
E 1 - HI & LOW FREQUENCY
2230 POKE 54276,65 :REM VOICE 1 ON
2240 FOR M=1 TO 1500
2250 NEXTM
2260 POKE 54276,0 :REM VOICE 1 OFF
2270 POKE 54296,0 :REM MIN. VOLUME
2280 PRINT"{HM}{CD}{CD}{CD}{CD}{CD}{CD}{CD}
}{CD}{CD}{CD}"
2290 FOR I=1 TO 9
2300 PRINT TAB(17) " "
2310 NEXT I
2320 PRINT TAB(11) " "
2330 RETURN

```

Typing Hints ...

To make the sad face, you need to use graphics characters. The keyboard is pictured in the preface of this book.

In line Number ...	To get ...	Which is (are) ...	Press ...
2020, 2090	—	top and bottom of face	 @
2030, 2070	/ and \	left and right upper corners of face and left and right corners of mouth	SHIFT N and SHIFT M
2040-2080	and	left and right sides of face	 G and  M
2040	+	eyes	+
2050	•	nose	SHIFT Q
2070	—	the mouth	 T
2090	\ and /	left and right lower corners of face	SHIFT M and SHIFT N

Highlights ...

Be sure you have the Happy Face subroutine typed in before you type in the Sad Face subroutine.

The sad face is similar to the happy face. Lines 2020 to 2100 print out the sad face. Line 2110 prints the sad message. Line 2120 clears the Commodore 64 sound registers. Lines 2130 to 2270 play the sad sound. Lines 2280 to 2320 erase the sad face and the message.

To test the Sad Face, just type GOSUB 2010.

Note that the Sad Face starts on line 2000. The Sad Face is a subroutine, just like the Happy Face.

You should continue to the next section and follow the directions in the Highlights section of HAP&SAD.

Variables ...

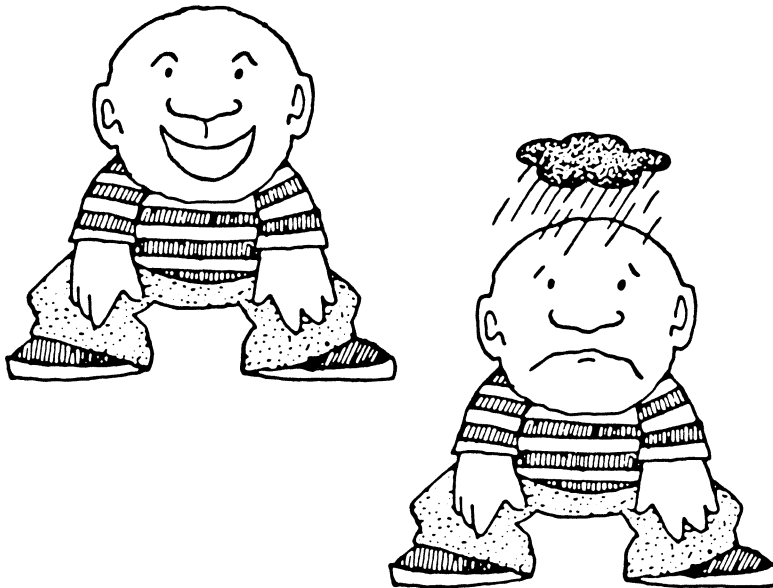
- L Clears the Commodore 64 sound registers.
- M Delay counter—computer pauses between sounds.
- I Row counter—used in erasing face.

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?

3

HAPPY FACE AND SAD FACE



For Parents and Teachers ...

This chapter contains the combined Happy and Sad subroutine which can be attached onto most of the game programs in this book.

The programs that utilize this subroutine do not have these lines in the program listing. Check the Typing Hints section of each game to see if the program uses HAP&SAD.

The Program ...

Program Name: **HAP&SAD**

```

1000 REM HAPPY FACE AND SOUND
1010 PRINT"{HM}{CD}{CD}{CD}{CD}{CD}{CD}{CD}
      ){CD}{CD}{CD}"
1020 PRINTTAB(17)"          "
1030 PRINTTAB(17)"  /      \"
1040 PRINTTAB(17)" | *    * |"
1050 PRINTTAB(17)" |   .   |"
1060 PRINTTAB(17)" |   V   |"
1070 PRINTTAB(17)" |   ^   |"
1080 PRINTTAB(17)" |   V   |"
1090 PRINTTAB(17)" \      /"
1100 PRINTTAB(17)"          "
1110 PRINTTAB(17)" RIGHT!"
1120 FOR L=54272 TO 54296:POKE L,0:NEXT
1130 POKE 54296,15 :REM MAX. VOLUME
1140 POKE 54277,92 :REM ATTACK/DECAY
1150 POKE 54278,136:REM SUSTAIN/RELEASE
1160 FOR L=1 TO 9
1170 POKE 54273, 59:POKE 54272,190 :REM VO
ICE 1 - HIGH & LOW FREQUENCY
1180 POKE 54276,17 :REM TRIANGLE WAVEFORM
1190 FOR M=1 TO 30
1200 NEXT M
1210 POKE 54276,16 :REM VOICE 1 OFF
1220 POKE 54273,29:POKE 54272,223 :REM VOI
CE 1 - HIGH & LOW FREQUENCY
1230 POKE 54276,17 :REM TRIANGLE WAVEFORM
1240 FOR M=1 TO 60
1250 NEXT M
1260 NEXT L
1270 POKE 54276,0:REM VOICE 1 OFF
1280 POKE 54296,0:REM MIN. VOLUME
1290 PRINT"{HM}{CD}{CD}{CD}{CD}{CD}{CD}{CD}
      ){CD}{CD}{CD}"
1300 FOR I=1 TO 10
1310 PRINT TAB(17)"          "
1320 NEXT I
1330 RETURN
2000 REM SAD FACE AND SOUND
2010 PRINT"{HM}{CD}{CD}{CD}{CD}{CD}{CD}{CD}
      ){CD}{CD}{CD}"

```

```

2020 PRINTTAB(17)"  _____  "
2030 PRINTTAB(17)" /         \  "
2040 PRINTTAB(17)"| +       + |"
2050 PRINTTAB(17)"|   ●   |"
2060 PRINTTAB(17)"|         |"
2070 PRINTTAB(17)"|   ^   |"
2080 PRINTTAB(17)"|         |"
2090 PRINTTAB(17)" \         /  "
2100 PRINTTAB(17)"  _____  "
2110 PRINTTAB(11)"SORRY ... TRY AGAIN"
2120 FOR L=54272 TO 54296:POKE L,0:NEXT
2130 POKE 54296,15 :REM VOICE 1 VOLUME
2140 POKE 54277,92 :REM ATTACK/DECAY
2150 POKE 54278,16 :REM SUSTAIN/RELEASE
2160 POKE 54273,5 :POKE 54272,152 :REM VOI
CE 1 - HI & LOW FREQUENCY
2170 POKE 54275,12:POKE 54274,15 :REM HI &
LOW PULSE WIDTH
2180 POKE 54276,65 :REM PULSE WAVEFORM
2190 FOR M=1 TO 700
2200 NEXT M
2210 POKE 54276,64 :REM VOICE 1 OFF
2220 POKE 54273,3 :POKE 54272,134:REM VOIC
E 1 - HI & LOW FREQUENCY
2230 POKE 54276,65 :REM VOICE 1 ON
2240 FOR M=1 TO 1500
2250 NEXT M
2260 POKE 54276,0 :REM VOICE 1 OFF
2270 POKE 54296,0 :REM MIN. VOLUME
2280 PRINT"{HM}{CD}{CD}{CD}{CD}{CD}{CD}{CD}
}{CD}{CD}{CD}"
2290 FOR I=1 TO 9
2300 PRINT TAB(17)"          "
2310 NEXT I
2320 PRINT TAB(11)"          "
2330 RETURN

```

Typing Hints ...

Programs requiring HAP&SAD can be identified by GOSUB 1010 and GOSUB 2010 commands.

Save this combined subroutine on disk by SAVE "HAP&SAD",8 or on tape by SAVE "HAP&SAD".

Highlights ...

Before typing in another program requiring these subroutines, load the combined Happy and Sad subroutines into memory using the LOAD command.

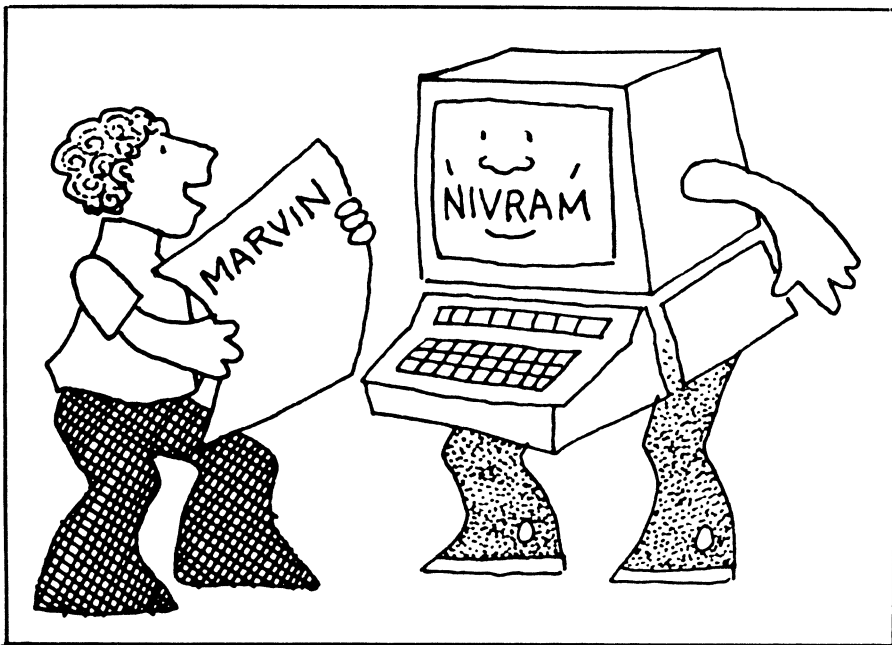
Use LOAD“HAP&SAD”,8 (disk) or LOAD“HAP&SAD” (tape).

Next, type in the new program.

Last, save the combined, new program with a SAVE command.

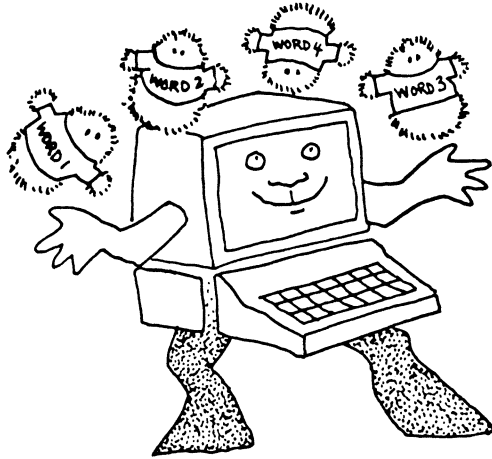
Use SAVE“NEW PROGRAM”,8 (disk) or SAVE“NEW PROGRAM” (tape).

WORDS



4

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
52 DIM A$(10)
55 POKE 53280,3 :REM CYAN BORDER
65 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}" T
AB(9) "*** SPELLING BEE ***"
66 FOR PAUSE=1 TO 1000:NEXT PAUSE
81 S$="975318642090817263543928170645675
8493021"
100 FOR SPELL=1 TO 10
110 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}{CD}{C
D}{CD}{CD}WORD #"SPELL;
115 INPUT A$(SPELL)
130 NEXT SPELL
135 MI=INT(RND(1)*4)
140 FOR SPELL=1 TO 10
145 S1 = VAL(MID$(S$,MI*10+SP,1))+1
160 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}{CD}{C
D}{CD}{CD}{CD}WORD: "A$(S1)
170 FOR PAUSE=1 TO 800:NEXT PAUSE
180 INPUT"{SC}{CD}{CD}{CD}{CD}{CD}{CD}{C
D}{CD}{CD}{CD}WORD ";B$
190 IF A$(S1)=B$ THEN GOSUB 1010:GOTO 21
0
200 GOSUB 2010:GOTO160
210 NEXT SPELL
220 INPUT"{SC}{CD}{CD}{CD}{CD}{CD}{CD}{C
D}{CD}{CD}{CD}SAME WORDS AGAIN";A$
230 IF LEFT$(A$,1)="Y" THEN 135
240 IF LEFT$(A$,1)<>"N" THEN 220
250 PRINT"{SC}":END

```

Typing Hints ...

Remember to LOAD the Happy and Sad routine (HAP& SAD) before typing in this program.

Highlights ...

The program uses any array A\$() to store the words. On line 52, the computer is told to set aside 10 spaces in the array A\$() by a DIMension statement.

This program does everything in two small loops.

First, on lines 100 to 130, the computer lets you enter each word you want included in the spelling bee. It stores your answer (A\$) in the "W" variables, from W\$(1) to W\$(10). It expects 10 spelling words.

The computer uses the second loop (lines 140 to 210) to quiz you on the spelling words you just entered and scramble 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 145, the computer uses the following function to pick each of the 10 spelling-bee words:

$$S1 = \text{VAL}(\text{MID}\$(S\$,MI*10+SP,1))+1$$

We interpret this function (as the computer does) from the inside out. First, we know the variable MI was chosen on line 135. It is a random number between 0 and 3. Let's say that MI 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. Note that SP and SPELL are identical in Commodore BASIC.

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

$$S1 = \text{VAL}(\text{MID}\$(S\$,3*10+1,1))+1$$

This translates to VAL (MID\$(S\$,31,1))+1. We know that (MID\$(S\$,31,1) 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, so S1 = 7.

Now we are ready to pick our spelling-bee word. Look at line 160. The computer prints out A\$(S1) which is the seventh word.

Voila!

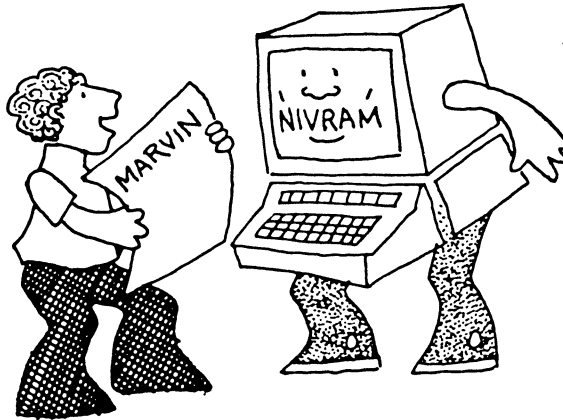
Note that on line 230, we now send the computer back to line 135. Line 135 is where the computer juggles the words and chooses a new scrambled word sequence.

Variables ...

- A\$()** Array—stores correct spelling of words.
- B\$** Stores your answer.
- M1** Selects a random pattern for the words to be displayed.
- SPELL** Loop counter—index for storing and retrieving spelling-bee words.
- PAUSE** Delay loop counter.
- S1** Subscript of the particular word picked from array A\$().
- S\$** Stores the four possible sequences in which words can be displayed.

5

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 *** BACKWORDS GAME
55 POKE 53280,3 :REM CYAN BORDER
65 PRINT "{SC}{CD}{CD}{CD}{CD}{CD}" TAB(
9)  "*** BACKWORDS GAME ***"
66 FOR PAUSE=1 TO 1000:NEXT PAUSE
80 OPEN 1,0
100 PRINT "{SC}"
105 M$="":M2$=""
110 PRINT TAB(9) "FRONTWORDS?"
120 PRINT
130 INPUT#1,M$
160 PRINT "{SC}{CD}{CD}{CD}{CD}{CD}{CD}"
170 PRINT "          {RV}BACKWORDS MACHINE NO
W WORKING"
180 FOR PAUSE=1 TO 1500:NEXT PAUSE
190 FOR I=LEN(M$) TO 1 STEP -1
195 IF MID$(M$,I,1)=" " THEN M2$=M2$+" "
:GOTO 210
200 M2$=M2$+MID$(M$,I,1)
210 NEXT I
220 PRINT "{SC}"
230 PRINT TAB(9) "FRONTWORDS:"
240 PRINT
250 PRINT M$
260 PRINT:PRINT:PRINT
270 PRINT TAB(9) "BACKWORDS:"
280 PRINT
290 PRINT M2$
300 PRINT:PRINT:PRINT
310 PRINT TAB(9) "PLAY AGAIN";:INPUT A$
320 IF LEFT$(A$,1)="Y" THEN 100

```



```
330 IF LEFT$(A$,1) <> "N" THEN 310
340 CLOSE 1
350 PRINT "{SC}":END
```

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\$.

On line 80, channel 1 is open for the keyboard (device 0). This suppresses the “?” normally associated with the INPUT instruction. INPUT # is then used in line 130. The “?” appears as part of the PRINT command in line 110. The channel is closed in line 340.

Do-It-Yourself ...

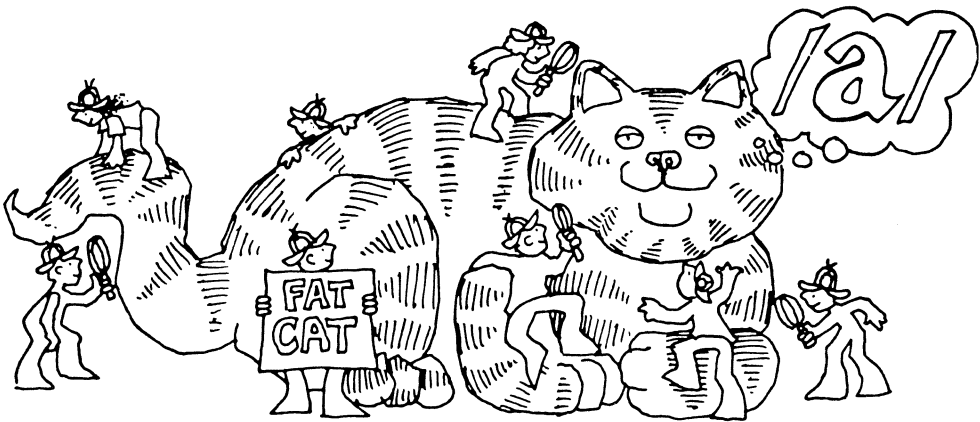
You might modify this program so that you can store the “backwards” on tape or disk.

Also, “backwards” 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—ininitely malleable.

One way to get your computer to play with words is to translate all the words from regular video into **reverse video**. This is accomplished by using ASCII character 18 (PRINT CHR\$(18)). You can change the Backwords program so that every time you type in a letter A, for example, the program 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.

6

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 ex-

ample, 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
55 POKE 53280,3 :REM CYAN BORDER
65 PRINT "{SC}{CD}{CD}{CD}{CD}{CD}"
TAB(9) "*** FAT CAT GAME ***"
66 FOR PAUSE=1 TO 1000:NEXT PAUSE
70 DIM P$(9,5)
75 PRINT "{SC}"
80 GOSUB 4510 :REM READ ARRAY
90 POKE 53281,0:POKE 53280,14
100 FOR W=1 TO 5
102 TRY=0
110 PRINT CHR$(14):GOSUB 3005 :REM SET U
P SCREEN
115 IF TRY=0 THEN GOSUB 4010 :REM SELECT
WORD
120 GOSUB 4110 :REM PRINT WORD
125 GOSUB 3520
130 POKE 53281,6:POKE 53280,3:PRINT "{SC
}"CHR$(142)
135 IF PH=C THEN GOSUB 1010:GOTO 150
140 GOSUB 2010:TRY=1:POKE 53281,0:POKE 5
3280,14:GOTO 110
150 POKE 53281,0:POKE 53280,14:NEXT W
160 POKE 53281,6:POKE 53280,3:INPUT "{HM
}{CD}{CD}PLAY AGAIN";A$:A$=LEFT$(A$,1)
170 IF A$="Y" THEN RESTORE:GOTO 75
180 IF A$<>"N" THEN 160
190 PRINT "{SC}":END

```

```
3000 REM *** SET UP PHONETIC SOUNDS
3005 SC=1024
3010 LI=2:OF=6:GOSUB 3200:OF=7:C=1:GOSUB
3210:OF=8:GOSUB 3200:REM /A/
3020 LI=5:OF=6:GOSUB 3200:OF=7:C=3:GOSUB
3210
3025 OF=8:C=8:GOSUB 3210:OF=9:GOSUB 3200
:REM /CH/
3030 LI=8:OF=6:GOSUB 3200:OF=7:C=5:GOSUB
3210:OF=8:GOSUB 3200:REM /E/
3040 LI=11:OF=6:GOSUB 3200:OF=7:C=6:GOSU
B 3210:OF=8:GOSUB 3200:REM /F/
3050 LI=14:OF=6:GOSUB 3200:OF=7:C=12:GOS
UB 3210:OF=8:GOSUB 3200:REM /L/
3060 LI=2:OF=17:GOSUB 3200:OF=18:C=19:GO
SUB 3210
3065 OF=19:C=8:GOSUB 3210:OF=20:GOSUB 32
00:REM /SH/
3070 LI=5:OF=17:GOSUB 3200:OF=18:C=19:GO
SUB 3210:OF=19:GOSUB 3200 :REM /S/
3080 LI=8:OF=17:GOSUB 3200:OF=18:C=20:GO
SUB 3210
3085 OF=19:C=8:GOSUB 3210:OF=20:GOSUB 32
00:REM /TH/
3090 LI=11:OF=17:GOSUB 3200:OF=18:C=114:
GOSUB 3210:OF=19:GOSUB 3200
3095 LI=12:OF=18:C=91:GOSUB 3210:REM PHO
NETIC
3110 PRINT "{HM}":FOR I=1 TO 20:PRINT "{
CD}{CL} ";:NEXT I
3115 PRINT SPC(6) "{RV}{CY}RETURN{RO} =
YES {RV}SPACE{RO} = NO";
3120 RETURN
3200 POKE SC+LI*40+OF,47 :REM "/"
3205 POKE 55296+LI*40+OF,7 :RETURN :REM
COLOR
3210 POKE SC+LI*40+OF,C :REM PRINT CHAR
3215 POKE 55296+LI*40+OF,7 :RETURN :REM
COLOR
3500 REM *** ARROW SUBROUTINE
3520 OX=-1:OY=-1:C=0
3550 FOR X=4 TO 15 STEP 11:FOR Y=0 TO 12
STEP 3
3552 C=C+1
```

```
3555 IF OX<>-1 THEN GOSUB 3655
3560 POKE SC+(Y+2)*40+X,62:POKE 55296+(Y
+2)*40+X,1
3570 GET A$:FOR L=1 TO 100:NEXT
3580 IF A$=CHR$(13) THEN X=15:Y=12:GOTO
3625
3590 IF A$=CHR$(32) THEN 3610
3600 GOTO 3570
3610 OY=Y:OX=X
3620 IF X=15 AND Y=9 THEN GOSUB 3655:Y=1
2:NEXT Y:NEXT X:GOTO 3520
3625 NEXT Y:NEXT X
3630 RETURN
3650 REM *** ERASE ARROW
3655 POKE SC+(OY+2)*40+OX,32
3670 RETURN
4000 REM *** SELECT SOUND/WORD
4010 PH=(INT(RND(1)*9)+1)
4030 WD=INT(RND(1)*5)+1
4040 W$=P$(PH,WD)
4090 RETURN
4100 *** REM PRINT WORD
4110 PRINT "{HM}{CD}{CD}{CD}{CD}{CD}{CD}
{CD}{WH}"SPC(27)W$
4130 RETURN
4500 REM *** SET UP ARRAY
4510 FOR I=1 TO 9:FOR J=1 TO 5
4520 READ P$(I,J)
4530 NEXT J:NEXT I
4540 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, 47 is POKEd to the screen in line 3200.

To make the special phonetic symbol (**ƒ**), you need to use two graphics characters. This is defined in lines 3090 and 3095, using screen characters 91 and 114.

Remember to LOAD the Happy and Sad routine (HAP& SAD) before typing in this program.

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.

On line 110 the command PRINT CHR\$(14) changes the character set from graphics and upper-case letters to lower-case letters.

Variables ...

- | | |
|--------------|--|
| PAUSE | Delay loop counter. |
| A\$ | Child's yes-no answer to “PLAY AGAIN?” and character code for either SPACE or RETURN (lines 3570 to 3590). |
| W\$ | Word with mystery phoneme. |
| W | Loop counter—controls number of words/game. |
| TRY | If TRY=1, child has given an incorrect answer. |

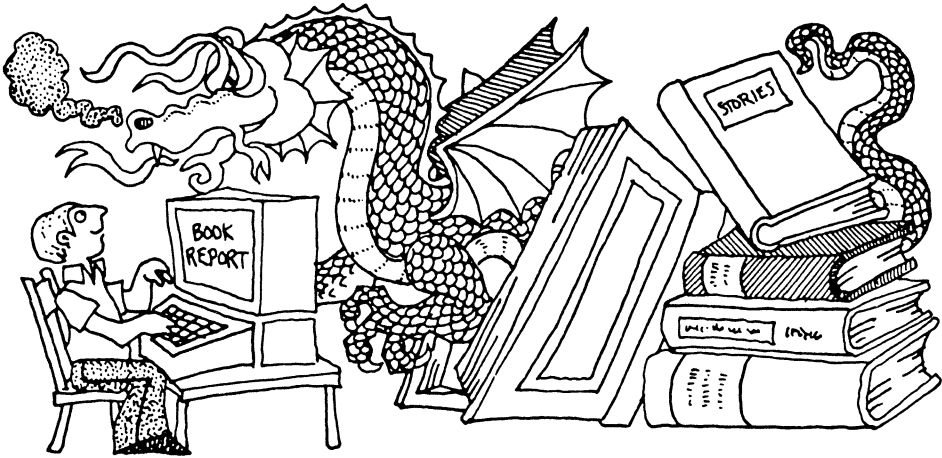
PH	Pointer to list of words containing a certain phoneme.
WD	Pointer to one word in the list.
C	Pointer to phoneme symbol (child's answer) in lines 135 and 3552. Also, screen code of character to be printed (line 3210).
X	Current column position of arrow.
Y	Current row position of arrow.
OX	Old column position of arrow.
OY	Old row position of arrow.
P\$	Array containing the list of words.
LI	Line to print character and color.
OF	Offset into line to print character and color.
SC	Start of screen memory.

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?

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
55 POKE 53280,3 :REM CYAN BORDER
65 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}" TAB
(10) "*** BOOK REPORT ***"
67 FOR PAUSE=1 TO 1000:NEXT PAUSE
70 DIM R$(100)
92 B$="
    ":REM 40 SPACES INSIDE QUOTES
95 P=1
100 PRINT "{SC}"
110 INPUT "WHO'S OUT THERE";NM$
120 PRINT "{SC}"
125 INPUT "TODAY'S DATE";DT$
127 PRINT "{SC}"
130 INPUT "BOOK TITLE";T$

```

```
140 PRINT "{SC}"
150 INPUT "AUTHOR";A$
160 PRINT "{SC}"
170 PRINT "WHAT HAPPENED FIRST?"
175 OPEN 1,0:INPUT#1,L$
180 R$(P)=" "
190 R$(P)=R$(P)+L$
200 PRINT "{SC}"
205 PRINT "WHAT HAPPENED NEXT?"
210 INPUT#1,L$
215 IF L$="END" OR L$="THE END" THEN 300
220 IF MID$(R$(P),LEN(R$(P)),1)="." THEN R
$(P)=R$(P)+" "
227 IF L$=CHR$(13) THEN 235
230 R$(P)=R$(P)+L$
235 P=P+1
240 L$="":GOTO 180
300 PRINT"{SC}"
303 L$="BOOK REPORT BY"
304 GOSUB 3010 :REM *** CENTER LINE
305 PRINT SPC(X)L$
306 L$ =NM$
307 GOSUB 3010
308 PRINT SPC(X)L$
309 L$=DT$:GOSUB 3010:PRINT:PRINT SPC(X)L$
310 L$=T$:GOSUB 3010:PRINT:PRINT SPC(X)L$
320 GOSUB 3510 :REM *** TITLE LINE
330 PRINT SPC(X)L$
340 L$="BY ":L$=L$+A$:GOSUB 3010:PRINT :PR
INT SPC(X)L$
345 PRINT:PRINT
350 GOSUB 4010 :REM *** PRINT REPORT
355 IF PEEK(214)>23 THEN GOSUB 4510
360 L$="THE END":GOSUB 3010:PRINT:PRINT SP
C(X)L$
390 GOTO 390
3000 REM *** CENTER LINE SUBROUTINE
3010 X=INT((40-LEN(L$))/2)
3015 IF X<0 THEN X=0
3020 RETURN
3500 REM *** TITLE LINE SUBROUTINE
3510 L$=""
3520 FOR I=1 TO LEN(T$)
3530 L$=L$+"-"
```

```

3540 NEXT I
3550 RETURN
4000 REM *** PRINT REPORT
4010 Q=P:D=0:P=1
4015 IF LEN(R$(P))-D>39 THEN 4018
4016 PRINT MID$(R$(P),D+1,LEN(R$(P))-D):IF
  PEEK(214)>23 THEN GOSUB 4510
4017 GOTO 4120
4018 RS=39
4020 IF MID$(R$(P),D+RS,1)=" " THEN 4090
4030 FOR RS=39 TO 1 STEP -1
4040 IF MID$(R$(P),D+RS,1)=" " THEN 4090
4050 NEXT RS
4090 IF PEEK(214)>23 THEN GOSUB 4510
4095 PRINT MID$(R$(P),D+1,RS)
4096 D=D+RS
4100 IF D>=LEN(R$(P)) THEN 4120
4110 GOTO 4015
4120 P=P+1
4130 IF P>Q THEN RETURN
4140 D=0
4150 GOTO 4015
4500 REM *** PAUSE/CLEAR ROUTINE
4510 FOR PAUSE=1 TO 4000:NEXT PAUSE
4520 PRINT"{HM}{CD}{CD}{CD}{CD}{CD}{CD}{CD}
  {CD}{CD}{CD}{CD}"
4530 FOR L=1 TO 12 :PRINT B$
4540 NEXT L
4550 PRINT"{HM}{CD}{CD}{CD}{CD}{CD}{CD}{CD}
  {CD}{CD}{CD}{CD}"
4560 RETURN

```

Highlights ...

The computer copies the entire story or book report into a giant character string array called R\$, which has been DIMensioned in line 70 for 100 messages.

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 205). The **OPEN 1,0** command on line 175 opens channel 1 to device 0 (keyboard). This allows **INPUT#1** on lines 175 and 210 to accept characters without printing the "?" prompt.

When you press the **RETURN** button without entering any other information—line 227 checks this—the computer starts a new string within the array **R\$** (line 235). When the computer prints out the report, it ends an old paragraph and creates a new, indented paragraph automatically by adding these spaces to the beginning of the string (line 180).

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 the **PEEKs** to memory position 214 on lines 355 and 4090? The computer goes to location 214 to find where it is on the TV screen as it prints out lines of text. The computer has only 23 lines. The book report uses location 214 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 (an array).
L\$	A single text line.
DT\$	The current date.
B\$	A blank line (to erase a single screen line).
P	Points to the next unoccupied spot in array R\$.
X	The column position of centered text.
I	Loop counter for centering title line.
L	Loop counter in loop that erases old text from screen.
Q	Holds the total number of text lines in the book report.
RS	Ending position of current line in text string array R\$.
D	Starting position of current line in string array R\$.

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 4000. 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.

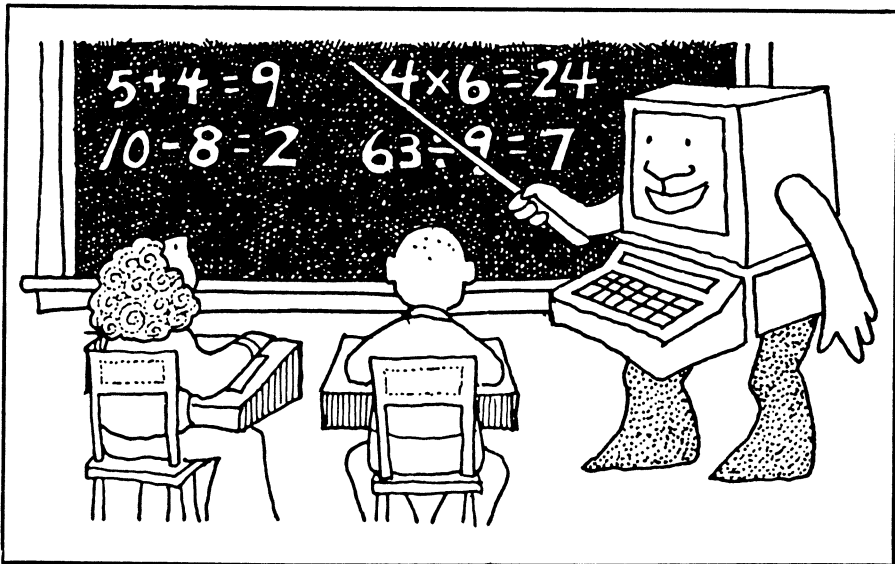
And how about a subroutine that takes the report and stores it on a tape or disk? Your child might spend a long time typing in a report and might want to save sections of it at one time on tape or disk to work on later.

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.

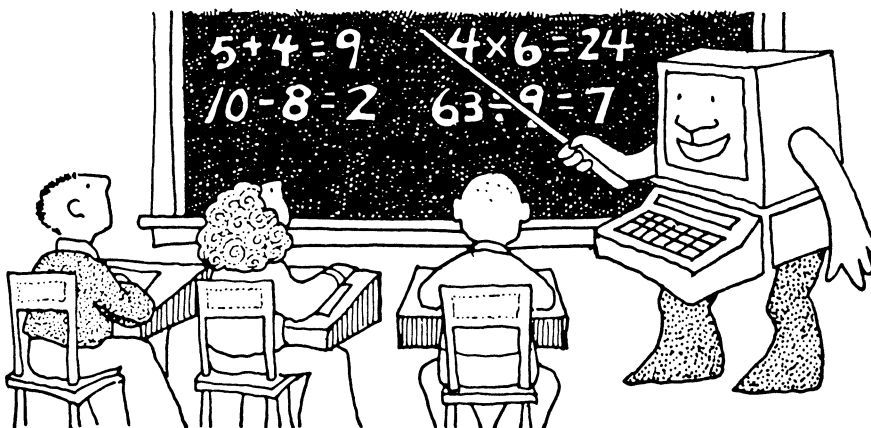


NUMBERS



8

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 **INST/DEL** 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
55 POKE 53280,3 :REM CYAN BORDER
65 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}" T
AB(6) "*** THE ARITHMETIC GAME ***"
66 FOR PAUSE=1 TO 1000:NEXT PAUSE
100 GOSUB 3010 :REM *MENU
170 FOR Y=1 TO 5
180 N1=INT(RND(1)*10)+1
190 N2=INT(RND(1)*10)+1
195 POKE 53281,0:POKE 53280,0:PRINT"{YL}
"
200 GOSUB 3510 :REM * GAME
250 NEXT Y

```

```
260 PRINT"{SC}"
265 INPUT"{CD}{CD}{CD}{CD}{CD} PLAY AGA
IN";A$
270 IF LEFT$(A$,1)="Y" THEN 100
280 IF LEFT$(A$,1)<>"N"THEN 260
300 PRINT"{SC}":END
3000 REM *** MENU SUBROUTINE
3010 PRINT"{SC}{CD}{CD}"
3020 PRINT TAB(12) "{CD}1.  ADDING"
3030 PRINT TAB(12) "{CD}2.  SUBTRACTING"
3040 PRINT TAB(12) "{CD}3.  MULTIPLYING"
3050 PRINT TAB(12) "{CD}4.  DIVIDING"
3060 PRINT TAB(9)  "{CD}{CD}{CD}{CD}{CD}{
CD}WHICH NUMBER
3065 PRINT TAB(9)  "(1, 2, 3, OR 4)";:INP
UT B$:B=VAL(B$)
3070 IF B>=1 AND B<=4 THEN 3090
3080 GOTO 3010
3090 RETURN
3500 REM *** ARITHMETIC SUBROUTINE
3510 PRINT "{SC}"
3512 POKE 53281,0:POKE 53280,0:PRINT"{WH
}"
3513 N$=""
3515 ON B GOSUB 6010,6110,6210,6310
3517 X=25:IF N1>9 OR N2>9 THEN X=26
3518 IF N1>9 AND N2>9 THEN X=27
3520 GET K$
3521 IF K$="" THEN 3520
3522 IF ASC(K$)<>20 THEN 3530
3523 IF ASC(K$)=20 AND LEN(N$)=0 THEN 35
20
3524 PRINT "{CU}"SPC(X+LEN(N$)-1)" "
3525 IF LEN(N$)=1 OR LEN(N$)=0 THEN N$=""
:GOTO 3520
3526 N$=LEFT$(N$,LEN(N$)-1)
3528 GOTO 3520
3530 IF K$=CHR$(13) THEN 3600
3540 IF VAL(K$)<0 OR VAL(K$)>9 THEN 3520
3545 IF LEN(N$)>4 THEN3520
3550 N$=N$+K$
3570 PRINT "{CU}"SPC(X)N$
3580 GOTO 3520
3600 ON B GOSUB 7010,7110,7210,7310
```

```
3602 POKE 53281,6:POKE 53280,3:PRINT"{PU
}"
3605 IF W=VAL(N$) THEN PRINT "{SC}":GOSU
B 1010:PRINT"{HM}":CT=0:GOTO 3650
3610 PRINT"{SC}":GOSUB 2010:CT=1:GOTO 35
10
3650 RETURN
6000 REM *** ADDING SUBROUTINE
6010 GOSUB 7500
6012 N=22:GOSUB 8000:PRINT SPC(17)"{WH}{
RV}ADDING{YL}"
6020 PRINT "{HM}{CD}{CD}{CD}{CD}{CD}{CD}
{CD} " SPC(12)N1" + "N2" = ?"
6030 RETURN
6100 REM *** SUBTRACTING SUBROUTINE
6110 GOSUB 7500
6112 N=22:GOSUB 8000:PRINT SPC(15)"{WH}{
RV}SUBTRACTING{YL}"
6120 IF N2>N1 THEN W=N2:N2=N1:N1=W
6130 PRINT "{HM}{CD}{CD}{CD}{CD}{CD}{CD}
{CD} " SPC(12)N1" - "N2" = ?"
6140 RETURN
6200 REM *** MULTIPLYING SUBROUTINE
6210 GOSUB 7500
6212 N=22:GOSUB 8000:PRINT SPC(15)"{WH}{
RV}MULTIPLYING{YL}"
6220 PRINT "{HM}{CD}{CD}{CD}{CD}{CD}{CD}
{CD} " SPC(12)N1" X "N2" = ?"
6230 RETURN
6300 REM *** DIVIDING SUBROUTINE
6310 GOSUB 7500
6312 N=22:GOSUB 8000:PRINT SPC(16)"{WH}{
RV}DIVIDING{YL}"
6315 IF CT=1 THEN 6330:REM SAME DIVIDEND
ON WRONG ANSWER
6320 W=N1*N2:N2=W
6330 PRINT "{HM}{CD}{CD}{CD}{CD}{CD}{CD}
{CD} " SPC(12)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
7500 SC=1024:BE=40*22:GS=55296
7510 DR=SC+BE:GR=GS+BE
7520 FOR I=0 TO 119:POKE DR+I,160:POKE G
R+I,3:NEXT I
7530 RETURN
8000 PRINT"{HM}":FOR I=1 TO N:PRINT"{CD}
{CL}":NEXT I
8010 RETURN
```

Typing Hints ...

Remember to LOAD the Happy and Sad routine (HAP& SAD) before typing in this program.

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. 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 3570) controls the 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 **INST/DEL** key to erase the number and start again.

The routine from lines 3520 to 3580 allows you to enter up to five digits (a number between 10,000 and 99,999). If you try to enter more than five digits, line 3545 ignores any new entries. Then the routine will accept only the **RETURN** key. This routine also processes numbers from the keyboard, prints them to the screen, and allows you to delete incorrect entries using the **INST/DEL** key ($ASC(K\$)=20$ or $\langle 20$) in lines 3522 and 3523.

If you incorrectly press a number key less than 1 or greater than 4 or press a non-number key to answer the computer's question on line 3065, then lines 3070 and 3080 make the computer jump back to line 3010 and print the menu and question again.

On lines 7500 to 7530 is a drawing routine that draws a cyan colored message area at the bottom of the screen. In line 4020, 120 reversed spaces (cursors) are drawn (3 screen lines). At the same time, corresponding color memory locations are **POKEd** with character 3 (cyan).

Line 8000 is a cursor positioning routine. The counter variable **N** is passed to this routine. First the cursor is placed at the home position (top left corner of the screen). The routine then positions the cursor **N** lines down, preparatory to writing a message in the cyan colored message area of the screen.

Variables ...

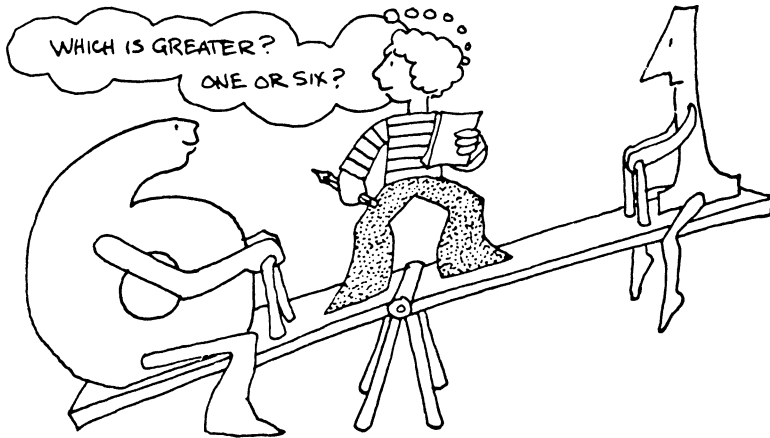
PAUSE	Delay loop counter.
A\$	Your answer to the question "PLAY AGAIN?"
N\$	Stores your answer to the arithmetic problem. (N\$ is built from 1-byte key entries originally stored in K\$ by the command GET K\$.)
Y	Loop counter—main loop.
N1	First random number for arithmetic problem.
N2	Second random number 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.
W	The computer's answer to the arithmetic problem.
CT	Used as an error flag to prevent the dividend from changing in line 6320 when the child has answered a division problem incorrectly.
SC	Holds the start of screen character memory.
GS	Holds the start of screen color memory.
DR	Holds the starting screen character memory address to begin drawing the message area.
GR	Holds the starting screen color memory address to begin drawing the message area color.
BE	Pointer to the start of the message area, BE=40*22 means that the start of the message begins after 22 40-character lines.

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.

9

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, less than the second, greater than or equal to the second, or less than or equal to the second.

The game helps children learn the number comparison symbols: $>$ (greater than), $<$ (less than), \geq (greater than or equal to), and \leq (less than or equal to).

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 $>$ symbol means **greater than**. A thousand is greater than two.

You can also write the numbers like this:

$$2 < 1000$$

The $<$ symbol 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:

$$X \geq 14$$

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

$$14 \leq X$$

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 *** THE GREATER THAN GAME
55 POKE 53280,3 :REM CYAN BORDER
65 PRINT" {SC}{CD}{CD}{CD}{CD}{CD}{CD}" T
```

```

AB(7) "*** GREATER-THAN GAME ***"
66 FOR PAUSE=1 TO 1000:NEXT PAUSE
100 GOSUB 3010 :REM *MENU
170 FOR Y=1 TO 5
180 N1=INT(RND(1)*10)+1
190 N2=INT(RND(1)*10)+1
195 POKE 53281,0:POKE 53280,0:PRINT"{YL}
"
200 ON B GOSUB 3510,4010,4510,5010
250 NEXT Y
260 PRINT"{SC}"
265 INPUT"{CD}{CD}{CD}{CD}{CD} TRY AGAI
N";A$
270 IF LEFT$(A$,1)="Y" THEN 100
280 IF LEFT$(A$,1)<>"N"THEN 260
300 PRINT"{SC}":END
3000 REM *** MENU SUBROUTINE
3010 B$="":PRINT"{SC}{CD}{CD}"
3020 PRINT TAB(9) "{CD}1. GREATER THAN .
. >"
3030 PRINT TAB(9) "{CD}2. LESS THAN ....
. <"
3040 PRINT TAB(9) "{CD}3. GREATER THAN":
PRINT TAB(12) "OR EQUAL ..... >="
3050 PRINT TAB(9) "{CD}4. LESS THAN":PRI
NT TAB(12) "OR EQUAL ..... <="
3060 PRINT TAB(9) "{CD}{CD}{CD}{CD}{CD}{
CD}WHICH NUMBER
3065 PRINT TAB(9) "(1, 2, 3, OR 4)";:INP
UT B$:B=VAL(B$)
3070 IF B>=1 AND B<=4 THEN 3090
3080 GOTO 3010
3090 RETURN
3200 REM *** INITIALIZE SCREEN
3210 PRINT "{SC}"
3212 POKE 53281,0:POKE 53280,0:PRINT"{WH
}"
3215 SC=1024:BE=40*22:GS=55296
3220 DR=SC+BE:GR=GS+BE
3230 FOR I=0 TO 119:POKE DR+I,160:POKE G
R+I,3:NEXTI
3240 PRINT"{HM}{CD}{CD}{CD}{CD}{CD}{CD}{
CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{
CD}{CD}{CD}{CD}{CD}{CD}{CD}"SPC(7)"{WH}{

```

```
RV}RETURN=YES{CY}           {WH}SPACE=NO{YL}
"
3250 RETURN
3500 REM *** GREATER-THAN (>)
3510 GOSUB 3210
3515 PRINT "{HM}{CD}{CD}{CD}{CD}{CD}{CD}
{CD} " SPC(12)N1" > "N2" ?"
3520 GET K$
3525 IF K$="" THEN 3520
3530 IF N1>N2 AND K$=CHR$(13) THEN 3580
3535 IF N1<=N2 AND K$=CHR$(32) THEN 3580
3540 IF N1>N2 AND K$=CHR$(32) THEN 3590
3545 IF N1<=N2 AND K$=CHR$(13) THEN 3590
3550 GOTO 3520
3570 RETURN
3580 GOSUB 7500:GOTO 3570
3590 GOSUB 7600:GOTO 3510
4000 REM *** LESS-THAN (<)
4010 GOSUB 3210
4015 PRINT "{HM}{CD}{CD}{CD}{CD}{CD}{CD}
{CD} " SPC(12)N1" < "N2" ?"
4020 GET K$
4025 IF K$="" THEN 4020
4030 IF N1<N2 AND K$=CHR$(13) THEN 4080
4035 IF N1>=N2 AND K$=CHR$(32) THEN 4080
4040 IF N1<N2 AND K$=CHR$(32) THEN 4090
4045 IF N1>=N2 AND K$=CHR$(13) THEN 4090
4050 GOTO 4020
4070 RETURN
4080 GOSUB 7500:GOTO 4070
4090 GOSUB 7600:GOTO 4010
4500 REM *** GREATER-THAN/EQUAL (>=)
4510 GOSUB 3210
4515 PRINT "{HM}{CD}{CD}{CD}{CD}{CD}{CD}
{CD} " SPC(12)N1" >= "N2" ?"
4520 GET K$
4525 IF K$="" THEN 4520
4530 IF N1>=N2 AND K$=CHR$(13) THEN 4580
4535 IF N1<N2 AND K$=CHR$(32) THEN 4580
4540 IF N1>=N2 AND K$=CHR$(32) THEN 4590
4545 IF N1<N2 AND K$=CHR$(13) THEN 4590
4550 GOTO 4020
4570 RETURN
4580 GOSUB 7500:GOTO 4570
4590 GOSUB 7600:GOTO 4510
```

```

5000 REM *** LESS-THAN/EQUAL (<=)
5010 GOSUB 3210
5015 PRINT "{HM}{CD}{CD}{CD}{CD}{CD}{CD}
{CD} " SPC(12)N1" <= "N2" ?"
5020 GET K$
5025 IF K$="" THEN 5020
5030 IF N1<=N2 AND K$=CHR$(13)THEN 5080
5035 IF N1>N2 AND K$=CHR$(32) THEN 5080
5040 IF N1<=N2 AND K$=CHR$(32) THEN 5090
5045 IF N1>N2 AND K$=CHR$(13) THEN 5090
5050 GOTO 5020
5070 RETURN
5080 GOSUB 7500:GOTO 5070
5090 GOSUB 7600:GOTO 5010
7500 PRINT"{SC}{PU}":POKE 53281,6:POKE 5
3280,3:GOSUB 1010:PRINT"{HM}"
7510 RETURN
7600 PRINT"{SC}{PU}":POKE 53281,6:POKE 5
3280,3:GOSUB 2010:PRINT"{HM}"
7610 RETURN

```

Typing Hints ...

Note the various control (CTRL) key sequences in line 3240 that print the words in reverse video, with white letters on a cyan blue background.

Remember to LOAD the Happy and Sad routine (HAP& SAD) before typing in this program.

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 (see the subroutine beginning on line 3210). 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 ASCII value of 13). You give a "NO" answer by pressing the **SPACE** bar (an ASCII value of 32).

Line 7500 clears the screen and restores the character color to light blue, the screen text area to dark blue (POKE 53281,6), and the screen color to cyan (POKE 53280,3). It then calls the Happy Face subroutine at line 1010. After the face is cleared, the routine positions the cursor to the home screen position. Line 7600 performs an identical function, except that the Sad Face subroutine is called at line 2010.

Variables ...

PAUSE	Delay loop counter.
A\$	Answer to the question "TRY AGAIN?"
Y	Loop counter—main problem 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).
SC	Holds the start of screen character memory.

- GS Holds the start of screen color memory.
- DR Holds the starting screen character memory address to begin drawing the message area.
- GR Holds the starting screen color memory address to begin drawing the message area color.
- BE Pointer to the start of the message area; BE=40*22 means that the start of the message begins after 22 40-character lines.

Do-It-Yourself ...

You can add a new problem type—not equal (\neq).

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 symbols ($>$, $<$, \geq , \leq , or \neq).

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:

$$\text{IF } X = 100 \text{ THEN IS } X/5 \geq .1X + 9 ?$$

10

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 fires a pistol into the sky. You and the other children start munching burgers.

Half an hour later, the President fires the pistol again. "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 ***BURGER CONTEST
45 POKE 52,48:POKE56,48
50 POKE 53281,0:POKE 53280,0
55 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{
CD}"SPC(16) "{YL}HAMBURGER"
60 PRINTSPC(17) "{CD}{CD}{CD}CONTEST"
70 GOSUB 4000
80 PRINT"{SC}"
120 MAN=INT(RND(0)*9+1):HAM=INT(RND(0)*9+1
)
130 T=HAM:B=MAN
210 GOSUB 3000
300 PRINT"{WH}{HM}"
310 FOR I=1 TO 20:PRINT"{CD}{CL}":NEXT
320 IF B=1 AND T=1 THEN PRINT "IF" B "PERS
ON EATS" T "BURGER,":GOTO 330
321 IF B>1 AND T=1 THEN PRINT "IF" B "PEOP
LE EAT" T "BURGER EACH,":GOTO 330
322 IF B=1 AND T>1 THEN PRINT "IF" B "PERS
ON EATS" T "BURGERS,":GOTO 330
323 IF B>1 AND T>1 THEN PRINT "IF" B "PEOP
LE EAT" T "BURGERS EACH,":GOTO 330

```

```
330 PRINT:PRINT "{CY}HOW MANY BURGERS ARE
EATEN";:INPUT FF
350 IF FF=B*T THEN 380
360 PRINT"{SC}{PU}":POKE 53272,21:POKE 532
81,6:POKE 53280,3:GOSUB 2010
365 POKE 53281,0:POKE 53280,0:POKE 53272,(
PEEK(53272) AND 240)+12:GOTO 210
380 PRINT"{SC}{PU}":POKE 53272,21:POKE 532
81,6:POKE 53280,3:GOSUB 1010
385 POKE 53272,(PEEK(53272) AND 240)+12
400 INPUT"{SC}{CD}{CD}{CD}{CD}{CD}PLAY AGA
IN";A$:A$=LEFT$(A$,1)
410 IF A$="Y"THEN POKE 53281,0:POKE 53280,
0:GOTO 80
420 IF A$<>"N"THEN 400
430 PRINT"{SC}":END
3000 REM ***MAN SUBROUTINE
3010 REM B=MEN
3020 X=4:Y=0
3030 FOR I=1 TO B
3040 FOR L=0 TO 1:FOR M=0 TO 2
3050 POKE 1024+X+40*Y+M*40+L,33+3*L+M
3060 POKE 55296+X+40*Y+M*40+L,4
3070 NEXT:NEXT
3080 X=X+5:IF X>14 THEN X=4:Y=Y+7
3090 NEXT
3200 REM ***BURGER SUBROUTINE
3210 REM T=BURGERS
3220 X=24:Y=0
3230 FOR I=1 TO T
3240 FOR L=0 TO 1:FOR M=0 TO 1
3250 POKE 1024+X+40*Y+M*40+L,39+2*L+M
3260 POKE 55296+X+40*Y+M*40+L,2
3270 NEXT:NEXT
3280 X=X+5:IF X>34 THEN X=24:Y=Y+7
3290 NEXT
3400 REM ***MULTIPLY (X) SUBROUTINE
3410 REM X- POS. BASED ON BURGER & MEN ROW
S
3412 IF B<4 AND T<4 THEN Y=0:GOTO 3420
3415 IF B<7 AND T<7 THEN Y=4:GOTO 3420
3418 Y=7
3420 X=19
3440 FOR L=0 TO 1:FOR M=0 TO 1
```

```
3450 POKE 1024+X+40*Y+M*40+L,28+2*L+M
3460 POKE 55296+X+40*Y+M*40+L,1
3470 NEXT:NEXT
3490 RETURN
4000 REM ***CHANGE CHARACTER MEMORY
4010 POKE 56334,PEEK(56334) AND 254 :REM T
URN OFF KEY SCAN INTERVAL TIMER
4020 POKE 1,PEEK(1) AND 251 :REM TURN OFF
I/O
4030 FOR I=0 TO 511:POKEI+12288,PEEK(I+532
48):NEXT
4040 POKE 1,PEEK(1) OR 4 :REM TURN ON I/O
4050 POKE 56334,PEEK(56334) OR 1 :REM TURN
ON KEY SCAN INTERVAL TIMER
4060 POKE 36869,255:REM CHANGE CHAR. POINT
ER
4070 FOR I=13568 TO 13575:POKE I,255:NEXT
:REM CURSOR (160) IN RAM
4200 REM ***BUILD MAN
4210 FOR MN=12552 TO 12599:READ I:POKE MN,
I:NEXT
4220 DATA 255,128,128,128,136,136,128,129
4230 DATA 065,033,032,032,032,048,016,019
4240 DATA 024,008,008,012,004,006,003,000
4250 DATA 255,001,001,001,017,017,001,129
4260 DATA 130,132,004,004,004,012,008,200
4270 DATA 024,016,016,048,032,096,192,000
4300 REM ***BUILD BURGER
4310 FOR BU=12600 TO 12631:READ I:POKE BU,
I:NEXT
4320 DATA 000,000,031,032,032,032,255,255
4330 DATA 255,255,032,032,032,031,000,000
4340 DATA 000,000,248,004,004,004,255,255
4350 DATA 255,255,004,004,004,248,000,000
4400 REM ***BUILD (X) OPERATOR
4410 FOR MY=12512 TO 12543:READ I:POKE MY,
I:NEXT
4420 DATA 024,008,012,004,006,003,001,001
4430 DATA 001,001,003,006,004,012,008,024
4440 DATA 024,016,048,032,096,192,128,128
4450 DATA 128,128,192,096,032,048,016,024
4460 POKE 53272,(PEEK(53272)AND 240)+12 :R
EM POINT TO RAM ADDRESS AT 12288
4470 RETURN
```

Typing Hints ...

Remember to LOAD the Happy and Sad routine (HAP&SAD) before typing in this program.

Background ...

The Hamburger Contest game was designed 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 line 120. The number of people and hamburgers varies between one and nine.

Graphics are drawn by constructing the hamburgers, faces, and multiplication symbol from user-supplied characters POKEd into RAM. A detailed explanation of the technique is found in the *Commodore 64 User's Guide*. Note the use of POKE 53272 in lines 365, 380, and 4460. This causes the computer to use the RAM character set. POKE 53272,21 in

lines 360 and 380 causes the computer to use the standard ROM characters before printing either the sad or happy face on lines 360 and 380. Data to draw the hamburger are on lines 4320 to 4350. Face data are on lines 4220 to 4270. The multiplication symbol data are on lines 4420 to 4450. Line 4030 transfers the standard character set to RAM memory addresses 12288 to 12799. Faces are drawn in the subroutine in lines 3000 to 3090, burgers in lines 3200 to 3290, and the multiplication symbol in lines 3400 to 3490. Notice the instructions that determine the position of the multiplication symbol in lines 3410 to 3415. Line 45 lowers BASIC memory to allow room for the RAM character set.

Variables ...

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.
X	Column position of a hamburger, person, and multiplication symbol displayed on the TV screen. Also controls the number of burgers and faces per line.
Y	Row position of a hamburger, person, and multiplication symbol.
L, M	Loop counters—used to position face, multiplication symbol, and user character selection.
FF	Your answer—the number of burgers that you think are eaten.
I	Loop counter and READ variable.

BU, MY, MN

Loop counters—memory locations in RAM for hamburgers, multiplication symbol, and people, respectively.

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.

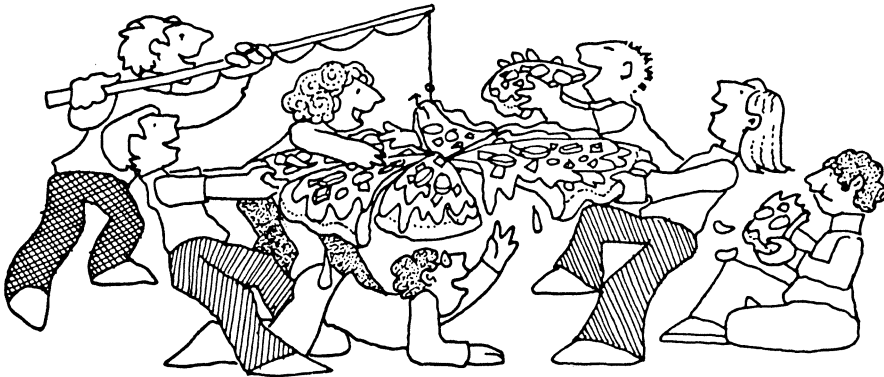
You can experiment with ways to make the color of the people different from the color of the hamburgers.

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.

11

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**

```
45 POKE 52,48:POKE 56,48
50 POKE 53281,0:POKE 53280,0
55 PRINT "{SC}{CD}{CD}{CD}{CD}{CD}{CD}" SP
C(12) "{YL}PEPPERONI PLEASE!"
60 PRINT SPC(13) "{CD}{CD}{CD}{CD}AN EXERC
ISE IN"
```

```
70 PRINT SPC(15) "{CD}{CD}{CD}{CD}FRACTION
S":GOSUB 4010
80 DIM S5(48),S6(52),S7(55),S8(44)
100 FOR I=1 TO 48:READ S5(I):NEXT
110 FOR I=1 TO 52:READ S6(I):NEXT
120 FOR I=1 TO 55:READ S7(I):NEXT
130 FOR I=1 TO 44:READ S8(I):NEXT
190 PRINT "{SC}":S=RND(-TI):S=INT(RND(1)*7
)+2
200 GOSUB 3210:ON S-1 GOSUB 4410,4510,4610
,4710,4810,4910,5010
210 GOSUB 6000:PRINT TAB(12) "COUNT THE SL
ICES"
213 FOR D=1 TO 3000:NEXT:GOSUB 6000:PRINT
TAB(12) "COUNT THE PEOPLE"
215 IF CT<>0 THEN 250
230 ON S-1 GOSUB 610,660,710,760,810,860,9
10
250 GOSUB 3010
260 FOR D=1 TO 2500:NEXT
330 GOSUB 6000:PRINT TAB(4) "{CY}HOW MANY
SLICES PER PERSON";
335 INPUT YY
340 PRINT TAB(2) "{CU}HOW MANY SLICES IN T
HE WHOLE PIE";
350 INPUT YZ
380 IF YY=S/N AND YZ=S THEN 395
384 CT=CT+1:IF CT>2 THEN CT=0:PRINT "{SC}"
:GOTO 397
385 GOSUB 6000
386 PRINT "
387 GOSUB 6000: PRINT TAB(11) "{PU}SORRY!
TRY AGAIN!{WH}
390 FOR D=1 TO 3000:NEXT:PRINT "{SC}":GOTO
200
395 PRINT "{SC}{PU}":POKE 53272,21:POKE 53
281,6:POKE 53280,3:GOSUB 1010
396 POKE 53272,(PEEK (53272) AND 240)+12 :
REM POINT TO RAM AT ADDRESS 12288
397 POKE 53281,0:POKE 53280,0:GOSUB 8010:P
OKE 53281,6:POKE 53280,3
400 INPUT "{SC}{PU}{CD}{CD}{CD}{CD}{CD}
}PLAY AGAIN";A$:A$=LEFT$(A$,1)
410 IF A$="Y" THEN POKE 53281,0:POKE 53280
,0:PRINT "{WH}":CT=0:GOTO 190
```

```
420 IF A$<>"N"THEN 400
430 PRINT "{SC}":END
600 REM 2 PEOPLE
610 N=S
620 RETURN
650 REM 3 PEOPLE
660 N=S
670 RETURN
700 REM 2 OR 4 PEOPLE
710 N=4/(INT(RND(1)*2+1))
720 RETURN
750 REM 5 PEOPLE
760 N=S
770 RETURN
800 REM 2,3, OR 6 PEOPLE
810 N=6/(INT(RND(1)*3+1))
820 RETURN
850 REM 7 PEOPLE
860 N=S
870 RETURN
900 REM 2,4, OR 8 PEOPLE
910 N=(INT(RND(1)*4+1))
920 IF N=3 THEN 910
930 N=8/N
940 RETURN
3000 REM *** DRAW PEOPLE
3010 X=10:Y=5:FOR I=1 TO N
3020 FOR L=0 TO 1:FOR M=0 TO 2
3030 POKE 1024+X+40*Y+M*40+L,57+3*L+M
3040 POKE 55296+X+40*Y+M*40+L,1
3050 NEXT:NEXT:X=X+4:NEXT:RETURN
3200 REM *** DRAW PIZZA
3210 X=5:Y=7:Z=0
3220 FOR L=0 TO 3:FOR M=0 TO 3
3230 IF L=1 AND M=1 THEN Z=2
3240 POKE 1024+X+40*Y+M*40+L,27+4*L+M+Z
3250 POKE 55296+X+40*Y+M*40+L,1
3260 NEXT:NEXT:RETURN
4000 REM *** CHANGE CHARACTER MEMORY
4010 POKE 56334,PEEK(56334) AND 254 :REM T
URN OFF KEYSKAN INTERRUPT TIMER
4020 POKE 1,PEEK(1) AND 251 :REM TURN OFF
I/O
4030 FOR I=0 TO 511:POKE I+12288,PEEK(I+53
248):NEXT :REM MOVE CHARS. TO RAM
```

```
4040 POKE 1,PEEK(1) OR 4 :REM TURN ON I/O
4050 POKE 56334,PEEK(56334) OR 1 :REM TURN
ON KEYSKAN INTERRUPT TIMER
4060 FOR I=13568 TO 13575:POKE I,255:NEXT
:REM CURSOR (160) IN RAM
4100 REM *** BUILD PIZZA PIE
4110 FOR PI=12504 TO 12543:READ I:POKE PI,
I:NEXT
4120 FOR PI=12560 TO 12647:READ I:POKE PI,
I:NEXT
4130 DATA 000,000,001,002,004,008,016,032
4140 DATA 032,064,064,064,128,128,128,128
4150 DATA 128,128,128,128,064,064,064,032
4160 DATA 032,016,008,004,002,001,000,000
4170 DATA 015,112,128,000,000,000,000,000
4180 DATA 000,000,000,000,000,000,000,000
4190 DATA 000,000,000,000,000,000,000,000
4200 DATA 000,000,000,000,000,128,112,015
4210 DATA 240,014,001,000,000,000,000,000
4220 DATA 000,000,000,000,000,000,000,000
4230 DATA 000,000,000,000,000,000,000,000
4240 DATA 000,000,000,000,000,001,014,240
4250 DATA 000,000,128,064,032,016,008,004
4260 DATA 004,002,002,002,001,001,001,001
4270 DATA 001,001,001,001,002,002,002,004
4280 DATA 004,008,016,032,064,128,000,000
4300 REM *** BUILD PERSON
4310 FOR PE=12744 TO 12791:READ I:POKE PE,
I:NEXT
4320 DATA 007,008,016,036,033,032,018,009
4330 DATA 004,003,001,015,017,017,015,001
4340 DATA 006,012,024,048,048,048,048,048
4350 DATA 224,016,008,036,132,004,072,144
4360 DATA 032,192,128,240,136,136,240,128
4370 DATA 096,048,024,012,012,012,012,012
4380 POKE 53272,(PEEK (53272) AND 240)+12
:REM POINT TO RAM AT ADDRESS 12288
4390 RETURN
4400 REM *** BUILD 2 SLICES
4410 FOR I=12584 TO 12615:POKEI,PEEK(I) OR
128:NEXT:RETURN
4500 REM *** BUILD 3 SLICES
4510 FORI=12584 TO 12600:POKEI,PEEK(I) OR
128:NEXT
```

```
4520 POKE 12523,129:POKE 12524,78:POKE 125
25,112:POKE 12569,7:POKE 12570,56
4530 POKE 12571,192:POKE 12601,112:POKE 12
602,14:POKE 12603,1:POKE 12635,193
4540 POKE 12636,58:POKE 12637,6:RETURN
4600 REM *** BUILD 4 SLICES
4610 FORI=12584 TO 12615:POKEI,PEEK(I) OR
128:NEXT
4620 POKE 12520,255:POKE 12568,255:POKE 12
600,255:POKE 12632,255:RETURN
4700 REM *** BUILD 5 SLICES
4710 FOR I=1 TO 4 :POKE (I+12510),S5(I):NE
XT
4720 FOR I=1 TO 4 :POKE (I+12525),S5(I+4):
NEXT
4730 FOR I=1 TO 12:POKE (I+12562),S5(I+8):
NEXT
4740 FOR I=1 TO 28:POKE (I+12585),S5(I+20)
:NEXT
4750 POKE 12632,255
4760 RETURN
4761 DATA 048,040,070,065
4762 DATA 065,034,036,024
4763 DATA 192,032,024,004,002,001,002,004,
024,032,064,128
4764 DATA 009,008,008,016,016,016,032,032,
032,064,064,064,128,128
4765 DATA 255,128,128,064,064,064,032,032,
032,016,016,016,008,009
4800 REM *** BUILD 6 SLICES
4810 POKE 12520,255:POKE 12568,255:POKE 12
600,255:POKE 12632,255
4820 FOR I=1 TO 5 :POKE (I+12538),S6(I):NE
XT
4830 FOR I=1 TO 21:POKE (I+12559),S6(I+5):
NEXT
4840 FOR I=1 TO 26:POKE (I+12587),S6(I+26)
:NEXT
4850 RETURN
4851 DATA 128,064,064,032,016,016,008,008,
004,002,002,001,001
4852 DATA 255,001,002,002,004,008,008,016,
016,032,064,064,128
4853 DATA 001,002,002,004,008,008,016,016,
032,064,064,128,255
```

```
4854 DATA 128,128,064,064,032,016,016,008,
008,004,002,002,001
4900 REM *** BUILD 7 SLICES
4910 FOR I=1 TO 3 :POKE (I+12512),S7(I):NE
XT
4920 FOR I=1 TO 3 :POKE (I+12523),S7(I+3):
NEXT
4930 FOR I=1 TO 6 :POKE (I+12537),S7(I+6):
NEXT
4940 FOR I=1 TO 22:POKE (I+12559),S7(I+12)
:NEXT
4950 FOR I=1 TO 21:POKE (I+12589),S7(I+34)
:NEXT
4960 POKE 12620,96:POKE 12621,144:POKE1263
2,255:POKE12643,160
4970 RETURN
4971 DATA 080,076,067
4972 DATA 065,070,088
4973 DATA 136,008,004,004,004,004,002,002,
002,002,193,049,013,003
4974 DATA 003,005,025,097,129,002,002,002,
002,004,004,004,008,136
4975 DATA 001,001,002,004,008,008,016,032,
064,128,255,128,064,032
4976 DATA 016,016,008,004,004,002,001
5000 REM *** BUILD 8 SLICES
5010 GOSUB 4610 :REM DRAW FOUR SECTIONS
5020 FOR I=1 TO 3 :POKE (I+12508),S8(I):NE
XT
5030 FOR I=1 TO 3 :POKE (I+12527),S8(I+3):
NEXT
5040 FOR I=1 TO 16:POKE (I+12559),S8(I+6):
NEXT
5050 FOR I=1 TO 16:POKE (I+12591),S8(I+22)
:NEXT
5060 FOR I=1 TO 3 :POKE (I+12620),S8(I+38)
:NEXT
5070 FOR I=1 TO 3 :POKE (I+12639),S8(I+41)
:NEXT
5080 RETURN
5081 DATA 012,018,033
5082 DATA 033,018,012
5083 DATA 128,064,032,016,008,004,002,001,
255
5084 DATA 002,004,008,016,032,064,128
```

```

5085 DATA 129,130,132,136,144,160,192,128,
255
5086 DATA 192,160,144,136,132,130,129
5087 DATA 048,072,132
5088 DATA 132,072,048
6000 PRINT "{HM}":FOR I=1 TO 18:PRINT "{CD
}{CL}":NEXT
6010 RETURN
8000 *** REM SUMMARY OF SLICES, PEOPLE AND
FRACTION
8010 PRINT "{HM}{YL}{CD}{CD}" TAB(9) S/N "
SLICE";:IF S/N<>1 THEN PRINT "S";
8020 PRINT " PER PERSON{CD}{CD}{CD}{CD}{CD
}":PRINT TAB(9) S "TOTAL SLICES"
8030 PRINT TAB(9) "{CD}{CD}{CD}{CD}{CD} EA
CH PERSON EATS{CD}{CD}{CD}{CD}{CD}"
8040 PRINT TAB(9) S/N "OF THE PIE":PRINT T
AB(9) " -":PRINT TAB(9) S
8050 FOR I=12504 TO 12543:POKE I,0:NEXT:FO
R I=12560 TO 12647:POKE I,0:NEXT
8060 RESTORE:GOSUB 4010:RETURN

```

Typing Hints ...

Remember to LOAD the Happy and Sad routine (HAP& SAD) before typing in this program.

Background ...

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

Howard had 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 a computer 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."

Highlights ...

Slice data for five, six, seven, and eight pizza slices are loaded into arrays S5, S6, S7, and S8 in lines 80 to 130.

The subroutine beginning on line 3200 draws the pizza pie. The subroutine beginning on line 3000 draws the people.

The seven subroutines to draw the different number of pizza slices (randomly selected, from 2 to 8) begin on lines 4400, 4500, 4600, 4700, 4800, 4900, and 5000.

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 190. The real-time clock variable (-TI) is used to produce a more random distribution of pizza slices.

Lines 600 to 940 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 ...

S5, S6, S7, S8	Array variables of 48, 52, 55, and 44 elements, respectively. Used to store the plot points for 5, 6, 7, and 8 pizza slices.
A\$	Your answer to the question "PLAY AGAIN?"
D	Delay loop counter.
S	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.
L, M	The position for the people and the circle.
X	The column position for the circle and the people.
Y	The row position for the circle and the people.
CT	Controls number of times to answer questions correctly. If $CT > 2$, then the answer and fraction of pizza allotted to each is displayed and the game ends.
I	Loop counter.
TI	The Commodore 64 internal TImmer.

Do-It-Yourself ...

You might experiment with making the pizza pie and the people appear in different colors.

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

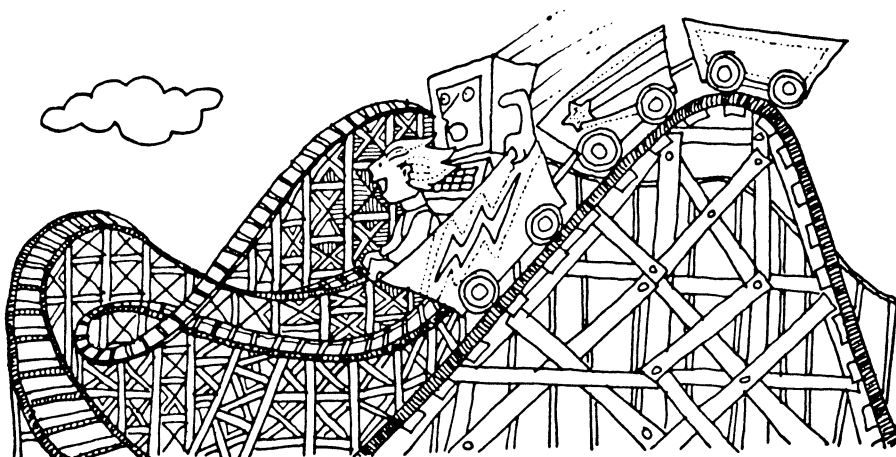
Also, you can show how some of the fractions can be reduced. Sometimes the program (see the subroutine begin-

ning on line 8000) prints a fraction like $2/4$. It would be nice if the program showed that the fraction is equivalent to $1/2$.

Finally, when the child answers the question correctly, you can have the number of pizza slices that go to one person turn a different color.

12

THE ROLLER COASTER



For Parents and Teachers ...

This game helps children learn about the **cosine** function in trigonometry. It draws colorful cosine curves 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 multi-colored roller coaster. The roller coaster looks as if it is standing up on the TV screen.

The Game ...

Program Name: **ROLLER**

```

50 REM *** ROLLER COASTER
55 POKE 53280,3 :REM CYAN BORDER
65 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}" TAB
(9) "*** ROLLER COASTER ***"
70 FOR PAUSE=1 TO 1000: NEXT PAUSE
80 PRINT"{SC}"
90 POKE 53280,12:POKE 53281,15 :REM GRAY S
CREEN WITH DARK GRAY BORDER
100 DIM Y(50)
105 H=1:O=27
110 PRINT"{SC}"
115 FOR X=0 TO 39
➔ 120 R=X*PI/180
125 Y(X)=H*COS(O*R)
130 Z=INT(Y(X))
135 IF Z=Y(X) THEN 155
140 V=Y(X)-Z
145 IF V>=.5 THEN Y(X)=Z+1:GOTO 155
150 Y(X)=Z
155 NEXT X
160 FOR W=-15 TO -4
170 FOR X=0 TO 39
175 Y=Y(X)+W
180 PRINT"{HM}"
190 IF Y=0 THEN 250
200 Y=-Y
210 FOR N=1 TO Y:PRINT "{CD}{CL}":NEXT

```

```
250 ON X+1 GOTO 500,510,520,530,540,550,560,570
260 ON X-7 GOTO 500,510,520,530,540,550,560,570
270 ON X-15 GOTO 500,510,520,530,540,550,560,570
280 ON X-23 GOTO 500,510,520,530,540,550,560,570
290 ON X-30 GOTO 500,510,520,530,540,550,560,570
300 ON X-38 GOTO 500,510,520,530,540,550,560,570
500 PRINT CHR$(5); :GOTO 600 :REM WHITE
510 PRINT CHR$(28); :GOTO 600 :REM RED
520 PRINT CHR$(30); :GOTO 600 :REM GREEN
530 PRINT CHR$(31); :GOTO 600 :REM BLUE
540 PRINT CHR$(144);:GOTO 600 :REM BLACK
550 PRINT CHR$(156);:GOTO 600 :REM PURPLE
560 PRINT CHR$(158);:GOTO 600 :REM YELLOW
570 PRINT CHR$(159);:GOTO 600 :REM CYAN
600 IF H=5 AND W=-5 THEN 620
610 IF W<>-4 THEN 630
620 PRINT SPC(X) "{WH}O":GOTO 650
➤ 630 IF H=5 AND W=-15 AND X=39 THEN PRINT SPC(X) "{RV}";:GOTO 650
➤ 640 PRINT SPC(X) "{RV}"
650 NEXT X
660 IF H=5 AND W=-5 THEN 700 :REM PREVENTS "Y" FROM GOING POSITIVE
670 NEXT W
680 FOR PAUSE=1 TO 2000:NEXT PAUSE
690 H=H+1:IF H<6 THEN 110
700 PRINT "{HM}{CD}{CD}" SPC(4) "ROLLER{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}" SPC(14) "COSINE"
710 POKE 53281,12:POKE 53280,15
720 FOR PAUSE=1 TO 1000:NEXT PAUSE
730 POKE 53281,15:POKE 53280,12
740 FOR PAUSE=1 TO 1000:NEXT PAUSE
750 GOTO 710
```

Typing Hints ...

The character printed on the screen is a special graphics character. Press **G** and "+". The value of π may be accessed from the keyboard by pressing **SHIFT** and the \uparrow (up arrow) keys.

Highlights ...

On line 90, you set the screen background color and border to gray and dark gray, respectively (POKE 53280,12 and POKE 53281,15).

The FOR-NEXT loop from lines 115 to 155 is used to calculate the Y coordinate for each of 40 values of X (X = 0 to 39).

On line 120, the angle R is converted from radian measure to degrees by multiplying each value of X by $\pi/180$, where $\pi = 3.14159$.

On line 125, the array Y(X) is calculated. H is the amplitude of the cosine function and O is its frequency. H also acts as a counter, controlling the number of cosine plots displayed on the TV screen.

Lines 130 to 150 are used to round the calculated values of the cosine prior to actually plotting them.

The FOR-NEXT loop from lines 160 to 670 draws the cosine function. W is used as the Y axis offset. This allows us to draw the roller coaster by plotting multiple cosine functions on the screen at the same time.

All plotting begins from the top of the TV screen (line 180). Line 210 positions the argument. Note that Y is *always* negative—line 200 is used to convert it to a positive value.

The ON-GOTO commands on lines 250 to 300 control the color displayed (lines 500 to 570).

Line 630 prevents the screen from jumping vertically when printing on the bottom line at the right-most character position (character 40).

Line 660 prevents overdrawing at the top of the screen when the cosine amplitude is at its maximum, and the Y axis offset is at a minimum ($H = 5$ and $W = -5$).

Variables ...

PAUSE	Delay loop counter.
H	Cosine function amplitude.
O	Cosine function frequency multiple.
Y(X)	Array—stores 40 values of the cosine function.
Y	Used to hold the array Y(X) plus the Y axis offset W.
W	Y axis offset.
X	Loop counter—also used to control the color of each cosine segment.
N	Loop counter—positions the computed cosine value on the screen.

Do-It-Yourself ...

This program does more than produce video art. It enables you to visualize simple wave functions; 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

$$\text{TAN}(X) = \text{SIN}(X)/\text{COS}(X)$$

$$\text{COT}(X) = \text{COS}(X)/\text{SIN}(X)$$

$$\text{SEC}(X) = 1/\text{COS}(X)$$

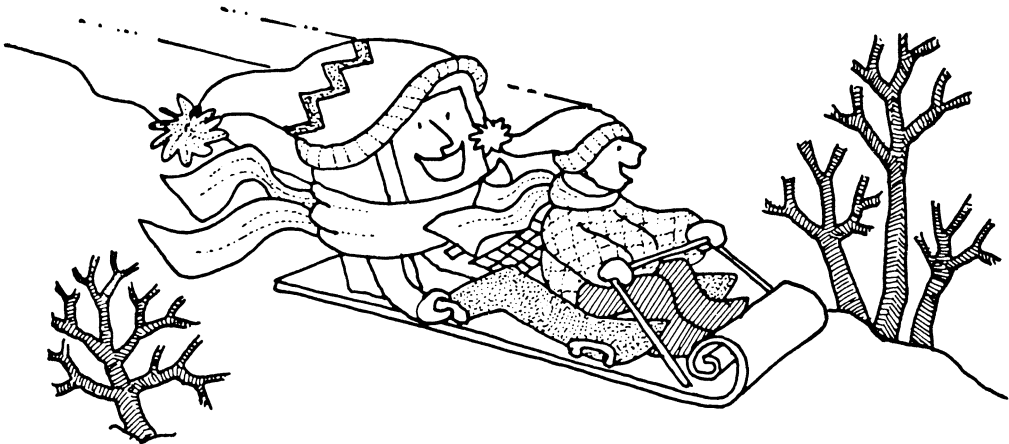
$$\text{CSC}(X) = 1/\text{SIN}(X)$$

To get your child to interact with these functions, you can add INPUT statements that allow variable O to change. The child can enter different values for variable O to see how they affect the roller coaster's shape.

You can add sound commands 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.

13

SLEDS!



For Parents and Teachers ...

This game is similar to The 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 roller coaster. 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
55 POKE 53280,3 :REM CYAN BORDER
65 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}{CD}" TAB
(12) "*** SLEDS! ***"
70 FOR PAUSE=1 TO 1000:NEXT PAUSE
80 PRINT"{SC}"
90 POKE 53280,12:POKE 53281,15 :REM GRAY S
CREEN WITH DARK GRAY BORDER
100 DIM Y(50)
105 H=1:O=40
110 PRINT"{SC}"
115 FOR X=0 TO 39
  ▶ 120 R=X*π/180
    125 Y(X)=H*COS(O*R)
    127 IF Y(X)=0 THEN Y(X)=1/(Y(X)+1):GOTO 13
    0
    128 Y(X)=INT(1/Y(X)) :REM SECANT
    130 Z=INT(Y(X))
    140 IF Z=Y(X) THEN 180
    150 V=Y(X)-Z
    160 IF V>=.5 THEN Y(X)=Z+1:GOTO 180
    170 Y(X)=Z
    180 NEXT X
    190 FOR W=-17 TO -2
    200 FOR X=0 TO 39
    210 Y=Y(X)+W
    220 PRINT"{HM}"
    230 Y=-Y
    240 FOR N=1 TO Y:PRINT "{CD}{CL}":NEXT
    250 ON X+1 GOTO 500,510,520,530,540,550,56
    0,570
    260 ON X-7 GOTO 500,510,520,530,540,550,56
    0,570
    270 ON X-15 GOTO 500,510,520,530,540,550,5
    60,570
    280 ON X-23 GOTO 500,510,520,530,540,550,5
    60,570
```

```

290 ON X-30 GOTO 500,510,520,530,540,550,5
60,570
300 ON X-38 GOTO 500,510,520,530,540,550,5
60,570
500 PRINT CHR$(5); :GOTO 600 :REM WHITE
510 PRINT CHR$(28); :GOTO 600 :REM RED
520 PRINT CHR$(30); :GOTO 600 :REM GREEN
530 PRINT CHR$(31); :GOTO 600 :REM BLUE
540 PRINT CHR$(144);:GOTO 600 :REM BLACK
550 PRINT CHR$(156);:GOTO 600 :REM PURPLE
560 PRINT CHR$(158);:GOTO 600 :REM YELLOW
570 PRINT CHR$(159);:GOTO 600 :REM CYAN
600 IF H=1 AND W=-17 AND X=39 THEN PRINT S
PC(X) "{RV} ";:GOTO 620
610 PRINT SPC(X) "{RV} "
620 NEXT X
630 IF H=1 AND W=-5 THEN W=-2
640 NEXT W
650 FOR PAUSE=1 TO 2000:NEXT PAUSE
660 H=H+1:IF H<4 THEN 110
670 PRINT "{WH}{HM}" TAB(10) "SLEDS! - S
ECANT"
680 POKE 53281,12:POKE 53280,15
690 FOR PAUSE=1 TO 1000:NEXT PAUSE
700 POKE 53281,15:POKE 53280,12
710 FOR PAUSE=1 TO 1000:NEXT PAUSE
720 GOTO 680

```

Highlights ...

This program is similar to the `ROLLER` program. For details take a look at the last chapter. For this example I chose the secant function. To get the secant, you invert the cosine by inverting the array in line 128:

$$\text{SEC}(X) = 1/\text{COS}(X)$$

This is simple—except for one problem. We are using degrees (see the conversion from radians to degrees), 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 ($Y(X)$) equal to 0 on line 127. If the cosine is equal to 0 we add 1 to it. The secant ends up being $1/1$, or 1.

Later, on line 230, we have to convert the value of Y to a positive value. If we don't, we can't plot a point with a negative Y value, and the program does not run correctly.

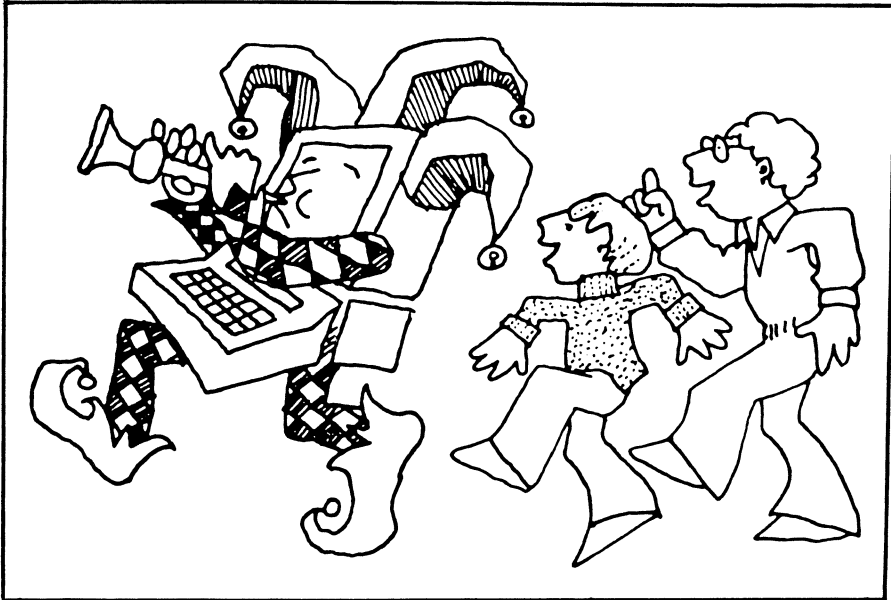
Line 630 prevents overdrawing at the top of the screen when the secant amplitude is maximum and the Y axis offset is -5 .

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.

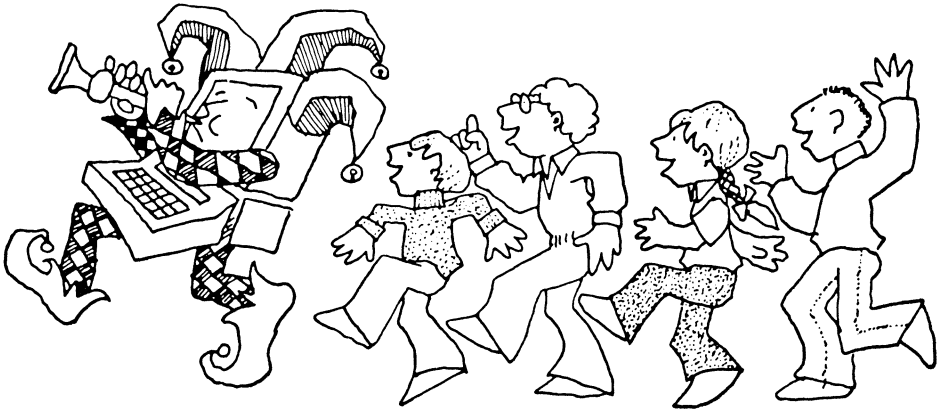


COLORS



14

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: "PLAY 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 purple on the screen and plays a low C note, you press the **5** button.

When the computer flashes the color green and plays a middle C note, you press the **6** button.

When the computer flashes the color blue and plays a high C note, you press the **7** button.

And when the computer flashes yellow and plays a high-high C, you press the **8** button.

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

FLASHES	PURPLE	PURPLE	YELLOW	GREEN
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: 5 - 5 - 8 - 6. The computer will print a happy face. You won!

If you missed one of the colors (say you typed 5 - 5 - 7 - 6), 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"{SC}{CD}{CD}{CD}{CD}{CD}" T
AB(9) "*** THE PIED PIPER ***"
64 FOR PAUSE=1 TO 1000:NEXT PAUSE
65 PRINT"{SC}{CD}{CD}{CD}"
82 HUE(1)=4:HUE(2)=5:HUE(3)=6:HUE(4)=7
85 DIM PICH(8):DIM COMBO(50)
90 PICH(1)=16:PICH(2)=33:PICH(3)=67:PICH
(4)=134
92 PICH(5)=195:PICH(6)=135:PICH(7)=15:PI
CH(8)=30
93 FOR L=54272 TO 54296:POKE L,0:NEXT
95 HI=1
100 FOR SUM=1 TO HI
110 COMBO(SUM)=INT(RND(1)*4)+1
115 POKE 54296,15:POKE 54277,60
120 FOR PLAY=1 TO SUM
130 PRINT"{SC}":POKE 53281,HUE(COMBO(PLA
Y)):POKE 53280,HUE(COMBO(PLAY))
131 POKE 54273,PICH(COMBO(PLAY)):POKE 54
272,PICH(COMBO(PLAY)+4)
132 POKE 54276,17
133 FOR T=1 TO 400:NEXT T:POKE 54276,16:
FOR T=1 TO 50:NEXT
135 NEXT PLAY
137 POKE 53281,12:POKE 53280,12
140 FOR I=1 TO 10:PRINT"{CD}{CL}":NEXT:P
RINT TAB(15)"{WH}YOUR TURN"
145 FOR TRY=1 TO SUM
155 GET A$:IF A$="" THEN 155
160 IF A$=CHR$(13) THEN 155
165 A$=STR$(VAL(A$)-4)

```

```
170 IF VAL(A$)=COMBO(TRY) THEN 178
175 POKE 53281,6:POKE 53280,3:PRINT"{SC}
{PU}":GOSUB 2010:PRINT"{SC}":TRY=SUM:SUM
=HI
176 NEXT:NEXT:GOTO 235
178 PRINT"{SC}"
180 POKE 53281,HUE(COMBO(TRY)):POKE 5328
0,HUE(COMBO(TRY))
182 POKE 54273,PICH(COMBO(TRY)):POKE 542
72,PICH(COMBO(TRY)+4):POKE 54276,17
185 A=PEEK(197):IF A<>64 THEN 185
190 POKE 54276,16
195 POKE 53281,1:POKE 53280,1 :REM WHITE
OUT SCREEN
200 NEXT TRY
205 FOR PAUSE=1 TO 500:NEXT PAUSE
210 NEXT SUM
220 POKE 53281,6:POKE 53280,3:PRINT"{SC}
{LB}":GOSUB 1010:PRINT"{SC}"
235 FOR I=1 TO 10:PRINT"{CD}{CL}":NEXT:P
RINT TAB(9) "AGAIN? (PRESS RETURN)"
237 GET A$:IF A$="" THEN 237
240 IF A$=CHR$(13) THEN HI=HI+1:IF HI<=5
0 THEN 100
260 PRINT"{SC}{PU}":END
```

Typing Hints ...

Remember to LOAD the Happy and Sad routine (HAP& SAD) before typing in this program.

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 (HI = 1). From then on, in each new game the computer adds a new color and tone (on line 240, HI = HI + 1). Line 93 clears the Commodore 64 sound chip.

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. Line 165 is used to subtract 4 from the value of the selected key, translating it from keys 5 to 8 to the numbers 1 to 4, to identify the position in the array HUE in line 82.

If you push the right button (5, 6, 7, 8) 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 not pressing the key, the number in memory location 197 is 64. When you press the key, the number changes.

Variables ...

- COMBO** Array variable—stores the number of the correct key (5, 6, 7, or 8) 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 value associated with each key.
- PICH** Array variable—contains the tone value associated with each key.
- SUM** FOR-NEXT loop counter—the number of colors and tones in the current round of the current game.
- HI** 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).
- L** Loop counter—used to reset the Commodore 64 sound chip.

Do-It-Yourself ...

You might want to change the HI variable to something less miraculous than 50. For example, if you have a five-year-old playing the Pied Piper game, you might set HI (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.

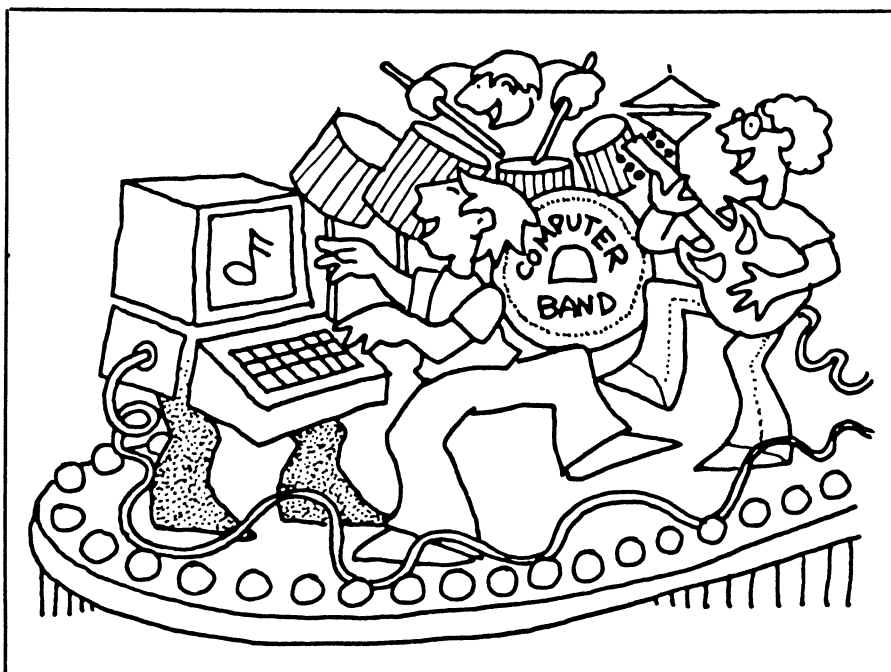
You can also make the game harder. You can create a Super Piper game by adding new tones and colors. Presently the game has only four colors and tones to work with. If you double this number, a child will have to keep track of eight sounds and eight tones strung together in combinations of up to 50 sounds 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 133. 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.

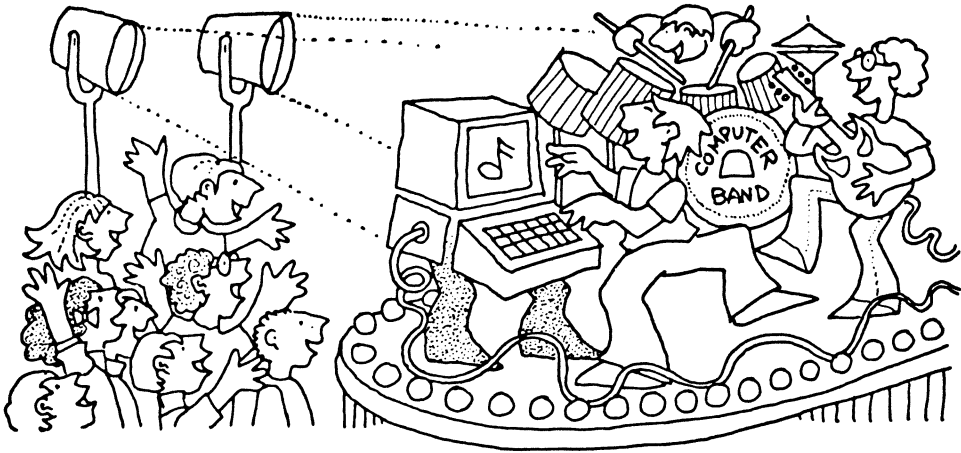
Perhaps you and your children are up to the challenge!

MUSIC



15

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” until you have created all the chords in your song.

When you answer “N” 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.

On lines 5510 to 5540 is a drawing routine that draws a cyan colored message area at the bottom of the screen. In line 5530, 160 reversed spaces (cursors) are drawn (4 screen lines). At the same time, corresponding color memory locations are POKEd with character 3 (cyan).

The Game ...

Program Name: **CHORDS**

```
50 REM CHORDS
55 POKE 53280,3 :REM CYAN BORDER
65 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}" T
AB(7) "*** PLAY-A-CHORD GAME ***"
67 FOR PAUSE=1 TO 1000:NEXT
80 DIM PLAY(2,10),KEY(13),SHARP(13),HIMU
SIC(13),LOMUSIC(13)
```



```
85 FOR L=54272 TO 54296:POKE L,0:NEXT
90 B$=" " :B$=B$+B$
92 GOSUB 3010
94 PRINT"{SC}":POKE 53281,0:POKE 53280,0

95 POKE 54277,0 :POKE 54278,240
96 POKE 54284,0 :POKE 54285,240
97 POKE 54291,0 :POKE 54292,240
98 POKE 54296,15
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 4011
106 FOR PAUSE=1 TO 300:NEXT PAUSE
110 FOR C=1 TO 10:FOR O=0 TO 2:PLAY(O,C)
=0:NEXT:NEXT
117 FOR C=1 TO 10
119 GOSUB 3510
120 FOR O=0 TO 2
140 FOR J=1 TO 13
142 PRINT"{HM}"
144 FOR K=1 TO 8:PRINT"{CD}{CL}":NEXT K:
PRINT TAB(17) " {CU}"
145 IF J<>13 THEN 150
147 PRINT TAB(17) "{YL}HI-CHR$(KEY(J))
148 GOTO 160
150 PRINT TAB(19) "{YL}CHR$(KEY(J))CHR$(
SHARP(J))
160 POKE 54273,HIMUSIC(J):POKE 54272,LOM
USIC(J)
165 POKE 54276,17
170 GET K$
180 IF K$=CHR$(32) THEN 230
190 IF K$<>CHR$(13) THEN 170
200 PLAY(O,C)=J
210 J=13:NEXT J:GOTO 245
230 NEXT J
235 GOTO 140
245 NEXT O
247 IF C>=10 THEN C=10:GOTO 290
250 PRINT"{HM}":FOR K=1 TO 8 :PRINT "{CD
}{CL}":NEXT
255 A$=""
260 PRINT TAB(15)"NEW CHORD?"
265 GET A$
```

```
270 IF A$="Y" THEN PRINT "{CU}"B$:GOTO 2
90
280 IF A$<>"N" THEN 250
283 PRINT "{CU}"B$
285 C=10
290 NEXT C
295 C=10
300 GOSUB 4510
310 GOSUB 4710
330 PRINT"{HM}":FOR K=1 TO 8:PRINT "{CD}
{CL}":NEXT
331 A$=""
332 PRINT TAB(13) "REPEAT CHORDS";:INPUT
A$
335 IF LEFT$(A$,1)="Y" THEN PRINT"{CU}"B
$:GOTO 310
337 IF LEFT$(A$,1)<>"N" THEN 330
340 A$=""
342 PRINT TAB(13) "{CU}PLAY NEW CHORDS";
:INPUT A$
345 IF LEFT$(A$,1)="Y" THEN PRINT"{CU}"B
$:GOTO 110
347 IF LEFT$(A$,1)<>"N" THEN 330
350 POKE 53281,6:POKE 53280,3:PRINT"{SC}
{PU}":END
3000 REM *** LOAD KEYS
3010 FOR I=1 TO 13:READ HIMUSIC(I),LOMUS
IC(I),KEY(I),SHARP(I):NEXT
3020 RETURN
3500 REM *** MESSAGE
3510 PRINT "{HM}":FOR K=1 TO 20:PRINT"{C
D}{CL}":NEXT K
3515 PRINT TAB(7) "{YL}      *** PICK A NOT
E ***
"
3520 RETURN
4000 REM *** PLAY NOTES
4010 GOSUB 5510
4011 PRINT"{HM}":FOR K=1 TO 20:PRINT"{CD
}{CL}":NEXT K
4012 PRINT TAB(7) "{YL}*** LISTEN TO THE
NOTES ***"
4015 FOR J=FIRST TO LAST STEP HOP
4020 POKE 54273,HIMUSIC(J):POKE 54272,LO
MUSIC(J)
```

```

4030 POKE 54276,17
4040 PRINT"{HM}":FOR K=1 TO 8:PRINT"{CD}
{CL}":NEXT K:PRINT TAB(17) " {CU}"
4050 IF J<>13 THEN 4080
4060 PRINT TAB(17) "{YL}HI-"CHR$(KEY(J))
4070 GOTO 4090
4080 PRINT TAB(19) "{YL}"CHR$(KEY(J))CHR$
(SHARP(J))
4090 FOR PAUSE=1 TO 300:NEXT PAUSE
4100 NEXT J
4105 POKE 54276,16
4110 RETURN
4500 REM *** CHORD NOTES
4510 PRINT"{HM}":FOR K=1 TO 20:PRINT"{CD
}{CL}":NEXT K
4512 PRINT TAB(7) "{YL}*** NOTES IN THE C
HORDS ***"
4513 FOR F=1 TO C
4514 IF PLAY(0,F)=0 THEN F=C:GOTO 4610
4515 FOR E=0 TO 2
4520 POKE 54273,HIMUSIC (PLAY(E,F)):POKE
54272,LOMUSIC (PLAY(E,F))
4530 POKE 54276,17
4550 PRINT"{HM}":FOR K=1 TO 8-3*E:PRINT"
{CD}{CL}":NEXT K
4555 IF PLAY(E,F)<>13 THEN 4580
4560 PRINT TAB(19) "{YL}HI-"CHR$(KEY(PLA
Y(E,F)))
4570 GOTO 4590
4580 PRINT TAB(19) "{YL}"CHR$(KEY(PLAY(E
,F)))CHR$(SHARP(PLAY(E,F)))
4590 FOR PAUSE=1 TO 500:NEXT
4592 POKE 54276,16
4595 NEXT E
4600 FOR PAUSE=1 TO 300:NEXT
4603 GOSUB 4650
4605 NEXT F
4610 RETURN
4650 REM *** ERASE SCREEN
4655 FOR I=8 TO 2 STEP-3
4660 PRINT"{HM}":FOR K=1 TO I :PRINT"{CD
}{CL}":NEXT K:PRINT TAB(19) " "
4670 NEXT
4680 RETURN
4700 REM *** PLAY CHORDS

```

```
4710 PRINT "{HM}":FOR K=1 TO 20:PRINT"{C
D}{CL}":NEXT K
4712 PRINT TAB(7) "{YL}*** LISTEN TO THE
CHORDS ***
4713 FOR A=1 TO 3
4715 FOR F=1 TO C
4717 IF PLAY(0,F)=0 THEN F=C:GOTO 4805
4720 POKE 54276,17:POKE 54283,17:POKE 54
290,17
4730 FOR E=0 TO 2
4740 PRINT"{HM}":FOR K=1 TO 8-3*E:PRINT"
{CD}{CL}":NEXT K
4750 IF PLAY(E,F)<>13 THEN 4780
4760 PRINT TAB(19) "{YL}HI-"CHR$(KEY(PLA
Y(E,F)))
4770 GOTO 4795
4780 PRINT TAB(19) "{YL}"CHR$(KEY(PLAY(E
,F)))CHR$(SHARP(PLAY(E,F)))
4795 NEXT E
4796 POKE 54273,HIMUSIC (PLAY(0,F)):POKE
54272,LOMUSIC (PLAY(0,F))
4797 POKE 54280,HIMUSIC (PLAY(1,F)):POKE
54279,LOMUSIC (PLAY(1,F))
4798 POKE 54289,HIMUSIC (PLAY(2,F)):POKE
54288,LOMUSIC (PLAY(2,F))
4799 POKE 54276,17:POKE 54283,17:POKE 54
290,17
4800 FOR PAUSE=1 TO 800:NEXT
4803 POKE 54276,16:POKE 54283,16:POKE 54
290,16
4804 GOSUB 4650
4805 NEXT F
4806 FOR PAUSE=1 TO 400:NEXT PAUSE
4807 NEXT A
4810 RETURN
5000 DATA 16,195,67,32,17,195,67,35,18,2
09,68,32
5010 DATA 19,239,68,35,21,31,69,32,22,96
,70,32
5020 DATA 23,181,70,35,25,30,71,32,26,15
6,71,35,28,49,65,32
5030 DATA 29,223,65,35,31,165,66,32,33,1
35,67,32
5500 REM *** MESSAGE BORDER
5510 SC=1024:BE=40*20:GS=55296
```

```
5520 DR=SC+BE:GR=GS+BE
5530 FOR I=0 TO 159:POKE DR+I,160:POKE G
R+I,3:NEXT
5540 RETURN
```

Highlights ...

Look at the DATA commands on lines 5000 to 5030. Each DATA command has four values: high and low musical tones, a code for the button that selects the note (a “C,” a “D,” an “E,” etc.), and the 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 HIMUSIC and LOMUSIC arrays 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 Chord Notes 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 Chord 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 HIMUSIC and LOMUSIC arrays and the note-name values stored in the KEY array.

- SHARP** Array—stores either a space or a sharp value (#) to go with each musical note displayed on the TV screen.
- KEY** Array—stores 13 letter values representing the note (A through G).
- 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) and, in subroutine beginning at line 4000, controls playing all the notes in the octave.
- F** Loop counter in Chord Notes subroutine—controls playing of each of up to 10 chords you have selected.
- E** Loop counter in Chord Notes 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).
- PAUSE** Delay loop counter.
- HIMUSIC, LOMUSIC**
Arrays—musical tones (middle C through high C).
- FIRST** First note to be played in the loop.
- LAST** Last note played in the loop.
- L** Loop counter—used to clear the Commodore 64 sound chip.
- 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.

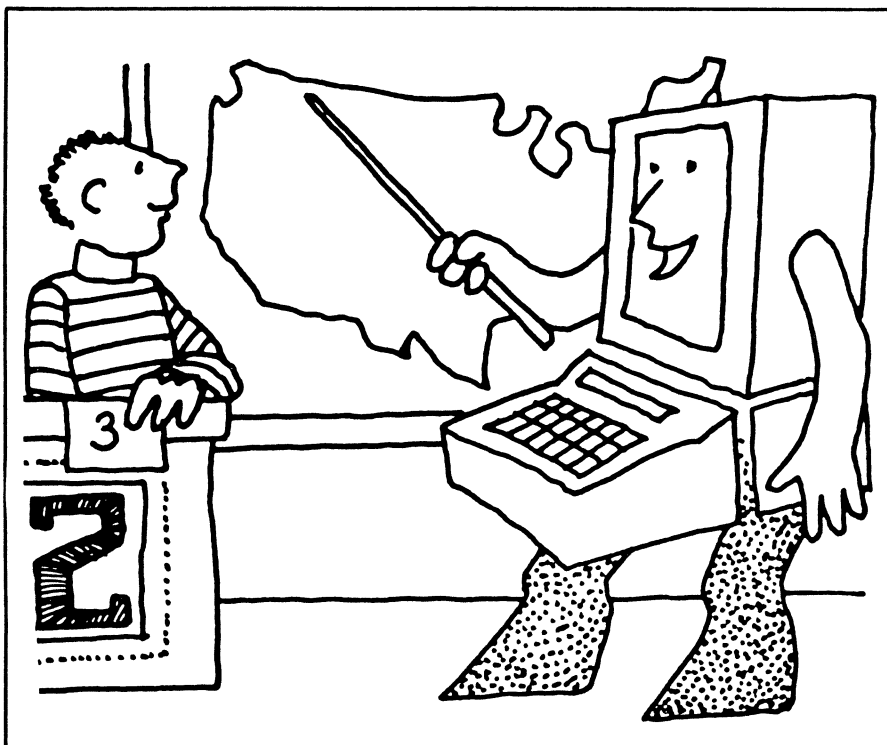
- SC Holds the start of screen character memory.
- GS Holds the start of screen color memory.
- DR Holds the starting screen character memory address to begin drawing the message area.
- GR Holds the starting screen color memory address to begin drawing the message area color.
- BE Pointer to the start of the message area; BE=40*20 means that the start of the message begins after 20 40-character lines.

Do-It-Yourself ...

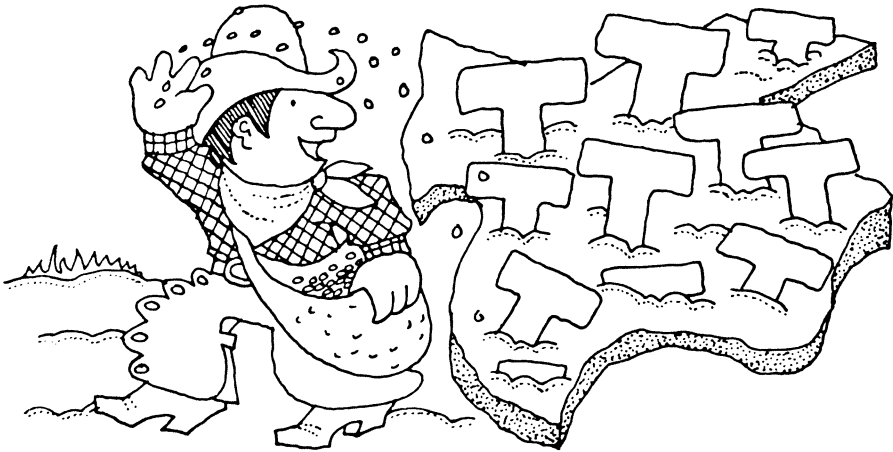
This program can be simplified or be made even more elaborate. For example, you might want to create a simpler game in which children compose a song out of notes instead of chords. Or you can load in a larger number of notes (see page 152 in the *Commodore 64 User's Guide*), so that children can create chords from up to eight octaves on the musical keyboard.

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.

KNOWLEDGE



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**

```

40 REM *** AL'S TOUR OF THE STATES
55 POKE 53280,3 :REM CYAN BORDER
65 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}{CD}" T
AB(4) "*** AL'S TOUR OF THE STATES ***"
66 FOR PAUSE=1 TO 1000:NEXT PAUSE
68 S=1024 :C=55296 :REM SCREEN START
70 FOR X=1 TO 26
75 IF X=2 OR X=5 OR X=10 OR X=17 OR X>23
   THEN 168
80 FOR L=54272 TO 54296:POKE L,0:NEXT
85 PRINT"{SC}"
90 FOR I=1 TO 75
92 POKE 54296,15:POKE 54277,92:POKE 5427
8,136
95 P=INT(RND(1)*1000)
100 POKE S+P,X:POKE C+P,14 :REM LIGHT BL
UE
102 SOUND=INT(RND(1)*128+128)
105 POKE 54273,255-SOUND:POKE 54272,SOUN
D:POKE 54276,33
110 NEXT I
115 POKE 54296,0
120 PRINT "{SC}{CD}{CD}{CD}{CD}{CD}WHAT
STATE IS AL IN";

```

```

130 A$="":INPUT A$
140 IF A$="" THEN GOSUB 3010:GOTO 85
156 RESTORE
157 FOR SE=1 TO 50:READ B$
158 IF LEFT$(A$,1)<LEFT$(B$,1) THEN GOSU
B 2010:SE=50:NEXT:GOTO 80
159 IF B$=A$ AND ASC(LEFT$(A$,1))=X+64 T
HEN GOSUB 1010:SE=50:NEXT:GOTO 168
163 NEXT SE
164 GOSUB 2010:GOTO 80
168 NEXT X
170 PRINT"{SC}":END
3000 REM *** DISPLAY STATES
3010 PRINT TAB(9) "{CD}{CD}STATES BEGINN
ING WITH"
3015 PRINT TAB(9) "{CD}      THE LETTER "
;CHR$(X+64)
➤ 3020 PRINT TAB(9) " _____
":PRINT
3025 RESTORE:FOR NUM=1 TO 50:READ B$
3026 IF ASC(LEFT$(B$,1))=X+64 THEN NUM=5
0
3028 NEXT NUM
3030 PRINT TAB(14) B$
3040 FOR ROW=1 TO 8
3050 READ B$
3060 IF ASC(LEFT$(B$,1))>X+64 THEN ROW=8
:NEXT:GOTO 3100
3070 PRINT TAB(14) B$
3090 NEXT ROW
3100 GET K$:IF K$="" THEN 3100
3110 RETURN
4000 DATA ALABAMA,ALASKA,ARIZONA,ARKANSA
S,CALIFORNIA,COLORADO,CONNECTICUT
4010 DATA DELAWARE,FLORIDA,GEORGIA,HAWAI
I,IDAHO,ILLINOIS,INDIANA,IOWA
4020 DATA KANSAS,KENTUCKY,LOUISIANA,MAIN
E,MARYLAND,MASSACHUSETTS,MICHIGAN
4030 DATA MINNESOTA,MISSISSIPPI,MISSOURI
,MONTANA,NEBRASKA,NEVADA,NEW HAMPSHIRE
4040 DATA NEW JERSEY,NEW MEXICO,NEW YORK
,NORTH CAROLINA,NORTH DAKOTA
4050 DATA OHIO,OKLAHOMA,OREGON,PENNSYLVA
NIA,RHODE ISLAND,SOUTH CAROLINA

```

4060 DATA SOUTH DAKOTA, TENNESSEE, TEXAS, U
TAH, VERMONT
4070 DATA VIRGINIA, WASHINGTON, WEST VIRGI
NIA, WISCONSIN, WYOMING, Z

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 Commodore 64 screen code for that number, the program jumps to line 168 and a new code (the next letter in the alphabet) is chosen.

On line 92, the Commodore 64 sound chip is cleared. On line 105, the sound frequency and the waveform are selected.

On line 120, 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 to 3020, the program prints out a title. Next (on lines 3025 to 3028), the program searches through the state names listed in the DATA statements on lines 4000 to 4070. 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 to 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 and 3110. If you haven't pressed a button yet, line 3100 keeps repeating.

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 4070) on lines 157, 158, and 159. The state named "Z" in line 4070 allows the READ B\$ in line 3050 to always find data.

If there is a match between your answer (A\$) and the state name the computer pulls from the list (B\$), then (on line 159) the computer jumps to the Happy Face subroutine on line 1010.

If the computer can't find a match (on lines 158 or 164), it jumps to the Sad Face subroutine. Remember to LOAD the Happy and Sad Face subroutine HAP&SAD before typing this program.

Variables ...

A\$	Your answer—what state Al is in.
B\$	A state from the computer's list.
X	Counter—the Commodore 64 screen code for the letters in the alphabet.
I	Counter—Al plants 75 letters in each state.
SE	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 a maximum of eight state names at a time (i.e., there are eight states beginning with the letter M).
P	Random number used to position the 75 letters on screen.
SOUND	Random number to produce a sound for each of the 75 letters.

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 168. Hint #2: To compute a number representing a random letter of the alphabet, use the command `INT(RND(1)*26)+1`. This gives you a random number between 1 and 26 (the Commodore 64 screen 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 107 by typing between lines 105 and 110 this line:

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

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 40 and 65. You will need to change the question on line 120. You will need to change the titles on lines 3000 and 3010. You will need to replace the DATA statements on lines 4000 to 4070. 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).

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 CHILDR
EN TO LEARN THEIR ADDRESS
55 POKE 53280,3 :REM CYAN BORDER
65 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}{CD}" T
AB(7) "*** WHERE DO YOU LIVE? ***"
67 FOR PAUSE=1 TO 1000:NEXT PAUSE
92 READ RD$,AP$,CT$,ST$,ZP$
120 PRINT"{SC}"
130 N=6:GOSUB 500
140 PRINT SPC(17)"HELLO!"
145 N=9:GOSUB 500
155 INPUT "WHAT IS YOUR {RV}FIRST{RO} NA
ME";N1$
156 N=9:GOSUB 500
157 PRINT SPC(14) "
"
158 N=9:GOSUB 500
159 PRINT SPC(13) "{RV}LAST{RO} NAME";:I
NPUT N2$
160 N$=N1$
161 N$=N$+" "
162 N$=N$+N2$
164 PRINT"{SC}"
165 N=2:GOSUB 500
166 PRINT "HI,THERE, "N$!"

```



```
169 GOSUB 3010 :REM DISPLAY ADDRESS
170 PRINT "{SC}"
172 RESTORE
173 FOR I=1 TO 5:READ RAD$(I):NEXT I
174 FOR I=1 TO 5:READ ADN$(I):NEXT I
175 OPEN 1,0
176 FOR J=1 TO 5
180 IF RAD$(J)=" " THEN 240
187 PRINT"{SC}"
188 N=6:GOSUB 500
190 PRINT SPC(6) "WHAT IS YOUR " ADN$(J)
"?{CD}"
192 PRINT SPC(6);
194 INPUT#1,GAD$
200 IF GAD$=RAD$(J) THEN PRINT "{SC}":GO
SUB 1010:GOTO 240
210 COUNT=COUNT+1:PRINT "{SC}":GOSUB 201
0:PRINT"{SC}":IF COUNT<3 THEN 187
215 PRINT"{SC}"
220 N=6:GOSUB 500
222 PRINT SPC(6) "YOUR ."ADN$(J)" IS:"
224 N=9:GOSUB 500
225 PRINT SPC(6) RAD$(J)
230 FOR PAUSE=1 TO 3000: NEXT PAUSE
232 COUNT=0
235 GOTO 187
240 NEXT J
245 CLOSE1
250 GOSUB 3010 :REM DISPLAY ADDRESS
260 PRINT"{SC}"
270 N=8:GOSUB 500:PRINT SPC(14) "PLAY AG
AIN";:INPUT A$
280 IF LEFT$(A$,1)="Y" THEN 170
290 IF LEFT$(A$,1)<>"N" THEN 260
300 PRINT"{SC}":END
500 PRINT"{HM}":FOR I=1 TO N:PRINT"{CD}{
CL}":NEXT :REM POSITION PRINT LINE
510 RETURN
3000 REM *** DISPLAY ADDRESS
3010 N=6:GOSUB 500
3020 PRINT SPC(14) "YOUR ADDRESS"
3021 PRINT SPC(14) "-----"
3024 N=10:GOSUB 500
3025 PRINT SPC(11) N$
3030 N=12:GOSUB 500
```

```
3035 PRINT SPC(11) RD$:N=N+2
3037 IF AP$="" THEN 3050
3040 GOSUB 500:PRINT SPC(11) AP$:N=N+2
3050 GOSUB 500:PRINT SPC(11) CT$, "ST$:
N=N+2
3060 GOSUB 500:PRINT SPC(LEN(CT$+ST$)+8)
ZP$
3080 FOR PAUSE=1 TO 4000:NEXT PAUSE
3090 RETURN
5001 DATA 2117 CARTER ROAD SW
5002 DATA 4F
5003 DATA ROANOKE
5004 DATA VIRGINIA
5005 DATA 24015
6001 DATA STREET OR ROAD
6002 DATA APARTMENT
6003 DATA CITY
6004 DATA STATE
6005 DATA ZIP
```

Typing Hints ...

Remember to LOAD the Happy and Sad Face subroutine HAP&SAD before typing this program.

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 5001 to 5005. You will need to replace this with your address. Your street goes on line 5001. Your apartment goes on 5002. If you don't live in an apartment, just leave the empty DATA command at 5002. Your city goes on line 5003, your state at 5004, your zip code at 5005.

The address categories are stored in DATA statements on lines 6001 to 6005 (i.e., STREET OR ROAD, APARTMENT,

CITY, STATE, and ZIP). Using FOR-NEXT loops, the program READs in the address categories and the actual addresses into string arrays on lines 173 and 174. If a category is empty (for example, your apartment address), the program skips to the next category (line 180).

The main action in the program takes place inside the loop from lines 176 to 240. 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.

By opening a channel for the keyboard (OPEN 1,0) in line 175, the INPUT# command (INPUT#1) may be used instead of INPUT. This suppresses the "?" normally associated with the INPUT instruction and allows the "?" to be used with the PRINT instruction instead. The channel is closed in line 245.

Variables ...

PAUSE	Delay loop counter.
N\$	Your name.
N1\$	First name.
N2\$	Last name.
A\$	Answer to question "PLAY AGAIN?"
GAD\$	Address guess.
ADN\$	Array—address category (street, city, etc.).
RAD\$	Array—address (your street name, city name, etc.).

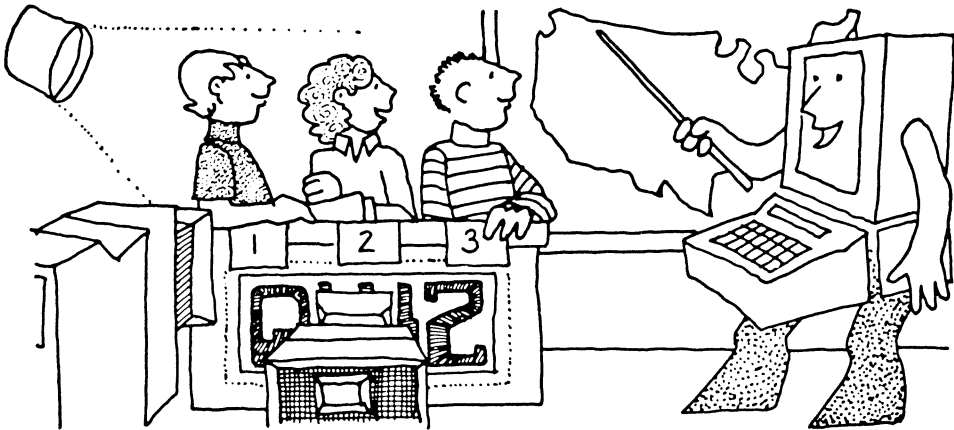
RD\$	Your road.
AP\$	Your apartment.
CT\$	Your city.
ST\$	Your state.
ZP\$	Your zip code.
J	Loop counter—main loop of game.
COUNT	Wrong-answer counter.
N	Row where part of address is displayed.

Do-It-Yourself ...

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

18

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 *** QUIZ SHOW
55 POKE 53280,3 :REM CYAN BORDER
65 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}" T
AB(11) "*** QUIZ SHOW ***"
67 FOR PAUSE=1 TO 1000:NEXT PAUSE
100 GOSUB 800:PRINT TAB(3) "WHO IS THE F
ATHER OF OUR COUNTRY?"
102 PRINT"{CD}{CD}A) GEORGE WASHINGTON":
PRINT "B) JOHN ADAMS"
103 PRINT"C) THOMAS JEFFERSON
104 GOSUB 950
106 PRINT "{SC}":IF A$="A" THEN R=R+1:GO
SUB 1010:GOTO 120
108 GOSUB 900
120 GOSUB 800:PRINT TAB(3) "WHAT IS THE
CAPITAL OF"
121 PRINT TAB(3) "THE UNITED STATES?"
122 PRINT "{CD}{CD}A) NEW YORK":PRINT "B
) WASHINGTON D.C.":PRINT "C) PHILADELPHI
A"
124 GOSUB 950

```

```
126 PRINT "{SC}":IF A$="B" THEN R=R+1:GO
SUB 1010:GOTO 140
128 GOSUB 900
140 GOSUB 800:PRINT TAB(3) "WHO IS THE C
URRENT PRESIDENT OF"
141 PRINT TAB(3) "THE UNITED STATES?
142 PRINT "{CD}{CD}A) JIMMY CARTER":PRIN
T "B) RICHARD NIXON"
143 PRINT "C) RONALD REAGAN"
144 GOSUB 950
146 PRINT "{SC}":IF A$="C" THEN R=R+1:GO
SUB 1010:GOTO 160
148 GOSUB 900
160 GOSUB 800:PRINT TAB(3) "WHICH COUNTR
Y GIVES ITS"
161 PRINT TAB(3) "PEOPLE THE MOST FREEDO
M?"
162 PRINT "{CD}{CD}A) RUSSIA":PRINT "B)
POLAND":PRINT "C) UNITED STATES"
164 GOSUB 950
166 PRINT "{SC}":IF A$="C" THEN R=R+1:GO
SUB 1010:GOTO 180
168 GOSUB 900
180 GOSUB 800:PRINT TAB(3) "WHICH COUNTR
Y WAS THE FIRST"
181 PRINT TAB(3) "TO PUT A MAN IN SPACE?"
"
182 PRINT "{CD}{CD}A) RUSSIA":PRINT "B)
UNITED STATES":PRINT "C) CHINA"
184 GOSUB 950
186 PRINT "{SC}":IF A$="B" THEN R=R+1:GO
SUB 1010:GOTO 200
188 GOSUB 900
200 GOSUB 800:PRINT TAB(3) "WHO DISCOVER
ED ELECTRICITY?"
202 PRINT "{CD}{CD}A) BENJAMIN FRANKLIN"
:PRINT "B) ISAAC NEWTON"
203 PRINT "C) THOMAS JEFFERSON"
204 GOSUB 950
206 PRINT "{SC}":IF A$="A" THEN R=R+1:GO
SUB 1010:GOTO 220
208 GOSUB 900
220 GOSUB 800:PRINT TAB(3) "WHO DISCOVER
ED AMERICA IN 1492?"
```

```
222 PRINT "{CD}{CD}A) AMERIGO VESPUCCI":  
PRINT "B) VIKINGS"  
223 PRINT "C) CHRISTOPHER COLUMBUS"  
224 GOSUB 950  
226 PRINT "{SC}":IF A$="C" THEN R=R+1:GO  
SUB 1010:GOTO 240  
228 GOSUB 900  
240 GOSUB 800:PRINT TAB(3) "WHAT COUNTRY  
WAS THE FIRST"  
241 PRINT TAB(3) "TO PUT A MAN ON THE MO  
ON?"  
242 PRINT "{CD}{CD}A) RUSSIA":PRINT "B)  
UNITED STATES":PRINT "C) GREAT BRITAIN"  
244 GOSUB 950  
246 PRINT "{SC}":IF A$="B" THEN R=R+1:GO  
SUB 1010:GOTO 260  
248 GOSUB 900  
260 GOSUB 800:PRINT TAB(3) "WHAT WAS THE  
FIRST COUNTRY"  
261 PRINT TAB(3) "TO ENTER THE NUCLEAR A  
GE?"  
262 PRINT "{CD}{CD}A) UNITED STATES":PRI  
NT "B) CHINA":PRINT "C) RUSSIA"  
264 GOSUB 950  
266 PRINT "{SC}":IF A$="A" THEN R=R+1:GO  
SUB 1010:GOTO 280  
268 GOSUB 900  
280 GOSUB 800:PRINT TAB(3) "WHAT WAR WON  
THE UNITED STATES"  
281 PRINT TAB(3) "HER FREEDOM?"  
282 PRINT "{CD}{CD}A) WAR OF 1812":PRINT  
"B) REVOLUTION":PRINT "C) CIVIL WAR"  
284 GOSUB 950  
286 PRINT "{SC}":IF A$="B" THEN R=R+1:GO  
SUB 1010:GOTO 300  
288 GOSUB 900  
300 GOSUB 800:PRINT TAB(3) "WHO INVENTED  
THE TELEPHONE?"  
302 PRINT "{CD}{CD}A) THOMAS EDISON":PRI  
NT "B) ALEXANDER GRAHAM BELL"  
304 PRINT "C) BENJAMIN FRANKLIN"  
306 GOSUB 950  
308 PRINT "{SC}":IF A$="B" THEN R=R+1:GO  
SUB 1010:GOTO 320
```



```
310 GOSUB 900
320 GOSUB 800:PRINT TAB(3) "WHEN IS THE
AMERICAN"
321 PRINT TAB(3) "INDEPENDENCE DAY CELEB
RATED?"
322 PRINT "{CD}{CD}A) NOVEMBER 25":PRINT
"B) OCTOBER 31":PRINT "C) JULY 4"
324 GOSUB 950
326 PRINT "{SC}":IF A$="C" THEN R=R+1:GO
SUB 1010:GOTO 340
328 GOSUB 900
340 GOSUB 800:PRINT TAB(3) "WHO WAS THE
PRESIDENT OF"
341 PRINT TAB(3) "THE UNITED STATES AT T
HE"
342 PRINT TAB(3) "BEGINNING OF THE CIVIL
WAR?"
344 PRINT "{CD}{CD}A) ABRAHAM LINCOLN":P
RINT "B) ANDREW JACKSON"
345 PRINT "C) U.S. GRANT"
346 GOSUB 950
347 PRINT "{SC}":IF A$="A" THEN R=R+1:GO
SUB 1010:GOTO 360
348 GOSUB 900
360 GOSUB 800:PRINT TAB(3) "WHICH OF THE
FOLLOWING IS"
361 PRINT TAB(3) "A NUMERIC VARIABLE?"
362 PRINT "{CD}{CD}A) AX$":PRINT "B) Z":
PRINT "C) 45"
364 GOSUB 950
366 PRINT "{SC}":IF A$="B" THEN R=R+1:GO
SUB 1010:GOTO 380
368 GOSUB 900
380 GOSUB 800:PRINT TAB(5) "WHAT TYPE OF
COMPUTER IS THIS?"
382 PRINT "{CD}A) APPLE":PRINT "B) COMMO
DORE":PRINT "C) TIMEX"
384 GOSUB 950
386 PRINT "{SC}":IF A$="B" THEN R=R+1:GO
SUB 1010:GOTO 600
388 GOSUB 900
600 GOSUB 800:PRINT TAB(9) "YOUR SCORE I
S:{CD}{CD}"
610 PRINT TAB(14) R "RIGHT"
```

```
620 PRINT TAB(14) W "WRONG"
630 PRINT:PRINT
640 IF R=15 THEN PRINT TAB(9) "{RV}YOU'R
E A GENIUS":GOTO 700
650 IF R>=11 THEN PRINT TAB(9) "{RV}YOU
ARE PRETTY SMART":GOTO 700
660 IF R>=8 THEN PRINT TAB(9) "{RV}YOU D
ID OKAY":GOTO 700
670 IF R>=4 THEN PRINT TAB(9) "{RV}YOU N
EED A LITTLE WORK":GOTO 700
680 IF R>=1 THEN PRINT TAB(9) "{RV}YOU N
EED TO STUDY MORE":GOTO 700
690 PRINT TAB(9) "{RV}OH! OH! OH!"
700 FOR PAUSE=1 TO 4000:NEXT PAUSE
710 PRINT"{SC}":END
800 PRINT"{SC}{HM}":FOR I=1 TO 4:PRINT"{
CD}{CL}":NEXT:A$=""
810 RETURN
900 W=W+1:GOSUB 2010:RETURN
950 INPUT "{CD} ";A$
960 IF A$<>"A" AND A$<>"B" AND A$<>"C" T
HEN PRINT"{CU} {CU}{CU}":GO
TO 950
970 RETURN
```

Typing Hints ...

Remember to LOAD the Happy and Sad routine (HAP& SAD) before typing in this program.

Background ...

The Quiz Show game was designed by Scott Rainey and Brian Francois, 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 ...

The bracketed codes in the PRINT statement at the beginning of each question (line 800) causes the computer to clear the screen and causes printing to begin from the top line.

There are two counters. The R counter keeps track of correct answers. The W 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).

Line 2100 in the Sad Face subroutine is changed to PRINT TAB(17) "SORRY!" since the program does not permit retries after an incorrect answer.

Variables ...

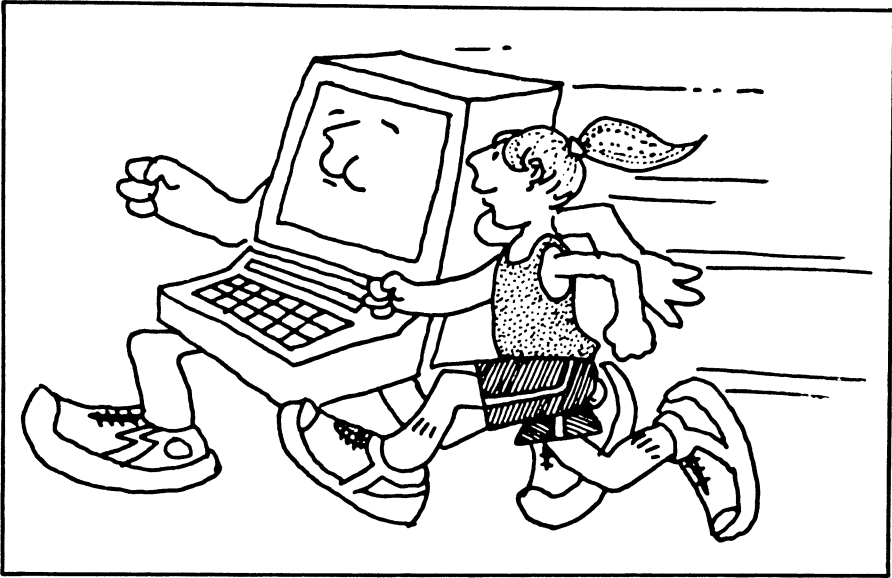
- | | |
|-------|--|
| PAUSE | Delay loop counter. |
| A\$ | Accepts your one-letter answer to quiz show questions. |
| R | Counter for right answers. |
| W | Counter for wrong answers. |

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



19

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 or any other key 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 yellow 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 too simple, change the rules. If you are right-handed, start pressing the **SPACE** bar with your left hand. If you are left-handed, 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?
55 POKE 53280,3 :REM CYAN BORDER
65 PRINT "{SC}{CD}{CD}{CD}{CD}{CD}{CD}"
TAB(7) "*** HOW FAST ARE YOU? ***"
67 FOR PAUSE=1 TO 1000:NEXT PAUSE
70 DIM A$(39)
80 FOR I=1 TO 39:READ A$(I):NEXT :REM RE
AD MESSAGE ARRAY
100 PRINT "{SC}"
110 POKE 53281,0:POKE 53280,0
120 S=19:E=159:C=3:GOSUB 7500 :REM DRAW
BORDER
130 S=5:E=239:C=7:GOSUB 7500 :REM DRAW T
RACK
140 S=20:E=80:C=3:GOSUB 7500 :REM ERASE
MESSAGE IN BORDER
150 POKE 198,0 :REM CLEAR KEYBOARD BUFFE
R CHAR. COUNTER
160 N=19:GOSUB 7000:PRINT SPC(11)"{RV}{W
H}*** GET READY! ***"

```



```
170 N=7:GOSUB 7000
180 FOR PAUSE=1 TO 2000:NEXT PAUSE
190 TS=TI :REM START CLOCK
200 FOR X=0 TO 38:IF PEEK(198)=0 THEN PR
INT "{RV}{YL}~";:FOR I=1 TO 65:NEXT:NEXT
210 IF X<>0 THEN TE=TI:GOTO 240 :REM STO
P CLOCK
220 S=20:E=40:C=3:GOSUB 7500 :REM ERASE
MESSAGE IN BORDER
230 N=19:GOSUB 7000:PRINT SPC(10) "{RV}{
WH}TOO SOON! TRY AGAIN!":GOTO 130
240 TD=(TE-TS)/60 :REM DIFFERENCE IN SEC
ONDS
250 S=20:E=80:C=3:GOSUB 7500 :REM ERASE
MESSAGE IN BORDER
260 N=19:GOSUB 7000:PRINT SPC((40-LEN(A$
(X)))/2) "{RV}{WH}" A$(X)
270 PRINT SPC(12) "{RV}{WH}("INT(TD*100+
.5)/100 "{CL} SECONDS )"
280 FOR PAUSE=1 TO 2000:NEXT PAUSE
290 GOTO 130
5000 DATA ACE!!
5010 DATA PRO!
5020 DATA VERY FAST!
5030 DATA FAST
5040 DATA QUICK
5050 DATA AVERAGE
5060 DATA AVERAGE
5070 DATA AVERAGE
5080 DATA AVERAGE
5090 DATA OKAY
5100 DATA OKAY
5110 DATA OKAY
5120 DATA SLOW
5130 DATA SLOW
5140 DATA VERY SLOW
5150 DATA VERY SLOW
5160 DATA SLOW AS A TURTLE
5170 DATA SLOW AS A SNAIL
5180 DATA SLOW AS A SLOTH
5190 DATA SLOW AS A SLOTH
5200 DATA WAKE UP!
5210 DATA WAKE UP!
5220 DATA WAKE UP!
5230 DATA WAKE UP!
```

```
5240 DATA WAKE UP!
5250 DATA WAKE UP!
5260 DATA WAKE UP!
5270 DATA WAKE UP!
5280 DATA YOU NEED HELP!
5290 DATA YOU NEED HELP!
5300 DATA YOU NEED HELP!
5310 DATA YOU NEED HELP!
5320 DATA YOU NEED HELP!
5330 DATA YOU NEED HELP!
5340 DATA YOU NEED HELP!
5350 DATA YOU NEED HELP!
5360 DATA YOU NEED HELP!
5370 DATA YOU NEED HELP!
5380 DATA IS ANYBODY THERE?
7000 PRINT "{HM}":FOR I=1 TO N:PRINT "{C
D}{CL}":NEXT I
7010 RETURN
7500 SC=1024:BE=40*S:GS=55296
7510 DR=SC+BE:GR=GS+BE
7520 FOR I=0 TO E:POKE DR+I,160:POKE GR+
I,C:NEXT I
7530 RETURN
```

Typing Hints ...

To obtain the graphics character (—) in line 200, press **G** Y.

Highlights ...

On lines 7500 to 7530 is a general drawing routine that draws a colored block starting at any line of the screen. In line 7500, variable S determines the screen line to position a message or erase a line from the screen.

In line 7520, variable E determines the amount of reversed spaces (cursors) to be drawn. At the same time, variable C determines the color of the corresponding color memory addresses.

In line 120, a four-line cyan message border is drawn at the bottom of the screen by setting variables S, E, and C appropriately, and calling the drawing routine at line 7500. Similarly, in line 130, a six-line yellow playing field is drawn in the middle of the TV screen. Again, variables S, E, C, and the drawing routine at line 7500 are used in lines 140, 220, and 250 to erase the message in the message border.

The PRINT command on line 200 draws the computer's line that sprints across the screen. The computer checks for keyboard input before plotting each point on the line by checking to see if there is a key in the keyboard buffer—PEEK (198).

If you press the SPACE bar before the computer starts drawing the sprint line, it tells you (on line 230), "TOO SOON! TRY AGAIN!" Then it starts all over.

The current column of the computer's sprinting line is represented by X. On line 190 the Commodore 64 real-time clock, TI, is stored in TS. This represents the starting time. TI is updated by the computer every 1/60th of a second. If the line starts appearing on the screen, then X does not equal zero on line 210, and the real-time clock, TI, is stored in TE. This represents the ending time. The actual time, TD, is calculated in line 240 in seconds. Line 260 centers one of the appropriate messages from the DATA lines 5000 to 5380 stored in the array A\$. Line 270 prints your reaction time in seconds rounded to two places.

Variables ...

N	Row number on which to print screen message.
TS	Starting time in 1/60th of a second.
TE	Ending time in 1/60th of a second.
TD	Actual time in seconds.
PAUSE	Delay loop counter.

A\$	Message array—computer tells you how well you did.
X	Current column position of the computer's sprint line.
S	Screen line to position message or erase line.
E	Number of characters to print in line 7520.
C	Color of character to print.
SC	Holds the start of screen character memory.
GS	Holds the start of screen color memory.
DR	Holds the starting screen character memory address to begin drawing the message area.
GR	Holds the starting screen color memory address to begin drawing the message area color.
BE	Pointer to the start of the message area; BE=40*19 means that the start of the message begins after 19 40-character lines. Note: The number 19 is supplied by the numeric variable S.

Do-It-Yourself ...

The program always starts the race after it counts to 2000 on line 180. 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 yellow 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



20

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:

"CERO! UNO! DOS! TRES! CUATRO! CINCO! SEIS!
SIETE! OCHO! NUEVE! DIEZ!"

The Game ...

Program Name: **SPANISH**

```

50 REM *** COUNT IN SPANISH
55 POKE 53280,3 :REM CYAN BORDER
60 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}{CD}" T
AB(5) "*** COUNT TO 10 IN SPANISH ***"
67 FOR PAUSE=1 TO 1000:NEXT PAUSE
80 DIM HIMUSIC(10),LOMUSIC(10),N$(10)
90 FOR I=0 TO 10:READ HIMUSIC(I),LOMUSIC
(I):NEXT I
92 FOR I=0 TO 10:READ N$(I):NEXT I
94 FOR L=54272 TO 54296:POKE L,0:NEXT
96 POKE 54277,92
100 PRINT "{SC}"
105 POKE 53281,0:POKE 53280,0 :REM BLACK
BACKGROUND
110 S=20:E=119:C=3:GOSUB 7500 :REM DRAW
BORDER
115 FOR J=1 TO 11
120 S=21:E=39:GOSUB 7500 :REM ERASE MESS
AGE IN BORDER
125 N=20:GOSUB 7000:PRINT SPC(12)"{RV}{W
H} NUMBER (0-10)? ";
127 K$=""
128 GET C$
130 IF C$=CHR$(20) THEN 120
131 IF C$=CHR$(13) THEN IF K$<>"" THEN 1
40
132 IF C$<"0" OR C$>"9" THEN 128

```



```
133 IF LEN(K$)=2 THEN 128
135 K$=K$+C$:PRINT C$;:GOTO 128
140 K=VAL(K$):IF K<0 OR K>10 THEN 120
145 S=21:E=39:GOSUB 7500 :REM ERASE MESS
AGE IN BORDER
160 N=20:GOSUB 7000
165 PRINT SPC(8) "{RV}{CY}SAY IN {WH}ENGL
ISH{CY} & {WH}SPANISH"
170 N=5:GOSUB 7000
175 PRINT SPC(18) "{YL}"K
177 N=10:GOSUB 7000
180 PRINT SPC((40-LEN(N$(K)))/2) N$(K):R
EM CENTER NUMBER
190 POKE 54296,15 :REM MAXIMUM SOUND VOL
UME
195 POKE 54273,HIMUSIC(K):POKE 54272,LOM
USIC(K)
197 POKE 54276,17 :REM TURN ON WAVEFORM
200 FOR PAUSE=1 TO 1800:NEXT PAUSE
210 POKE 54276,0 :REM TURN OFF WAVEFORM
220 N=5:GOSUB 7000
230 PRINT SPC(19) " "
240 N=10:GOSUB 7000
250 PRINT SPC(17) " "
260 NEXT J
270 POKE 53281,6:POKE 53280,3
280 PRINT "{SC}"
290 N=8:GOSUB 7000
295 POKE 54276,0 :REM TURN OFF VOLUME
300 PRINT SPC(17) "{LB}AGAIN";:INPUT A$
310 IF LEFT$(A$,1)="Y" THEN 100
320 IF LEFT$(A$,1)<>"N" THEN 280
330 PRINT "{SC}":END
5000 DATA 16,195,18,209,21,31,22,96,25,3
0
5005 DATA 28,49,31,165,33,135,37,162,42,
62,44,193
5010 DATA CERO,UNO,DOS,TRES,CUATRO,CINCO
,SEIS,SIETE,OCHO,NUEVE,DIEZ
7000 PRINT "{HM}":FOR I=1 TO N:PRINT"{CD
}{CL}":NEXT
7010 RETURN
7500 SC=1024:BE=40*S:GS=55296
7510 DR=SC+BE:GR=GS+BE
```

```
7520 FOR I=0 TO E:POKE DR+I,160:POKE GR+
I,C:NEXT I
7530 RETURN
```

Highlights ...

The musical tones that accompany each number are in the DATA command on lines 5000 and 5005. The Spanish names for the numbers 0 through 10 are in the DATA command on line 5010. The musical tones are read into the HIMUSIC and LOMUSIC arrays on line 90. Spanish numbers are read into the N\$ array on line 92.

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 115 and ends on line 260.

On line 94, the Commodore 64 sound chip is cleared. Line 96 sets the attack/decay value of each note. The sound frequency is set on line 195.

The program INPUTs your chosen numeric digits into the string variable C\$. The commands from lines 130 to 140 check for the INST/DEL key (ASCII 20), the RETURN key (ASCII 13), and characters out of range. This routine also rejects non-numeric entries. Finally, it does not allow the RETURN unless at least one valid digit has been entered, and it limits the number to no more than two digits.

The Spanish names are different lengths. Line 180 centers the name under the number.

On lines 7500 to 7530 is a general drawing routine that draws a colored block starting at any line of the screen. In line 7500, variable S determines the screen line to position a message or erase a line from the screen.

In line 7520, variable E determines the amount of reversed spaces (cursors) to be drawn. At the same time, variable C determines the color of the corresponding color memory addresses.

Variables ...

- S** Screen line to position message or erase line.
- E** Number of characters to print in line 7520.
- PAUSE** Delay loop counter.
- C** Color of character to print
- HIMUSIC, LOMUSIC**
Arrays—store musical tones for each Spanish number name.
- I** Loop counter.
- A\$** Stores your answer to “PLAY AGAIN?”
- J** Loop counter—major loop (controls how many numbers you can choose).
- C\$** Each digit of the number you choose.
- K\$** The number you choose (string).
- K** The numeric value of the number you choose.
- N\$** Array—stores Spanish words for each number from 0 to 10.
- N** Row number to print screen message.
- SC** Holds the start of screen character memory.
- GS** Holds the start of screen color memory.
- DR** Holds the starting screen character memory address to begin drawing the message area.
- GR** Holds the starting screen color memory address to begin drawing the message area color.
- BE** Pointer to the start of the message area; $BE=40*21$ means that the start of the message begins after 21 40-character lines. Note: The number 21 is supplied by the numeric variable S.

Do-It-Yourself ...

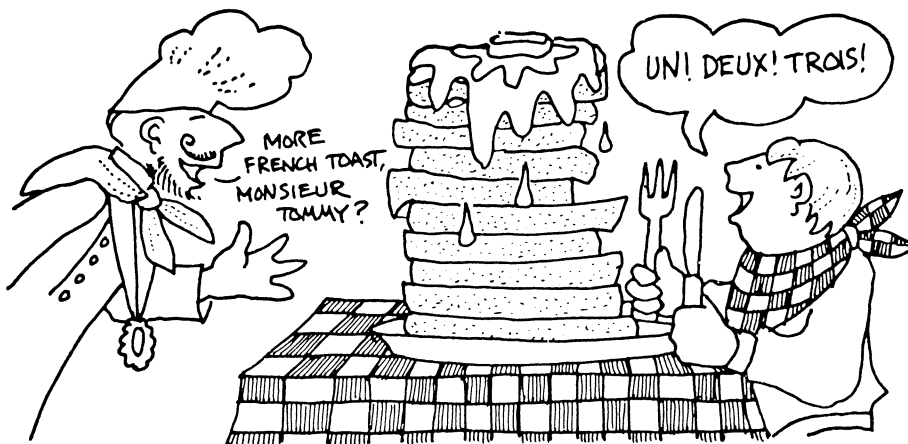
The program only accepts numbers (0, 1, 2, etc.) to indicate which Spanish number name you want to practice. You might modify the program to accept English number names—"zero," "one," "two," and so on. Or, you could have the program accept Spanish number names—"uno," "dos," "tres," etc. This would help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

21

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 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 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
55 POKE 53280,3 :REM CYAN BORDER
60 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}{CD}" T
AB(5) "*** COUNT TO 10 IN FRENCH ***"
67 FOR PAUSE=1 TO 1000:NEXT PAUSE
80 DIM HIMUSIC(10),LOMUSIC(10),N$(10)
90 FOR I=0 TO 10:READ HIMUSIC(I),LOMUSIC
(I):NEXT I
92 FOR I=0 TO 10:READ N$(I):NEXT I
94 FOR L=54272 TO 54296:POKE L,0:NEXT
96 POKE 54277,92
100 PRINT "{SC}"
105 POKE 53281,0:POKE 53280,0 :REM BLACK
BACKGROUND
110 S=20:E=119:C=3:GOSUB 7500 :REM DRAW
BORDER
115 FOR J=1 TO 11
120 S=21:E=39:GOSUB 7500 :REM ERASE MESS
AGE IN BORDER
125 N=20:GOSUB 7000:PRINT SPC(12)"{RV}{W
H} NUMBER (0-10)? ";
127 K$=""
128 GET C$
130 IF C$=CHR$(20) THEN 120
131 IF C$=CHR$(13) THEN IF K$<>"" THEN 1
40
132 IF C$<"0" OR C$>"9" THEN 128
133 IF LEN(K$)=2 THEN 128

```

```
135 K$=K$+C$:PRINT C$;:GOTO 128
140 K=VAL(K$):IF K<0 OR K>10 THEN 120
145 S=21:E=39:GOSUB 7500 :REM ERASE MESS
AGE IN BORDER
160 N=20:GOSUB 7000
165 PRINT SPC(9)"{RV}{CY}SAY IN {WH}ENGL
ISH{CY} & {WH}FRENCH"
170 N=5:GOSUB 7000
175 PRINT SPC(18) "{YL}"K
177 N=10:GOSUB 7000
180 PRINT SPC((40-LEN(N$(K)))/2) N$(K):R
EM CENTER NUMBER
190 POKE 54296,15 :REM MAXIMUM SOUND VOL
UME
195 POKE 54273,HIMUSIC(K):POKE 54272,LOM
USIC(K)
197 POKE 54276,17 :REM TURN ON WAVEFORM
200 FOR PAUSE=1 TO 1800:NEXT PAUSE
210 POKE 54276,0 :REM TURN OFF WAVEFORM
220 N=5:GOSUB 7000
230 PRINT SPC(19) " "
240 N=10:GOSUB 7000
250 PRINT SPC(17) " "
260 NEXT J
270 POKE 53281,6:POKE 53280,3
280 PRINT "{SC}"
290 N=8:GOSUB 7000
295 POKE 54276,0 :REM TURN OFF VOLUME
300 PRINT SPC(17) "{LB}AGAIN";:INPUT A$
310 IF LEFT$(A$,1)="Y" THEN 100
320 IF LEFT$(A$,1)<>"N" THEN 280
330 PRINT "{SC}":END
5000 DATA 16,195,18,209,21,31,22,96,25,3
0
5005 DATA 28,49,31,165,33,135,37,162,42,
62,44,193
5010 DATA NUL,UN,DEUX,TROIS,QUATRE,CINQ,
SIX,SEPT,HUIT,NEUF,DIX
7000 PRINT "{HM}":FOR I=1 TO N:PRINT"{CD
}{CL}":NEXT
7010 RETURN
7500 SC=1024:BE=40*S:GS=55296
7510 DR=SC+BE:GR=GS+BE
```

```
7520 FOR I=0 TO E:POKE DR+I,160:POKE GR+
I,C:NEXT I
7530 RETURN
```

Highlights ...

The musical tones that accompany each number are in the DATA command on lines 5000 and 5005. The French names for the numbers 0 through 10 are in the DATA command on line 5010. The musical tones are read into the HIMUSIC and LOMUSIC arrays on line 90. French numbers are read into the N\$ array on line 92.

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 115 and ends on line 260.

On line 94, the Commodore 64 sound chip is cleared. Line 96 sets the attack/decay value of each note. The sound frequency is set on line 195.

The program INPUTs your chosen numeric digits into the string variable C\$. The commands from lines 130 to 140 check for the INST/DEL key (ASCII 20), the RETURN key (ASCII 13), and characters out of range. This routine also rejects non-numeric entries. Finally, it does not allow the RETURN unless at least one valid digit has been entered, and it limits the number to no more than two digits.

The French names are different lengths. Line 180 centers the name under the number.

On lines 7500 to 7530 is a general drawing routine that draws a colored block starting at any line of the screen. In line 7500, variable S determines the screen line to position a message or erase a line from the screen.

In line 7520, variable E determines the amount of reversed spaces (cursors) to be drawn. At the same time, variable C determines the color of the corresponding color memory addresses.

Variables ...

S	Screen line to position message or erase line.
E	Number of characters to print in line 7520.
PAUSE	Delay loop counter.
C	Color of character to print.
HIMUSIC, LOMUSIC	Arrays—store musical tones for each French number name.
I	Loop counter.
A\$	Stores your answer to “PLAY AGAIN?”
J	Loop counter—major loop (controls how many numbers you can choose).
C\$	Each digit of the number you choose.
K\$	The number you choose (string).
K	The numeric value of the number you choose.
N\$	Array—stores French words for each number from 0 to 10.
N	Row number to print screen message.
SC	Holds the start of screen character memory.
GS	Holds the start of screen color memory.
DR	Holds the starting screen character memory address to begin drawing the message area.
GR	Holds the starting screen color memory address to begin drawing the message area color.
BE	Pointer to the start of the message area; BE=40*21 means that the start of the message begins after 21 40-character lines. Note: The number 21 is supplied by the numeric variable S.

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.

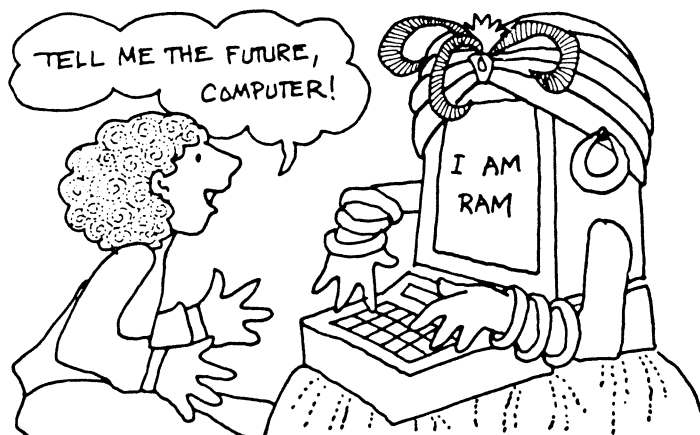
Also, the program could be expanded to French words and French phrases. It could also display the phonetic pronunciation of each of the numbers, words, and phrases to help the children say the words correctly.



IMAGINATION



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
UP TO 79 CHARACTERS LONG.
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
55 POKE 53280,3 :REM CYAN BORDER
60 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}{CD}" T
AB(7) "*** THE FORTUNE TELLER ***"
66 FOR PAUSE=1 TO 1000:NEXT PAUSE
100 PRINT "{SC}"
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 4500:NEXT PAUSE
170 PRINT "{SC}"
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 {RV}YOUR{RO} FUTURE
!"
235 PRINT "          ----"
240 FOR PAUSE=1 TO 4500:NEXT PAUSE
250 PRINT "{SC}"
260 PRINT "ASK ME ANY QUESTION WITH A {R
V}YES{RO} OR {RV}NO{RO}"
265 PRINT
270 PRINT "ANSWER UP TO 79 CHARACTERS LO
NG."
275 PRINT
280 PRINT "WHAT IS YOUR QUESTION?":PRINT
285 OPEN 1,0
290 INPUT#1,M$
295 CLOSE 1
300 PRINT "{SC}";
304 PRINT "I AM GAZING AT MY CRYSTAL BAL
L ..."
305 PRINT
306 PRINT
307 FOR PAUSE=1 TO 1500:NEXT PAUSE
310 FOR I=1 TO 3
320 PRINT TAB(9)·"{RV}GAZING ..."
325 FOR PAUSE=1 TO 600:NEXT PAUSE
330 PRINT
340 NEXT I
345 PRINT
350 PRINT "I SEE YOUR FUTURE!"
360 PRINT
370 FOR PAUSE=1 TO 1200: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 3000:NEXT PAUSE
400 IF INT(RND(1)*2)+1=1 THEN PRINT "{RV
}YES!!";:GOTO 430
410 PRINT "{RV}NO!!";
430 FOR PAUSE=1 TO 2000:NEXT PAUSE
440 PRINT "{SC}"
```



```
450 INPUT "WANT ME TO TELL YOUR FUTURE A  
GAIN";A$  
460 IF LEFT$(A$,1)="Y" THEN 250  
470 IF LEFT$(A$,1)<>"N" THEN 440  
480 PRINT "{SC}":END
```

Highlights ...

This program 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 “WANT ME TO TELL YOUR FUTURE AGAIN?” |
| 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. 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 a sound routine that makes 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.

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 IT!"

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
55 POKE 53280,3 :REM CYAN BORDER
60 PRINT "{SC}{CD}{CD}{CD}{CD}{CD}{CD}"
TAB(7) "*** SECRET AGENT GAME ***"
66 FOR PAUSE=1 TO 1000:NEXT PAUSE
100 PRINT "{SC}"
105 OPEN 1,0
110 PRINT "WHAT IS YOUR CODE NAME?":PRIN
T
115 INPUT#1,N$
120 PRINT "{SC}"
130 PRINT "GOOD NAME! I LIKE IT!"
140 PRINT
145 M$="":M2$=""
150 PRINT N$ ", WHAT IS YOUR SECRET"
155 PRINT:PRINT "MESSAGE?":PRINT:INPUT#1
,M$
160 PRINT "{SC}"

```

```

170 PRINT SPC(13) "{RV}CODING MACHINE"
175 PRINT SPC(14) "{RV}NOW WORKING"
180 FOR PAUSE=1 TO 3000:NEXT PAUSE
190 FOR I=1 TO LEN(M$)
195 IF MID$(M$,I,1)=CHR$(32) THEN M2$=M2
$ + "{SS}":GOTO 210
200 M2$=M2$ + CHR$(ASC(MID$(M$,I,1))+1)
210 NEXT I
220 PRINT "{SC}"
230 PRINT "ORIGINAL MESSAGE:"
240 PRINT
250 PRINT M$
260 PRINT :PRINT :PRINT
270 PRINT "CODED MESSAGE:"
280 PRINT
290 PRINT M2$
300 PRINT :PRINT :PRINT
310 PRINT "NEW MESSAGE, "N$"?":INPUT#1,A
$
320 PRINT "{SC}"
330 IF LEFT$(A$,1)="Y" THEN 145
340 IF LEFT$(A$,1)<>"N" THEN 310
350 CLOSE 1
360 END

```

Highlights ...

On line 105, channel 1 is open for the keyboard (device 0). This suppresses the "?" normally associated with the INPUT instruction. INPUT# is then used in line 115. This allows the "?" to be used with the PRINT instructions in lines 110, 155, and 310. The channel is closed in line 350.

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 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.

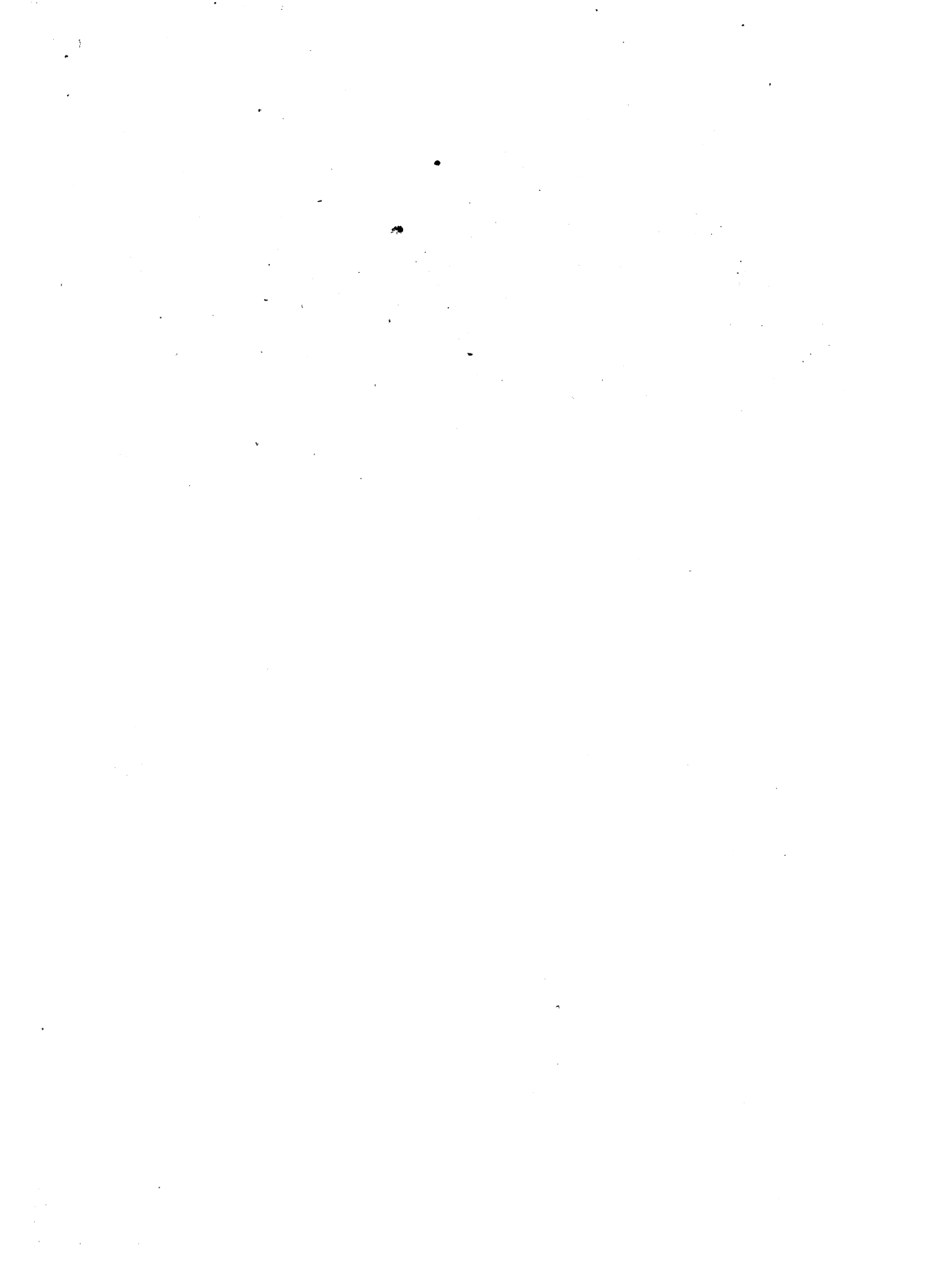
Do-It-Yourself ...

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 ASCII value of MID\$(M\$,I,1), 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.

APPENDIX TO SPECIAL PRINT CHARACTERS

HOW IT APPEARS IN LISTING	KEYS TO PRESS		HOW IT APPEARS ON SCREEN	DEFINITION
{BK}	CTRL	1		BLACK
{WH}	CTRL	2		WHITE
{RD}	CTRL	3		RED
{CY}	CTRL	4		CYAN
{PU}	CTRL	5		PURPLE
{GR}	CTRL	6		GREEN
{BL}	CTRL	7		BLUE
{YL}	CTRL	8		YELLOW
{RV}	CTRL	9		REVERSE ON
{RO}	CTRL	0		REVERSE OFF
{SC}	SHIFT	CLR HOME		CLEAR THE SCREEN AND HOME
{CD}		↑ CRSR ↓		CURSOR DOWN
{CU}	SHIFT	↑ CRSR ↓		CURSOR UP
{CR}		← CRSR →		CURSOR RIGHT
{CL}	SHIFT	← CRSR →		CURSOR LEFT
{HM}		CLR HOME		HOME



COMMODORE 64™ IN Wonderland

Fred D'Ignazio

A simple, entertaining introduction to the world of the Commodore 64 for youngsters. Learn how to write a book report, learn angle measure while "riding" a roller coaster, create songs, test reflexes, appear on a quiz show, and learn to count in French and Spanish. Twenty-three 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 Commodore 64 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 Commodore 64.

More adventures by Fred D'Ignazio ...

THE COMMODORE 64™ PLAYGROUND

Written in the same exciting format as *Commodore 64 in Wonderland*, *The Commodore 64 Playground* guides young children on a series of twenty-four adventures that will enhance their word and number skills. Children participate in a spelling bee, draw with a computer crayon, chase wild letters, watch ghosts appear and disappear, and play games against the Commodore 64. #6307-5, paper, 150 pages.



HAYDEN BOOK COMPANY
a division of Hayden Publishing Company, Inc.
Hasbrouck Heights, New Jersey

ISBN 0-8104-6308-3