

Commodore 64 BASIC Programs

Timothy Orr Knight & Darren LaBatt





Commodore 64 BASIC Programs



Timothy Orr Knight — At 13, Tim Knight started “playing around” with computers and “fell in love” with them. While attending high school, Tim has been busy writing reviews, articles, and programs for national computer magazines. This busy 17-year-old enjoys being with friends and being involved in school activities. He plans to write at least a dozen more books before starting to college. In the summer of 1983 he formed his own software company to produce both business and game software for a wide variety of computers. He is the author of the SAMS books *The World Connection* and *Megabucks From Your Microcomputer*.



Darren LaBatt has become an enthusiastic computerist in less than a year. His first and only computer has been the Commodore 64. He recently mastered machine language. He plans to use this book as a text in the computer class which he teaches in the elementary school he formerly attended.

Commodore 64 BASIC Programs

by
Tim Knight
and
Darren LaBatt

Howard W. Sams & Co., Inc.
4300 WEST 62ND ST. INDIANAPOLIS, INDIANA 46268 USA

Copyright © 1983 by Timothy Orr Knight and
Darren LaBatt

FIRST EDITION
FIRST PRINTING — 1983

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. While every precaution has been taken in the preparation of this book, the publisher assumes no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein.

International Standard Book Number: 0-672-22171-3
Library of Congress Catalog Card Number: 83-50832

Edited by: *C. Herbert Feltner*

Printed in the United States of America.

Preface

This is a book created for the Commodore 64 user interested in learning more about how to use and program his or her computer. Although the BASIC programs themselves are useful and entertaining, this book was written for more than simply typing programs into a Commodore 64 and running them.

Because the greatest knowledge comes from experience, we have made the information in this book versatile enough for you to use as you experiment with these programs and learn more about your computer from your experience with it. The Commodore 64 has a tremendous amount of potential to offer, and its power should be used to educate the user, as well as to serve a useful purpose.

Each program has an introduction that explains what the program does and how the program is used. In addition, the variables and lines of each BASIC program are explained, both to help you understand how the program works and also to let you modify a program in any manner you desire.

We do hope that you enjoy the wide variety of programs within this book, but we want even more for you to *learn* from these programs and from your experimentation with them. The 64 is a powerful and versatile computer, and you should feel good about yourself for wanting to know more about the machine and how you can use it for your benefit.

TIMOTHY ORR KNIGHT
DARREN LABATT

This work is dedicated to Mr. Donald A. Black, whose insistence on our working and thinking like good writers made this book possible. To him we are forever grateful.

Contents

CHAPTER 1

AN INTRODUCTION TO THE PROGRAMS	9
The Format – A Review of the Commodore 64	

CHAPTER 2

GRAPHICS OF THE COMMODORE 64	15
Program: Micro Sketch – Program: Sprinput – Program: Patterns – Program: The Art of Boxes – Program: Scrambler	

CHAPTER 3

SOUNDS GALORE ON THE SIXTY-FOUR	25
Program: Our First Song – Program: Computer Lullaby – Pro- gram: Piano Machine – Program: Sounds of Dixie – Program: Bach Minuet	

CHAPTER 4

GAMES	35
Program: Jelly Beans – Program: Rocket Race – Program: Dice Games – Program: Simon Says – Program: 21 (Blackjack)	

CHAPTER 5

EDUCATIONAL PROGRAMS	57
Program: History – Program: Places & Capitals – Program: Metric Converter – Program: French Tutor – Program: Geometry	

CHAPTER 6

THE COMPUTER TUTOR	73
Program: Computer Introduction – Program: Computer Tutor 1 – Program: Computer Tutor 2	

CHAPTER 7

PROGRAMS FOR THE HOME	91
Program: Checkbook – Program: Recipe Converter – Program: Mortgage Machine	

CHAPTER 8

UTILITY PROGRAMS	99
Program: Sound Developer – Program: Sprite Coder – Program: Music Machine	

CHAPTER 9

THE CREATION OF A PROGRAM	109
Idea/Planning – Writing a Program – Debugging and Finishing – You and Your Computer	

Chapter 1

An Introduction to the Programs

The Commodore 64 is one of the most powerful, exciting, and economical computers ever to be introduced to the personal computer market. It has the power to display graphics in vivid colors and resolution, generate sounds and music of amazing quality, and do most of the other great things that a good home computer can do.

Many people seem to be aware of these facts, since the Commodore 64 is also one of the best-selling computers today. Computer stores can't seem to get these machines in fast enough to meet the demand of the computer-buying public. However, it appears that a lot of people purchasing the Commodore 64 aren't quite sure what to do with it after the computer is ready to run. In fact, not many people learn to unleash the great power within the computer without a lot of trial and error. This book should solve that problem for Commodore 64 users, though. This is a book of programs in BASIC for the Commodore 64.

The purpose of the book is not a singular one. There are several ways this book may be used to benefit anyone using a Commodore

64, especially if that person is just learning about computers. The three main purposes of this book are:

1. To give a Commodore 64 user a total of 30 ready-to-run programs in the BASIC language to use.
2. To allow a person to see how the program works so that a greater knowledge of the computer may be attained.
3. To suggest certain modifications to the program that the computer user may try on his own. These modifications provide further education since modifying a program is excellent experience for program development and realizing the power of the Commodore 64.

This book contains 30 programs, and all include comments so that a person typing in a program, or even a person examining the finished program, may easily determine how the program works. By building, using, and modifying the programs contained in this book, nearly anyone may gain a greater knowledge of the Commodore 64 computer as well as the BASIC language.

THE FORMAT

To make using this book easy and efficient, a definite format has been set up for each of the 30 programs. First, the programs have been divided into seven "groups" (chapters) including programs for graphics, programs for sounds, and even programs to teach a person more about computers. If you have a particular interest in one subject, you might want to go directly to that chapter. The programs do *not* get more difficult as the book progresses, so there's no need to follow the order of the chapters if you have an interest in one particular type of program.

The format for each of the programs consists of the following five parts:

1. An introduction to the program, explaining the program's purpose, uses, and what special features of the Commodore 64 are used in the program.
2. A description of how the program works, including technical information on certain "POKEs", "PEEKs", and other functions being used. For example, if the high-resolution graphics characters called SPRITEs are used in a program, a description of how the SPRITEs are created and manipulated, on a

technical level, is given. More advanced programmers should pay close attention to the workings of a program, while novice users should concentrate on the program first and the technical information about the program later.

3. The program itself — every program has an abundance of REMark statements so that a person typing in a computer program and anyone who examines the program later may see what most of the lines in the program are for. The programs are ready to type in and ready to run, since they are all made exclusively for the Commodore 64.
4. A sample run of the program, so you can see what the program should actually “run” like when it is completely typed into the computer. Some programs, such as graphics programs, are represented by photographs so you may see what the final program should do, while programs strictly involving text (letters and numbers) are written out in the same way they should appear on your computer video screen.
5. Suggested changes and projects — many people, after typing in and running a program, feel an urge to make a few changes of their own, some of which may change the nature of a program entirely. To help you get started, there are several suggestions for modifications in the programs that you might want to try. Changing and even improving programs is an educational experience as well as an exciting one.

A REVIEW OF THE COMMODORE 64

To make sure all of the terms used later in this book are clear and understandable, here is a short review to acquaint you with the computer itself as well as some of the words used throughout this book.

Throughout this book, reversed letters (e.g., **RETURN**) designate keystrokes. That is, when this designation is used, it means the reversed letters are a single keystroke on the keyboard (or keys that are pressed simultaneously to produce a desired result) instead of being input as individual characters.

One of the most important parts of the computer is the keyboard. The keyboard looks similar to an electric typewriter. There are some important differences, however; between the computer's



keyboard and a regular typewriter. First, the **RETURN** key serves as the equivalent of a carriage return on a regular typewriter. Whenever you finish typing in a program line, you must hit the **RETURN** key for the computer to put the line into its memory. Also, there are two **CRSR** keys located at the bottom right of the main computer keyboard. These are for moving the "cursor," or flashing box, around the computer's video monitor. If you want to go up, down, left, or right, perhaps to different parts of a program to change the contents of lines, you should use the cursor keys. The **CTRL** key stands for CONTROL and may be used for a variety of purposes, including changing the color of the letters you type into the computer. The **RUN/STOP** key is used to stop a program in action, while the **SHIFT/LOCK** button is the equivalent of holding down the **SHIFT** key constantly. The strange-looking key with the Commodore trademark on it is used for typing certain graphics characters onto the video screen. The **CLR/HOME** key is primarily used to clear the video screen of everything. If you ever want to clear the screen, simply press the **SHIFT** key and the **CLR/HOME** key at the same time. The **INST/DEL** key is for inserting and deleting letters and numbers into your BASIC programs, as far as this book is concerned. Simply press this key if you want to delete (get rid of) a character on the video screen, or press **SHIFT** and this key if you would like to insert a character. The **RESTORE** key may be used in conjunction with the

RUN/STOP key to reset everything in the computer to a "normal" state, as if you had just turned the computer on. The only change would be that anything in the computer's memory would be left there, rather than erased, which is what would happen if you turned the computer off and then on again.

The video monitor on the Commodore 64, commonly called the "screen," is where the letters, numbers, and graphics are displayed. Since the Commodore is capable of producing a wide variety of colors and graphics, a color monitor is almost a necessity for the Commodore user. The color monitor sold by Commodore is an excellent buy and supports all of the Commodore's powerful features. The screen itself displays 40 columns by 25 rows for a total of 1000 places to put a character onto the screen. The monitor also displays 16 vivid colors, and is also capable of showing high-resolution graphics of just about anything a programmer wants displayed.

The input/output device on the Commodore 64 is usually a disk drive or a cassette recorder. Both perform the same function: storing data onto magnetic media. However, the disk saves information on a diskette while a cassette stores information on (you guessed it) a cassette tape. The important difference between the two is that a disk drive saves information much more quickly and efficiently than a cassette recorder. If you have the extra cash to buy a disk drive, you would be wise to purchase one, since a disk drive makes computing much faster and easier. Regardless of the media you use, the input/output device can save the programs you type into the computer for future use. It is highly recommended that you save each of these programs you type in, since retyping a program every time you want to use it becomes somewhat of a bother.

These, then, are the three important elements of the Commodore 64 system. There really aren't any other words you need to know to use this book to its full potential, since the definitions of RAM, KILOBYTE, and NIBBLE have more to do with a computer salesperson's talk than this book.

Now that we have cleared the way for you to begin working with these programs, there's nothing stopping us from beginning our adventure with the Commodore 64. Let's start our work and see what kind of power and excitement we can find hidden underneath the keyboard of this marvelous computer.



Chapter 2

Graphics of the Commodore 64

Certainly one of the most powerful features of the Commodore 64 is graphics. The graphics of this computer rival the best in the home computer field, and the programs within this chapter will show you how to use these graphics to even greater lengths than you already have.

PROGRAM: MICRO SKETCH

This is a simple low-resolution graphics program that allows you to move the cursor around the screen, using the Y, B, G, and H keys for up, down, left, and right movements, respectively (see Listing 2-1). As you move the cursor around the screen, a trail will be left against the black background, which allows you to draw any shape you desire on the screen.

Explanation of Lines Within Program

1-40 Specify VIC chip starting point, black out and clear screen.
50-67 Establish location of cursor and jump to line 150 to display location of cursor.

Listing 2-1. Micro Sketch

```

5 S=53248
7 POKE S+33,0
10 REM LOW-RESOLUTION MICRO SKETCH
20 REM BY TIM KNIGHT
30 REM CLEAR SCREEN
40 PRINT"?"
50 REM ESTABLISH LOCATION
60 L=55296
67 GOTO 150
70 GETA$:IF A$="" THEN 70
80 IF(A$="Y")AND(L>55335) THEN :L=L-40:GOTO 150
90 IF (A$="B") AND (L<56256) THEN L=L+40: GOTO 150
100 IF (A$="G") AND (L>55296) THEN L=L-1:GOTO 150
110 IF (A$="H") AND (L<56295) THEN L=L+1:GOTO 150
120 REM POKE IN WHITE CHARACTER
150 POKE L,1:POKE L-54272,102
160 GOTO 70

```

70-110 Get keyboard input and change location of cursor accordingly.

120-160 Put white character in cursor's trail and go back to line 70.

Important Variables in Program

S = VIC chip starting address

L = Location of cursor in color memory

A\$ = Keyboard input

Special Note: The VIC chip is simply the graphics chip that controls most of the functions to modify the video screen. In this case, it is used to fill the screen with black.

Sample Run of the Program

A sample run of the Micro Sketch program is shown in Fig. 2-1.

Suggested Changes and Projects

1. Try making a high-resolution version of this program.
2. See if you can make other user-definable features to this program, such as definable cursor color and background color.

PROGRAM: SPRINPUT

This program will allow you to enter data for a sprite of your own and will poke the sprite data into memory and display the final product on the screen (Listing 2-2). This is very helpful when you are trying to perfect a sprite and don't want to go through the tedium of making a subroutine to POKE your sprite data into the computer's memory and display the sprite on the screen.



Fig. 2-1. Sample run of Micro Sketch program.

Listing 2-2. Sprinput

```

1 REM DIMENSION DATA STATEMENT
5 DIM D(20,3)
10 REM SPRINPUT PROGRAM
20 REM BY TIM KNIGHT
30 PRINT":": REM CLEAR SCREEN
35 REM INSTRUCTIONS
40 PRINTTAB(14)"SPRINPUT"
50 PRINT PRINT" THIS PROGRAM WILL ALLOW YOU TO "
60 PRINT"ENTER DATA STATEMENTS WITH ERASE FOR"
70 PRINT"A SPRITE OF YOUR OWN. IT MAY BE VERY"
80 PRINT"HELPFUL FOR YOU TO USE THE 'SPRITE"
90 PRINT"CODER PROGRAM' IN CONJUNCTION WITH THIS"
100 PRINT"PROGRAM TO MAKE THE CREATION OF YOUR"
110 PRINT"SPRITES AND THEIR DATA STATEMENTS "
120 PRINT"SOMEWHAT EASIER. PLEASE BEGIN "
130 PRINT"ENTERING YOUR DATA STATEMENTS NOW":PRINT
133 REM SET UP BORDER AND SCREEN COLOR
135 POKE 53280,5:POKE 53281,6
136 V=53248:POKE V+34,3: REM BACKGROUND COLOR #1
137 POKE 53269,4:POKE 2042,13: REM ENABLE SPRITE
138 REM ENTER ALL DATA
140 FOR J= 0 TO 20:FOR I = 1 TO 3
150 PRINT"ENTER COLUMN";I;":", ROW";J
160 INPUT D
165 REM MAKE SURE DATA IS VALID
170 IF D<0 OR D>255 THEN PRINT"INVALID DATA!":GOTO 150
175 REM SET ARRAY VARIABLE EQUAL TO DATA
180 D(J,I)=D
183 REM CALCULATE WHERE TO POKE SPRITE DATA
185 POKE 831+(J*3)+I,D
190 NEXT: NEXT
195 REM DISPLAY SPRITE AT (200,100)
200 POKE 53252,200:POKE 53253,100

```

Explanation of Line Numbers Within Program

- 1-130 Dimension array D and give instructions for program.
- 133-136 Set up border color and screen color.
- 137 Enables sprite.
- 138-160 INPUT data.
- 170 Makes sure data is all right.
- 180 Sets array variable equal to input data.
- 185 POKEs piece of data into memory.
- 190 Continues loop until through.
- 200 Displays the sprite at location (200,100).

Important Variables in Program

D(20,3) = Array to store data

V = VIC graphics chip

I = Column

J = Row

D = Data input by user

Special Note: The memory locations 832-894 are used to store the sprite.

Sample Run of Part of the Program

```
ENTER COLUMN 1, ROW 0? 23
ENTER COLUMN 1, ROW 1? 333
INVALID DATA!
ENTER COLUMN 1, ROW 1?
```

Suggested Modifications and Projects

1. Try making a program that will allow you to modify some of the data you have entered so that you can modify a sprite without having to type in all of your data again.
2. See if you can make the finished sprite move across the screen when it is displayed.

PROGRAM: PATTERNS

Patterns is a graphics demonstration program to show off the colors and graphics of your Commodore 64 (Listing 2-3). Just type it in and let it run!

Listing 2-3. Patterns

```

1 REM SPECIFY GRAPHIC CHIP STARTING POINT
5 S=53248
6 REM BLACK OUT SCREEN
7 POKE S+33,0
10 REM PATTERNS
20 REM BY TIM KNIGHT
30 REM CLEAR SCREEN
40 PRINT"J"
50 REM ESTABLISH LOCATION (MIDDLE SCREEN)
60 L=55796
65 REM GET KEYBOARD INPUT
67 CH=101:GOTO 150
70 A=INT(RND(0)*7)+1
75 REM GO UP
80 IF(A=1)AND(L>55335)THEN :L=L-40:CH=92:GOTO 150
85 REM GO DOWN
90 IF (A=2) AND (L<56256) THEN L=L+40:CH=127:GOTO 150
95 REM GO LEFT
100 IF (A=3) AND (L>55296) THEN L=L-1:CH=102:GOTO 150
105 REM GO RIGHT
110 IF (A=4) AND (L<56295) THEN L=L+1:CH=81:GOTO 150
115 REM GO UP/RIGHT
120 IF (A=5) AND (L>55335) THEN L=L-39:CH=105:GOTO 150
125 REM GO UP/LEFT
130 IF (A=6) AND (L>55337) THEN L=L-41:CH=95:GOTO 150
135 REM GO DOWN/LEFT
140 IF (A=7) AND (L<56256) THEN L=L+39:CH=105:GOTO 150
141 REM GO DOWN/RIGHT
142 IF (A=8) AND (L<56258) THEN L=L+41:CH=95:GOTO 150
149 REM POKE IN CHARACTER
150 POKE L,INT(RND(1)*15):POKE L-54272,CH
152 REM CHANGE ALL COLORS
153 FOR XX=32 TO 36:POKE S+XX,INT(RND(0)*15) NEXT
155 REM REPEAT PROCESS
160 GOTO 70

```

Explanation of Lines Within Program

- 5 Specifies where VIC chip starts for graphics.
- 7 Blacks screen out.
- 40-60 Clear screen and make location of cursor middle of screen.
- 67 Makes character code equal to 101 and jump to line 150.
- 70 Chooses random direction.
- 80-142 Make appropriate variable changes for direction chosen and assign character code to variable CH.
- 150 Puts character and color in their place.
- 153 Changes all of the colors for impressive display.
- 160 Loops back to line 70.

Important Variables in Program

S = VIC chip start address

L = Location of cursor that draws pattern on screen

CH = Character drawn onto screen
A = Random number to establish direction
XX = For/next loop variable

Sample Run of Program

A sample run of the Patterns program is shown in Fig. 2-2.

Suggested Modifications and Projects

1. Change the color patterns so that only certain kinds of colors appear on the screen, rather than all 16 colors.
2. Try different graphics characters and see which ones look best when the pattern is traveling in a certain direction.



Fig. 2-2. Sample run of Patterns program.

PROGRAM: THE ART OF BOXES

This is also a simple demonstration program to please the eyes and increase your knowledge about the graphics of the Commodore 64 (Listing 2-4).

Listing 2-4. The Art of Boxes

```
10 REM THE ART OF BOXES
20 REM BY TIM KNIGHT
25 REM CLEAR SCREEN
30 PRINT"J"
35 REM BEGINNING OF GRAPHICS CHIP
40 S=53248
```

```
45 REM MAKE SCREEN ALL BLACK
50 POKE S+33,0:POKE S+32,0
55 REM GET RANDOM NUMBERS FOR BOXES
60 X1=INT(RND(1)*40):Y1=INT(RND(1)*25)
61 C=INT(RND(8)*15)
62 REM RANDOM COLOR=C
65 REM GET MORE RANDOM NUMBERS FOR BOXES
70 X2=INT(RND(1)*40):Y2=INT(RND(1)*25)
73 REM MAKE SURE RANDOM NUMBERS ARE OK
75 IF X1=X2 OR Y1=Y2 THEN 60
77 REM IF NUMBERS ARE OK, PROCEED...
80 IF X1<X2 THEN 100
85 REM EXCHANGE IN CASE NUMBERS ARE BACKWARDS
90 X3=X1:X1=X2:X2=X3
95 REM DO X-COORDINATE CHARACTERS
100 FOR X=X1 TO X2:POKE 1024+X+(Y1*40),121:POKE 1024+X+(Y2*40),120
101 REM FILL IN X-COORDINATE COLORS
105 POKE 55296+X+(Y1*40),C:POKE 55296+X+(Y2*40),C:NEXT
110 REM IF NUMBERS ARE OK, PROCEED
120 IF Y1<Y2 THEN 140
125 REM EXCHANGE IN CASE NUMBERS ARE BACKWARDS
130 Y3=Y1:Y1=Y2:Y2=Y3
135 REM DO Y-COORDINATE CHARACTERS
140 FOR Y=Y1 TO Y2:POKE 1024+X1+(Y*40),118:POKE 1024+X2+(Y*40),117
145 REM FILL IN Y-COORDINATE COLORS
150 POKE 55296+X1+(Y*40),C:POKE 55296+X2+(Y*40),C:NEXT
155 REM DO ANOTHER BOX....
160 GOTO 60
```

Explanation of Line Numbers Within Program

- 10-50 Clear screen, set S equal to VIC chip starting address, and make screen all black.
- 60-70 Generate random numbers for boxes.
- 75 Makes sure random numbers not equal to each other.
- 80-95 If X-locations are backwards, reverse them.
- 100-105 Fill in X-coordinate characters and colors.
- 120-130 If Y-locations are backwards, reverse them.
- 140-150 Fill in Y-coordinate characters and colors.
- 160 Do another box.

Important Variables in Program

- S = Start of graphics VIC chip
- X1,X2,Y1,Y2 = Coordinates for box vertices
- C = Random color

Sample Run of Program

See Fig. 2-3 for a sample run of The Art of Boxes program.

Suggested Modifications and Projects

1. Try and create more unusual shapes, such as triangles and cubes.

2. See if you can fill up the boxes with a random color as they are created.

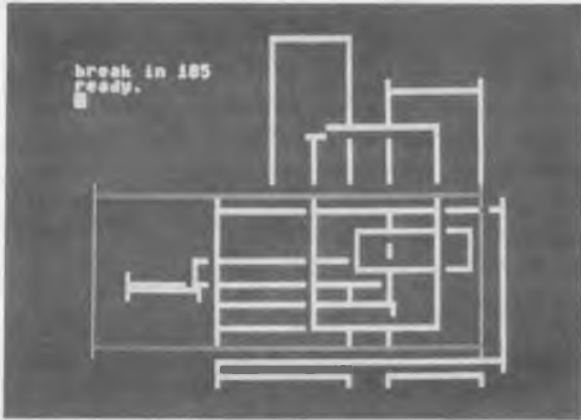


Fig. 2-3. Sample run of The Art of Boxes program.

PROGRAM: SCRAMBLER

This program, as the name suggests, will make a program unreadable by anyone else. However, the main purpose of this program is not encryption, considering that you cannot read the program either, after it has been scrambled. Actually, this program takes the Commodore 64's character set and changes it into random characters, resembling nothing you have ever seen on your screen before. Every letter, number, and symbol will become a strange shape so that no one may read your program in the computer's memory, even if they list it out. Everything about the computer still works, but the letters, numbers, and symbols are completely unreadable! This is done with the programmable character set available on the Commodore, and this program is a good example of how flexible this programmable character set can be (see Listing 2-5).

Listing 2-5. Scrambler

```
10 REM PROGRAM SCRAMBLER
15 REM USING THE PROGRAMMABLE CHARACTER SET OF THE COMMODORE 64
20 PRINT CHR$(142): REM UPPER CASE
```

```

30 POKE 52,8:POKE 56,48:CLR: REM RESERVE CHARACTER MEMORY
40 POKE 56334,PEEK(56334)AND254: REM TURN OFF INTERRUPTS
45 REM SWITCH IN CHARACTER
50 POKE 1,PEEK(1)AND 251
55 REM PUT CHARACTERS INTO RAM
60 FOR I = 0 TO 511:POKE I+12288,PEEK(I+53248):NEXT
65 REM SWITCH INPUT/OUTPUT
70 POKE 1,PEEK(1)OR 4
75 REM RESTART INTERRUPTS
80 POKE 56334,PEEK(56334) OR 1
85 REM TRANSFER COMMODORE'S CHARACTER SET INTO RAM
90 POKE 53272,(PEEK(53272)AND240)+12
95 REM BEGIN SCRAMBLING!
100 FOR I = 12288 TO 12800:POKE I,INT(RND(1)*255)+1:NEXT

```

Explanation of Lines Within Program

- 20 Puts letters into upper-case mode.
- 30 Makes sure memory is reserved for our "character set."
- 40 Disables interrupts to the computer . . . do not disturb!
- 50-60 Take the characters out of the Read-Only Memory and put them into our memory, the Random Access Memory, so we can modify them.
- 70-80 Restart interrupts and resume normal operation.
- 90 Makes the computer recognize the character set in RAM.
- 100 POKES our character set in RAM with random numbers.

Special Notes: When you run this program, you will see the listing itself be scrambled. If you type anything into the computer, it

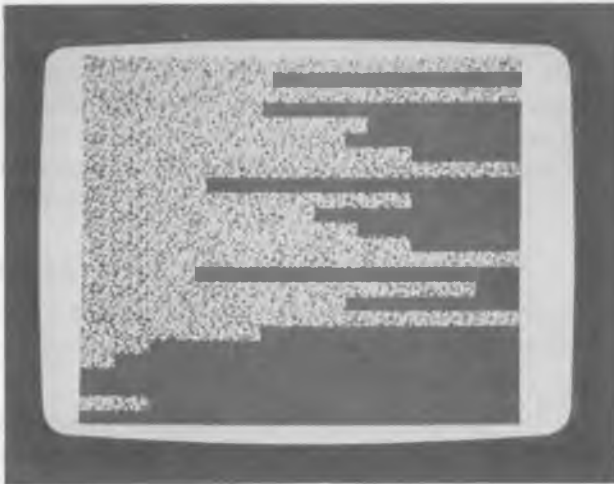


Fig. 2-4. Sample run of Scrambler program.

will still be recognized, except you won't be able to read what you are typing.

Sample Run of Program

See Fig. 2-4 for a sample run of the Scrambler program.

Suggested Modifications and Projects

1. Try and make this program "reversible" so you can switch back to your normal character set at any time.
2. Since you can do anything with the character set you have in RAM, be creative . . . try and make your own alphabet!

Chapter 3

Sounds Galore on the Sixty-Four

The sounds the Commodore 64 can create are as amazing as the graphics, especially considering the economical price of this computer. Here are a total of five programs, showing off the clear, crisp, versatile music the Commodore 64 is capable of, including a utility which makes the computer keyboard into a piano keyboard, so that you may code music more easily. Sample runs of these programs obviously aren't necessary since you just have to "hear the music" to enjoy these creations.

PROGRAM: OUR FIRST SONG

Here's a little number for your computer that utilizes one of the three voices available on the Commodore computer (Listing 3-1).

Explanation of Line Numbers Within Program

- 0-10 Set variable S equal to location of sound chip and clear sound chip.
- 20-27 Establish waveform pattern and set volume to maximum.

- 30-50 Read high frequency, low frequency, duration, and see if all data has been read yet.
- 55-70 Poke memory with high frequency, low frequency, and begin playing the note.
- 75-90 Hold note for duration and stop playing note.
- 100 Go to line 40 for next note.
- 105-1000 Data statements for music.
- 9999-10010 Stop program and clear sound chip.

Listing 3-1. Our First Song

```

0 REM SOMETHING WITH WHICH TO SING YOUR COMPUTER TO SLEEP AT NIGHT
1 REM SOUND CHIP
5 S=54272
7 REM CLEAR SOUND CHIP OF ALL MEMORY
10 FOR L = S TO S+24:POKE L,0:NEXT
20 POKE S+5,9:POKE S+6,0
25 REM SET VOLUME TO MAXIMUM
27 POKE S+24,15
30 REM READ HIGH FREQUENCY, LOW FREQUENCY, AND THE DURATION
40 READ HF,LF,DR
45 REM SIGNAL TO STOP MAKING MUSIC
50 IF HF<0 THEN 10000
55 REM POKE INTO MEMORY THE HIGH FREQUENCY AND LOW FREQUENCY
60 POKE S+1,HF:POKE S,LF
70 POKE S+4,33
75 REM THIS LOOP IS FOR DURATION
80 FOR T= 1 TO DR:NEXT
85 REM SHORT DELAY PROVIDED HERE SO MUSIC WON'T SEEM MUSHED TOGETHER!
90 POKE S+4,32:FOR T=1 TO 50:NEXT
100 GOTO 40
105 REM BEGIN DATA STATEMENTS
110 DATA 12,143,250,16,195,500
120 DATA 12,143,250,10,143,250
130 DATA 9,104,250
140 DATA 8,97,250,7,233,500
150 DATA 14,24,250,12,143,500
160 DATA 12,143,250,10,209,500
170 DATA 15,210,250,12,143,250
175 DATA 11,48,250,9,104,250,8,97,500
180 DATA 14,24,250,12,143,500
190 DATA 12,143,250,21,31,500
200 DATA 16,195,250,12,143,250,10,143,250,8,97,250
210 DATA 11,48,500,14,24,250
220 DATA 16,195,500,15,210,225
230 DATA 14,24,225,12,143,450
240 DATA 15,210,350,10,209,350
250 DATA 16,195,250,15,210,250
260 DATA 16,195,750
1000 DATA -1,-1,-1: REM END DATA STATEMENTS
9999 REM END PROGRAM AND CLEAR CHIP
10000 FOR L = S TO S+24:POKE L,0:NEXT
10010 END
20012 ,143,250,10,143,250,8,97,250

```

Important Variables in Program

S = Location of sound chip

HF = High Frequency

LF = Low Frequency

DR = Duration

T,L = Variables used for loops

Suggested Modifications and Projects

1. See if you can make this into a piece using more than one voice, perhaps with a soft accompanying voice.
2. Select one of your favorite songs and replace the data in this program with the data for your song.

PROGRAM: COMPUTER LULLABY

Here is another piece of music, using a different type of wavelength for an unusual musical effect (Listing 3-2).

Listing 3-2. Computer Lullaby

```
0 REM SOMETHING WITH WHICH TO SING YOUR COMPUTER TO SLEEP AT NIGHT
1 REM SOUND CHIP
5 S=54272
7 REM CLEAR SOUND CHIP OF ALL MEMORY
10 FOR L = S TO S+24:POKE L,0:NEXT
20 POKE S+3,88:POKE S+6,195
25 REM SET VOLUME TO MAXIMUM
27 POKE S+24,15
30 REM READ HIGH FREQUENCY, LOW FREQUENCY, AND THE DURATION
40 READ HF,LF,DR
45 REM SIGNAL TO STOP MAKING MUSIC
50 IF HF<0 THEN 10000
55 REM POKE INTO MEMORY THE HIGH FREQUENCY AND LOW FREQUENCY
60 POKE S+1,HF:POKE S,LF
70 POKE S+4,33
75 REM THIS LOOP IS FOR DURATION
80 FOR T= 1 TO DR:NEXT
85 REM SHORT DELAY PROVIDED HERE SO MUSIC WON'T SEEM MUSHED TOGETHER!
90 POKE S+4,32:FOR T=1 TO 50:NEXT
100 GOTO 40
105 REM BEGIN DATA STATEMENTS
110 DATA 16,195,125,18,209,125
120 DATA 18,209,125,16,195,125
130 DATA 18,209,750,16,195,125
140 DATA 18,209,125,21,31,125
150 DATA 18,209,250,16,195,250
160 DATA 15,210,125,16,195,125
170 DATA 16,195,125,15,210,125
175 DATA 16,195,750,15,210,125
180 DATA 16,195,125,18,209,125
190 DATA 16,195,250,15,210,250
200 DATA 14,24,125,15,210,125
210 DATA 15,210,125,14,24,125
220 DATA 15,210,700,14,24,125
230 DATA 15,210,125,16,195,125
240 DATA 15,210,250,14,24,250
250 DATA 22,96,1000
```

cont. on next page

Listing 3-2—cont. Computer Lullaby

```
260 DATA 18,209,125,21,31,125
270 DATA 21,31,125,18,209,125
280 DATA 21,31,750,18,209,125
290 DATA 21,31,125,22,96,125
300 DATA 21,31,250,18,209,250
310 DATA 16,195,125,18,209,125
320 DATA 18,209,125,16,195,125
330 DATA 18,209,750,16,195,125
340 DATA 18,209,125,21,31,125
350 DATA 18,209,250,16,195,250
360 DATA 15,210,125,16,195,125
370 DATA 16,195,125,15,210,125
375 DATA 16,195,750,15,210,125
380 DATA 16,195,125,18,209,125
390 DATA 16,195,250,15,210,250
400 DATA 25,30,1000,15,210,125
410 DATA 16,195,125,18,209,125
420 DATA 16,195,250,15,210,250
430 DATA 16,195,1500
1000 DATA -1,-1,-1: REM END DATA STATEMENTS
9999 REM END PROGRAM AND CLEAR CHIP
10000 FOR L = 8 TO 8+24:POKE L,0:NEXT
10010 END
```

Explanation of Line Numbers Within Program

- 0-10 Set variable S equal to sound chip address and clear all 25 bytes of the sound chip.
- 20-27 Establish waveform and set volume to maximum.
- 30-50 Read high frequency, low frequency, and duration, and see if end of data has been reached.
- 55-70 Poke in frequency data and begin playing note.
- 75-90 Hold note, stop playing it, then put in short delay so notes won't run together.
- 100 Go to line 40 for next note.
- 105-1000 Data for notes.
- 9999-10010 Clear sound chip and end program.

Important Variables in Program

- S = Location of sound chip in memory
- HF = High frequency
- LF = Low frequency
- DR = Duration of a note
- T,L = Variables for loops

Suggested Modifications and Projects

1. Try and use all three voices with this program, making a nice harmony for this melody.

2. See if you can get some exciting graphics to appear on the screen while the song is playing, perhaps having the screen appear in multiple colors to the beat of the music.

PROGRAM: PIANO MACHINE

When coding music into your programs, you may find it rather monotonous having to look up the low and high frequencies for every note you need. This program will put an end to that monotony, and also transform your Commodore keyboard into a one-octave piano keyboard! See Listing 3-3.

Listing 3-3. Piano Machine

```
1 REM THIS PROGRAM IS USEFUL FOR PLAYING
2 REM ONE-VOICE SONGS AND FINDING OUT THE
3 REM DATA STATEMENTS NEEDED TO RE-PLAY
4 REM THESE SONGS LATER.....
5 REM BY TIM KNIGHT
10 REM PIANO PROGRAM
20 PRINT"J": REM CLEAR SCREEN
30 PRINTTAB(12)"PIANO MACHINE"
31 PRINT:PRINT"USE THE D,F,G,H,J,K,L, AND : KEYS FOR"
32 PRINT"THE WHITE KEYS, MIDDLE C TO HIGH C,"
33 PRINT"AND USE THE R,T,U,I, AND O KEYS FOR THE"
34 PRINT"BLACK KEYS, C SHARP TO A SHARP":PRINT
35 PRINT:PRINT"HOW LONG A DURATION DO YOU WANT FOR"
40 INPUT"EACH NOTE? (NUMBER OF SIXTEENTH NOTES)";DR
45 DR=DR*125
51 S=54272 : REM SOUND ADDRESS
52 FOR L = S TO S+24:POKE L,0:NEXT
53 POKE S+5,9:POKE S+6,0
54 POKE S+24,15
56 POKE S+3,8:POKE S+2,0
58 PRINT "INPUT NOTES"
59 PRINT:PRINT"HIGH FREQUENCY", "LOW FREQUENCY"
60 GETA$:IFA$=""THEN 60
65 REM C
70 IF A$="D" THEN HF=8:LF=97
75 REM C SHARP
80 IF A$="R" THEN HF=8:LF=225
85 REM D
90 IF A$="F" THEN HF=9:LF=104
95 REM D SHARP
100 IF A$="T"THEN HF=9:LF=247
105 REM E
110 IF A$="G"THEN HF=10:LF=143
115 REM F
120 IF A$="H" THEN HF=11:LF=48
125 REM F SHARP
130 IF A$="U" THEN HF=11:LF=218
135 REM G
140 IF A$="J" THEN HF=12:LF=143
145 REM G SHARP
150 IF A$="I" THEN HF=13:LF=78
155 REM A
160 IF A$="K" THEN HF=14:LF=24
```

cont. on next page

Listing 3-3 – cont. Piano Machine

```
165 REM A SHARP
170 IF A$="O" THEN HF=14:LF=239
175 REM B
180 IF A$="L" THEN HF=15:LF=210
184 REM HIGH C
185 IF A$="I" THEN HF=16:LF=195
187 REM POKE HIGH AND LOW FREQUENCIES INTO MEMORY
190 POKE S+1,HF:POKE S,LF
200 POKE S+4,65
201 REM HOLD NOTE FOR EIGHTH NOTE
205 FOR I = 1 TO DR:NEXT
206 PRINTTAB(3):HF:TAB(25):LF
207 REM RELEASE NOTE
210 POKE S+4,64
220 GOTO 60
```

Explanation of Line Numbers Within Program

- 1-35 Clear screen and display instructions.
- 40-45 Input duration of notes.
- 51 Address for sound chip.
- 52 Clears sound chip.
- 53-56 Establish parameters for piano-like sound.
- 59 Prints columns for high frequency and low frequency.
- 60 Gets note.
- 65-185 Determine high and low frequencies for note.
- 187-210 Poke in high and low frequencies, play note, hold for duration, release note, display high/low frequency values.
- 220 Go to line 60 for next note.

Important Variables in Program

- S = Sound address in memory
- DR = Duration
- HF = High frequency
- LF = Low frequency

Suggested Modifications and Projects

1. Using Commodore graphics, display a blank piece of sheet music, and fill it up with notes in their appropriate places on the bars as they are played. Refer to a musical dictionary for extra help in what musical bar-coding looks like.
2. See if you can make duration completely dependent on how long a key is held down so that a duration does not have to be specified and also to create a more realistic "piano."

PROGRAM: SOUNDS OF DIXIE

Here is an electronic version of an old musical piece, ready to run for your musical enjoyment. See Listing 3-4.

Listing 3-4. Sounds of Dixie

```

10 S=54272
15 REM'SET BEGINNING OF SOUND CHIP'
123 FORFL=1TO57:READD1,D2,D3
124 REM'START LOOP TO READ DATA'
125 POKES+5.9:POKES+6.0
126 REM'SET ATTACK/DECAY AND SUSTAIN/RELEASE FOR VOICE 1'
127 POKES+4.33
128 REM'SET WAVEFORM OF VOICE 1 TO SAWTOOTH'
130 POKES+24.15:POKES+1.D1:POKES.D2
135 REM'SET VOLUME TO 15. SET HIGH AND LOW FREQUENCIES FOR VOICE 1'
140 FORDL=1TOD3#7:NEXTDL
145 REM'SET DURATION OF THE NOTE'
150 IFFL=13THENGOSUB 500
160 IFFL=28THENGOSUB 500
170 IFFL=41THENGOSUB 500
175 REM PAUSES ARE NEEDED AFTER THESE NOTES
180 POKES+1.0:POKES,0
185 REM'TURN OFF FREQUENCIES FOR VOICE 1'
190 POKES+4.0
195 REM'TURN OFF WAVEFORM FOR VOICE 1'
200 IFFL=1THEN1000
210 IFFL=5THEN1030
220 IFFL=9THEN1000
230 IFFL=12THEN1045
240 IFFL=13THEN1000
250 IFFL=18THEN1030
260 IFFL=22THEN1000
270 IFFL=25THEN1045
280 IFFL=27THEN1000
290 IFFL=28THEN1045
300 IFFL=32THEN1000
310 IFFL=36THEN1030
320 IFFL=40THEN1000
330 IFFL=46THEN1030
340 IFFL=50THEN1000
350 IFFL=53THEN1045
360 IFFL=56THEN1000
370 IFFL=57THEN1000
380 REM CHANGES IN VOICE 3 ARE NEEDED AT THESE POINTS
400 NEXTFL
450 POKES+24.0:END
460 REM'TURN OF THE VOLUME. END'
500 FORL3=1TO200:NEXTL3:RETURN
510 REM'SET THE DURATION OF PAUSES. RETURN TO THE SONG'
1000 POKES+18.0
1010 REM'TURN OF THE WAVEFORM FOR VOICE 3'
1015 POKES+19.0:POKES+20.240
1017 REM'SET A/D/S/R FOR VOICE 3'
1020 POKES+15.17:POKES+14.37
1022 REM'SET THE FREQUENCIES FOR VOICE 3'
1023 POKES+18.33
1024 REM'SET THE WAVEFORM FOR VOICE 3'
1025 GOTO400
1027 REM'RETURN TO SONG'

```

cont. on next page

Listing 3-4—cont. Sounds of Dixie

```
1030 POKES+18,0
1035 POKES+15,22:POKES+14,227
1037 POKES+18,33
1040 GOTO400
1045 POKES+18,0
1050 POKES+15,12:POKES+14,216
1053 POKES+18,33
1055 GOTO400
5000 DATA1,154,40,19,63,20,17,37,20,21,154,20
5003 DATA19,63,20,17,37,40,34,75,40
5005 DATA20,214,20,34,75,50,25,177,50
5007 DATA21,154,40,17,37,40,19,63,50
5010 DATA21,154,40,19,63,20,17,37,20
5013 DATA21,154,20,19,63,20
5015 DATA17,37,40,34,75,40,28,214,20
5017 DATA34,75,40,25,177,40,21,154,20,17,37,20
5020 DATA19,63,40,19,63,40,17,37,40
5023 DATA32,94,40,34,75,20
5030 DATA38,126,40,25,177,40,25,177,40,28,214,20
5033 DATA25,177,40,34,75,40,34,75,40,28,214,40
5035 DATA22,227,40,28,214,40
5037 DATA25,177,40,21,154,40
5040 DATA19,63,20,17,37,20,21,154,20,19,63,20
5045 DATA17,37,40,34,75,40,28,214,20,34,75,40
5050 DATA25,177,40,21,154,20,17,37,20,19,63,40
5060 DATA19,63,20,19,63,20,17,37,40
```

Explanation of Lines Within Program

- 10 Sets S equal to beginning of sound chip in memory.
- 123 Begin reading data D1, D2, and D3.
- 125 Sets attack, sustain, decay, and release.
- 127-130 Make a sawtooth waveform and poke high and low frequencies into memory.
- 140-170 Hold note for a period of time, and go to another delay subroutine if FL equals 13, 28, or 41.
- 180-190 Turn off frequency and waveform.
- 200-380 Change voice 3 as certain notes are encountered.
- 400 Continues loop.
- 450 Ends program.
- 500 Delays routine.
- 1000-1055 Voice 3 manipulation.
- 5000-5060 Data for song.

Important Variables in Program

- S = Location of sound chip in memory
- FL = Loop for reading notes and note length
- D1 = High frequency
- D2 = Low frequency

D3 = Note duration

DL = Delay loop

Suggested Modifications and Projects

1. Try using a song of your own, using two voices.
2. Attempt to add the second voice to this program, making this a three-voice musical piece.

PROGRAM: BACH MINUET

For you classical fans, here is a soothing piece from the musical genius Bach (Listing 3-5).

Listing 3-5. Bach Minuet

```

50 REM BACH MINUET
60 REM BY DARREN LABATT
100 POKE54296,15
105 REM'SET VOLUME AT HIGHEST LEVEL'
110 FORX=1TO128:READA,B,C
113 REM'READ DATA 3 AT A TIME'
120 POKE54277,9:POKE54278,0
125 REM'SET ATTACK/DECAY/SUSTAIN/RELEASE'
127 POKE54276,33
128 REM'SET WAVEFORM ON SAWTOOTH'
130 POKE54273,A:POKE54272,B
135 REM'SET HIGH AND LOW FREQUENCIES'
140 FORD=1TOD#9:NEXTD
145 REM'SUSTAIN NOTE FOR A CERTAIN LENGHT'
155 POKE54276,0
157 REM'TURN OFF WAVEFORM'
160 NEXTX
170 REM'PLAY NEXT NOTE'
180 POKE54296,0:END
200 REM'THE SONG IS OVER. TURN OFF VOLUME'
5000 DATA9,159,30,6,108,20,7,53,20,8,23,20,8,147,20
5010 DATA9,159,30,6,108,30,6,108,30,10,205,30,8,147,20
5020 DATA9,159,20,10,205,20,12,32,20,12,216,30,6,108,30
5030 DATA6,108,30,8,147,30,9,159,20,8,147,20,8,23,20
5040 DATA7,53,20,8,23,30,8,147,20,8,23,20,7,53,20
5050 DATA6,108,20,6,16,30,6,108,20,7,53,20,8,23,20
5060 DATA6,108,20,7,53,40,9,159,30,6,108,20,7,53,20
5070 DATA8,23,20,8,147,20,9,159,30,6,108,30,6,108,30
5080 DATA10,205,30,8,147,20,9,159,20,10,205,20,12,32,20
5090 DATA12,216,30,6,108,30,6,108,30,8,147,30,9,159,20
5100 DATA8,147,20,8,23,20,7,53,20,8,23,30,8,147,20
5110 DATA8,23,20,7,53,20,6,108,20,7,53,30,8,23,20
5120 DATA7,53,20,6,108,20,6,16,20,6,108,40,9,159,30
5130 DATA6,108,20,7,53,20,8,23,20,8,147,20,9,159,30
5140 DATA6,108,30,6,108,30,10,105,30,8,147,20,9,159,20
5150 DATA10,105,20,12,32,20,12,126,30,6,108,30,6,108,30
5160 DATA8,147,30,9,159,20,8,147,20,8,23,20,7,53,20
5170 DATA8,23,30,8,147,20,8,23,20,7,53,20,6,108,20
5180 DATA6,16,30,6,108,20,7,53,20,8,23,20,6,108,20
5190 DATA7,53,40,9,159,30,6,108,20,7,53,20,8,23,20
5200 DATA8,147,20,9,159,30,6,108,30,6,108,30,10,205,30
5210 DATA8,147,20,9,159,20,10,205,20,12,32,20,12,216,30
5220 DATA6,108,30,6,108,30,8,147,30,9,159,20,8,147,20
5230 DATA8,23,20,7,53,20,8,23,30,8,147,20,8,23,20
5240 DATA7,53,20,6,108,20,7,53,30,8,23,20,7,53,20
5250 DATA6,108,20,6,16,20,6,108,30
    
```

Explanation of Lines Within Program

- 100 Sets volume to highest level.
- 110 Reads in data A, B, and C.
- 120 Establishes attack, decay, sustain, and release.
- 127 Makes sawtooth waveform.
- 130-145 Poke in frequencies and hold note for certain length.
- 155-160 Turn off waveform and go to next note.
- 180 Ends program when loop is finished.
- 5000-5250 Data for music.

Important Variables in Program

- A,B = Frequency values for each note
- C = Duration for each note
- X = Loop to read data and play song
- D = Delay loop

Suggested Modifications and Projects

1. See if you can add special effects to this music by modifying the values for attack, decay, sustain, and release.
2. Modify other aspects of the musical tone by utilizing the sound filtering available on your Commodore 64. More information may be found on filtering in the *Commodore 64 Programmer's Reference Manual*.

Chapter 4

Games

A computer such as the Commodore 64, with its amazing graphics and sounds, is an excellent machine for playing games. A variety of entertainment programs with which you may play, modify, and learn about some of the games that a computer can play are included in this chapter. You might even get a game idea yourself that you could create on your own computer.

PROGRAM: JELLY BEANS

This game uses both the graphics and the sounds of the Commodore 64 to play a variation of the game, "high/low." The game is self-explanatory, and a lot of fun for youngsters who are trying to get used to a computer.

Listing 4-1. Jelly Beans

```
1 PRINTCHR$(147)
2 REM'CLEAR SCREEN'
3 FOR NL=1TO10
4 REM'BEGIN POKING EACH LETTER OF THE TITLE'
5 DE=1464:TL=55736
```

cont. on next page

Listing 4-1—cont. Jelly Beans

```
6 REM'SET SCREEN MEMORY LOCATION AND CORRESPONDING COLOR TO WHITE'
7 READD1,D2
8 REM'GET NEEDED DATA'
10 FORY=1TOD1
13 REM'SET NUMBER OF SPACES FOR THIS LETTER TO MOVE'
15 POKEDE,D2:FORX=1TO10:NEXTX
16 REM'POKE THIS LETTER, WAIT'
17 POKETL,1:TL=TL+1
18 REM'POKE THE COLOR. SET IT TO NEXT MEMORY LOCATION'
20 POKEDE,D2:DE=DE+1
25 REM'CLEAR THIS SPACE, SET LETTER TO NEXT MEMORY LOCATION'
35 NEXTY
37 REM'GO ON TO MOVE THE LETTER AGAIN'
50 POKEDE,D2:POKETL,1
55 REM'SET THE LETTER HERE PERMANENTLY'
60 NEXTNL
70 REM'GO ON TO THE NEXT LETTER'
100 FORFF=1TO1000:NEXTFF
103 REM'WAIT A WHILE'
105 PRINTCHR$(147)
107 REM'CLEAR THE SCREEN'
110 POKE53281,1
113 REM'COLOR THE SCREEN WHITE'
115 AB=INT(100*RNDC(1))+1
116 REM'GET A RND NUMBER FOR THE NUMBER OF DOTS TO BE USED'
117 FOR A=1TOAB
119 REM THIS LOOP IS USED TO PLACE AND COLOR THE DOTS
120 BC=INT(1000*RNDC(1))+1
123 REM THIS NUMBER WILL BE USED TO PLACE THE DOTS
125 IFPEEK(1024+BC)=81THEN120
127 REM'IF A DOT WAS ALREADY PLACED HERE GO BACK AND GET ANOTHER LOCATION'
130 POKE1024+BC,81
135 REM'POKE IN THE DOT'
140 CD=INT(7*RNDC(1))+2
142 REM'GET A RANDOM COLOR'
150 POKE55296+BC,CD
155 REM'COLOR THE DOT'
160 V=54296:W=54276:D=54277:H=54273:L=54273
165 REM'SET VARIABLES TO FORM A SOUND'
170 POKEV,13:POKEW,33:POKED,8:POKEH,16:POKEL,108
172 REM'MAKE THE SOUND'
174 FORB=1TO5:NEXTB
175 REM'SET THE DURATION OF THE SOUND'
176 POKEL,0:POKEH,0:POKEW,0
178 REM'TURN OFF THE SOUND'
180 NEXTA
185 REM'GO ON TO NEXT DOT'
190 POKEV,0
195 REM'TURN OFF THE VOLUME'
200 PRINTCHR$(147):POKE53281,6:PRINT:PRINT:PRINT
203 REM'CLEAR THE SCREEN, TURN IT BLUE AGAIN'
205 GU=0
207 REM'SET NUMBER OF GUESSES AT 0'
210 INPUT"HOW MANY JELLY BEANS DID YOU SEE";FE
215 GU=GU+1
217 REM'ADD ONE GUESS'
220 IFFE=ABTHEN230
221 REM'IF THE GUESS IS CORRECT GO TO 230'
222 IFFE>ABTHEN240
226 IFFE<ABTHEN250
230 PRINT"YOU GOT THE CORRECT AMOUNT IN";GU;"GUESSES.":GOTO260
240 PRINT"THERE WERE LESS JELLY BEANS.":GOTO210
245 REM'GO ASK THE QUESTION AGAIN'
250 PRINT "THERE WERE MORE JELLY BEANS.":GOTO210
```

```

260 INPUT"DO YOU WANT TO PLAY AGAIN(Y/N)";F0$
270 IFFG$="Y"THEN105
290 REM'IF YES GO START AT 105'
500 DATA23,19,22,14,21,1,20,5,19,2,17,25,16,12,15,12,14,5,13,10

```

Important Variables in Program

NL = Loop to make title
 DE = Screen memory
 TL = Color memory for screen
 Y = Moves letters, one at a time, across screen
 D1, D2 = Data for title
 FF = Delay loop
 AB = Random number for jelly beans
 BC = Random number for position of a jelly bean
 CD = Color of each jelly bean
 V = Volume
 W = Waveform
 D = Attack/decay rate
 H = High frequency
 L = Low frequency
 GU = Number of guesses
 FE = Input (line 210)

Explanation of Important Line Numbers Within Program

3-70 Print title of game.
 105-110 Clear screen and color it white.
 115-135 Put random number of jelly beans on screen.
 140-150 Color each dot.
 160-170 Sound routine.
 180 Return to put another dot on screen.
 200-250 Ask player to guess number of dots.
 260-270 Ask if player wants to play again.
 500 Data used for title.

Sample Run of Program

```

HOW MANY JELLY BEANS DID YOU SEE? 15
THERE WERE MORE JELLY BEANS
HOW MANY JELLY BEANS DID YOU SEE? 25
THERE WERE LESS JELLY BEANS
HOW MANY JELLY BEANS DID YOU SEE? 19

```

**YOU GOT THE CORRECT AMOUNT IN 3 GUESSES.
DO YOU WANT TO PLAY AGAIN (Y/N)?**

Suggested Modifications and Projects

1. See if you can program a little fanfare song if the person guesses the right number of jelly beans.
2. Experiment with different characters, rather than the "jelly beans" used.
3. Try to program a variable skill level into your program, so that the rate at which the jelly beans appear is slower for novice players and faster for experienced players.

PROGRAM: ROCKET RACE

If you are not allowed a day at the races, why not try a little futuristic race on your own computer? Place your bets on which rocket will win, and see how well you fare at gambling on the space-ships. See Listing 4-2.

Listing 4-2. Rocket Race

```

100 BM=100
105 REM THE PLAYER WILL START WITH $100
115 F=0:E1=0:E2=0:E3=0
117 H1=0:H2=0:H3=0:SR=0
118 REM SET VARIABLES
120 PRINTCHR$(147):PRINT:PRINT
125 REM CLEAR SCREEN, SPACING
140 PRINTSPC(4)"WHICH ROCKET DO YOU BET WILL FINISH FIRST?"
150 INPUT"A,B OR C";A$
153 REM GET PLAYER'S RESPONSE
155 IFA$="A"ORA$="B"ORA$="C"THEN160
157 GOTO150
159 REM MAKE SURE ONLY A,B OR C IS CHOSEN
160 PRINT:PRINT
170 PRINT"DO YOU WANT TO BET ON WHICH WILL FINISH"
175 INPUT"SECOND(Y/N)";I$
178 REM GET PLAYER'S RESPONSE
180 IFI$="Y"THENSR=1:GOTO200
190 PRINT:PRINT:GOTO220
200 PRINTSPC(3)"ON WHICH OTHER ROCKET DO YOU WISH TO"
202 REM SPC(X) IF FOR SPACING
203 INPUT"BET";B$
205 IFB$="A"ORB$="B"THEN210
206 IFB$="C"THEN210
207 GOTO200
208 REM MAKE SURE ONLY A,B,OR C CHOSEN
210 IFB$=A$THEN200
215 REM MAKE SURE THAT THE PLAYER DID NOT BET ON THE SAME ROCKET TWICE
220 PRINT"YOU HAVE ";BM;" DOLLARS."
230 INPUT"HOW MUCH DO YOU WISH TO BET";BT
240 IFT>BMTHEN230
245 REM MAKE SURE HE DOES NOT BET MORE THAN HE HAS!
250 BM=BM-BT
255 REM THIS WILL BE THE PLAYER'S HOLDINGS IF HE LOSES

```

```

1000 PRINTCHR$(147):V=53248:POKEV+21,7
1010 POKE1989,1:POKE1993,2:POKE1997,3
1015 REM THIS WILL IDENTIFY EACH ROCKET WITH A LETTER
1020 POKE56261,1:POKE56265,12:POKE56269,3
1025 REM THIS WILL COLOR EACH LETTER TO MATCH EACH OF THE ROCKETS
1050 REM ACTIVATE SPRITES 0-2
1100 POKE2040,13
1150 REM SET POINTERS FOR SPRITE 0
1200 FORS1=0TO62:READD1:POKE832+S1,D1:NEXTS1:RESTORE
1230 REM READ DATA TO FORM SPRITE 0,RESTORE DATA TO THE BEGINNING
1250 POKE2041,14
1300 FORS2=0TO62:READD2:POKE896+S2,D2::NEXTS2:RESTORE
1330 REM SET AND FORM SPRITE 1
1350 POKE2042,15:POKEV+40,15
1355 REM SET POINTERS AND COLOR (LT. GREY) FOR SPRITE 2 (OR ROCKET C)
1400 FORS3=0TO62:READD3:POKE960+S3,D3:NEXTS3:RESTORE
1450 Y1=230:Y2=230:Y3=230
1475 F=0:H1=0:H2=0:H3=0
1480 REM SET AND RESET CERTAIN VARIABLES
1500 POKEV+0,60:POKEV+1,Y1
1600 POKEV+2,90:POKEV+3,Y2
1700 POKEV+4,120:POKEV+5,Y3
1750 REM POSITION THE 3 SPRITES
1800 R1=INT(3*WRND(1)+1)
1900 IFR1=1THEN2200
2000 IFR1=2THEN2400
2100 IFR1=3THEN2600
2150 REM GET A RANDOM NUMBER TO MOVE ROCKETS RANDOMLY
2200 Y1=Y1-1:IFY1<50THEN3000
2250 REM SET A NEW Y POSITIONING VALUE FOR SPRITE0 AND BE SURE IT'S NOT BELOW 50
2300 POKEV+0,60:POKEV+1,Y1:GOTO1800
2350 REM POKE IN THIS NEW VALUE
2400 Y2=Y2-1:IFY2<50THEN3100
2500 POKEV+2,90:POKEV+3,Y2:GOTO1800
2550 REM SAME CHANGES TO SPRITE 1
2600 Y3=Y3-1:IFY3<50THEN3200
2700 POKEV+4,120:POKEV+5,Y3
2750 REM SAME CHANGES TO SPRITE 2
2800 GOTO1800
3000 IFH1=1THEN1800
3010 REM SEE IF ROCKET 'A' ALREADY REACHED 50
3050 GOTO3300
3080 REM IT DID NOT, SO GOTO 3300
3100 IFH2=1THEN1800
3110 REM CHECK IF 'B' ALREADY REACHED 50
3150 GOTO3400
3200 IFH3=1THEN1800
3210 REM CHECK IF 'C' ALREADY REACHED 50
3250 GOTO3500
3300 F=F+1:E1=E1+F:H1=1
3310 REM SEE VARIABLE LIST
3350 IFF=3THEN2000
3360 REM THE RACE MUST BE OVER IF F=3
3370 GOTO1800
3375 REM THE RACE IS NOT OVER, RETURN FOR MORE
3400 F=F+1:E2=E2+F:H2=1
3450 IFF=3THEN2000
3475 GOTO1800
3480 REM SAME ROUTINES FOR 'B'
3500 F=F+1:E3=E3+F:H3=1
3550 IFF=3THEN2000
3575 GOTO1800
3580 REM SAME ROUTINES FOR 'C'
10000 DATA0,24,0,0,60,0,0,126,0,0,126,0,0,126,0
10010 DATA0,126,0,0,126,0,0,126,0,0,126,0,0,126,0
10020 DATA0,126,0,0,126,0,0,126,0,0,126,0,0,126,0

```

cont. on next page

Listing 4-2—cont. Rocket Race

```
10030 DATA0,126,0,0,255,0,1,195,128,1,195,128
10040 DATA1,195,128,1,195,128
20000 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
21000 PRINT"ROCKET 'A' CAME IN AS NO.,";E1
22000 PRINT"ROCKET 'B' CAME IN AS NO.,";E2
23000 PRINT"ROCKET 'C' CAME IN AS NO.,";E3
25000 IFE1=1THEN26000
25050 REM IF 'A' WON, GOTO26000
25100 IFE2=1THEN26100
25150 REM IF 'B' WON, GOTO26100
25200 IFE3=1THEN26200
25250 REM IF 'C' WON, GOTO26200
26000 IFA$<"A"THEN27000
26005 REM DID THE PLAYER SAY 'A' WOULD WIN? IF NOT GOTO27000
26010 IFSR=1THEN26500
26015 REM NOW DID HE BET ON THE SECOND ONE TO WIN ALSO?
26020 BT=BT#4
26030 BM=BM+BT
26033 REM THE PLAYER WON. HE GETS 3X WHAT HE BET AND HIS BET BACK
26035 PRINT:PRINT:PRINT"YOU WIN"
26040 GOTO27200
26100 IFA$<"B"THEN27000
26105 REM SEE THE SUBROUTINE AT 26000
26110 IFSR=1THEN26500
26120 BT=BT#4
26130 BM=BM+BT
26135 PRINT:PRINT:PRINT"YOU WIN"
26140 GOTO27200
26200 IFA$<"C"THEN27000
26205 REM SEE THE SUBROUTINE AT 26000
26210 IFSR=1THEN26500
26220 BT=BT#4
26230 BM=BM+BT
26235 PRINT:PRINT:PRINT"YOU WIN"
26240 GOTO27200
26500 IFE1=2THEN26600
26510 IFE2=2THEN26700
26520 IFE3=2THEN26800
26530 REM SEE WHICH CAME IN SECOND
26600 IFB$<"A"THEN27000
26610 BT=BT#10
26615 REM THE PLAYER GUESSED THE ORDER OF THE ROCKETS
26616 REM HE DESERVES 9X HIS BET
26620 BM=BM+BT
26630 PRINT:PRINT:PRINT"YOU WIN!"
26640 GOTO27200
26700 IFB$<"B"THEN27000
26705 REM SEE SUBROUTINE AT 26600
26710 BT=BT#10
26720 BM=BM+BT
26730 PRINT:PRINT:PRINT"YOU WIN!"
26740 GOTO27200
26800 IFB$<"C"THEN27000
26805 REM SEE SUBROUTINE AT 26600
26810 BT=BT#10
26820 BM=BM+BT
26830 PRINT:PRINT:PRINT"YOU WIN!"
26840 GOTO27200
27000 PRINT:PRINT:PRINT"YOU LOSE"
27200 PRINT:PRINT:PRINT
27250 PRINT"YOU HAVE";BM;"DOLLARS."
27260 REM THIS IS THE NEW TOTAL HOLDINGS
27300 IFBM=0THEN29000
27310 REM IF HE HAS NO MONEY END THE GAME
```

```

27400 INPUT"DO YOU WISH TO TRY ROBIN(Y/N)";I2$
27500 IF I2$="Y" THEN I15
28000 END
29000 PRINT:PRINT:PRINT"YOU'RE BROKE!"

```

Important Variables in Program

BM = Betting money

E1, E2, E3 = Order in which rockets finish

H1, H2, H3 = Flags to see which rockets finish

A\$ = Rocket on which player bets

SR = Flag determining how many rockets are being bet on

B\$ = Rocket which player believes will come in second

BT = Bet money

V = Video chip starting address

S1, S2, S3 = Loops for sprites 1, 2, and 3

D1, D2, D3 = Data for sprites 1, 2, and 3

Y1, Y2, Y3 = Vertical position of sprites, 1, 2, and 3

R1 = Random number to move a random sprite

Explanation of Important Lines Within Program

10-117 Set up variables.

120-255 Ask player for bet and on which rocket he would like to bet.

1000-1400 Form 3 sprites.

1450-1700 Position sprites.

1800-2800 Move sprites randomly.

3000-3250 See what order the rockets finish.

3300-3575 See if race is over.

10000-10040 Data for sprites.

20000-23000 Display how rockets place.

25000-26840 See if player wins and pay accordingly.

Sample Run of Program

A sample run of the Rocket Race program is shown in Fig. 4-1.

Suggested Modifications and Projects

1. See if you can program the keyboard so that two players may press certain keys very rapidly to make the rockets move. In this way, the rockets are not moving randomly, but are controlled by competing players.

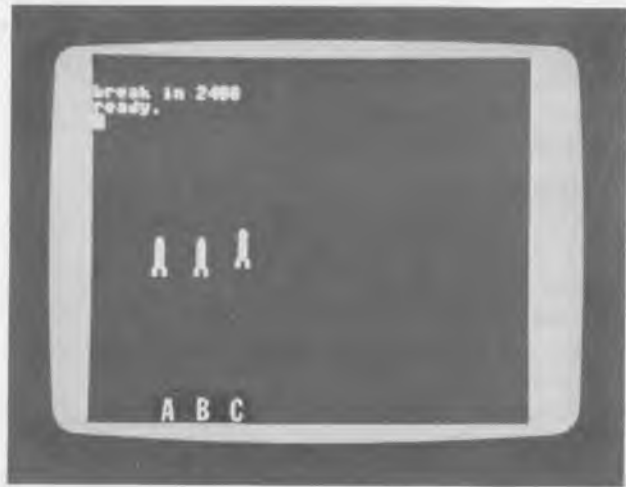


Fig. 4-1. Sample run of Rocket Race program.

2. Try to reprogram the sprites for different types of shapes — maybe even resembling horses!
3. Put a fanfare at the beginning and end of this “race,” just like a racing field would have.

PROGRAM: DICE GAMES

Chuck-a-luck and high-low, two popular dice games, are contained within this program (Listing 4-3). To play chuck-a-luck, simply tell the computer what number you think will appear most often on the three dice “thrown.” For every die that appears with your number, you win some of the money you bet. For instance, if you thought the number “3” would appear most often, and it appeared on two out of the three dice, you win some money. In high-low, you bet on two dice. You tell the computer that the dice will have a total of below seven, above seven, or exactly seven. The computer will then “roll the dice,” and you will be rewarded if your guess was correct. Naturally, if you guess that the dice will equal seven, and they do turn out to equal seven, you will be rewarded even more money, since the odds of getting exactly seven are against you.

Listing 4-3. Dice Games

```

50 PRINTCHR$(147)
100 PRINTSPC(15)"DICE"
110 PRINT:PRINT:PRINT
120 PRINTSPC(8)"1. CHUCK-A-LUCK
130 PRINT:PRINT:PRINT
140 PRINTSPC(8)"2. HIGH-LOW
150 PRINT:PRINT:PRINT
160 INPUT"PLEASE CHOOSE A GAME";GA
165 BM=100
167 REM'SET BETTING MONEY AT $100'
170 IFGA=1THEN300
180 IFGA=2THEN400
200 REM'GO TO SPECIFIC SUBROUTINES'
300 PRINTCHR$(147):PRINT:PRINT
305 PRINT:PRINT:PRINT
310 PRINT"YOU HAVE";BM;"DOLLARS."
315 IF BM=0 THEN 13450
320 PRINT "WHICH NUMBER DO YOU WISH TO BET ON?"
325 PRINT "NUMBERS 1-6?"
326 INPUT NM
327 PRINT
330 PRINT:PRINT
340 INPUT"HOW MUCH DO YOU WISH TO BET";BT
342 IF BT<=BM THEN 360
345 PRINT
350 IFBT>BMTHENPRINT"YOU DON'T HAVE THAT MUCH MONEY"
352 PRINT "BET AGAIN":PRINT:GOTO 340
355 REM'IF HE BETS MORE THAN HE HAS, GO ASK AGAIN'
360 BM=BM-BT:GOTO300
370 REM THIS WILL BE THE PLAYER'S HOLDINGS IF HE LOSES
380 REM'GO ROLL THE DICE'
400 PRINT CHR$(147):PRINT:PRINT:PRINT
403 REM'BEGIN HIGH-LOW BETTING'
405 PRINT"DO YOU THINK THE DICE WILL BE:"
410 PRINTSPC(5)"1. HIGHER THAN 7"
420 PRINTSPC(5)"2. LOWER THAN 7"
430 PRINTSPC(5)"3. EXACTLY 7"
440 INPUT" ".NB
450 PRINT:PRINT
460 PRINT"YOU HAVE";BM;"DOLLARS."
470 INPUT"WHAT IS YOUR BET";BT
472 IF BT<=BM THEN 490
475 PRINT
480 IFBT>BMTHENPRINT"YOU DON'T HAVE THAT MUCH MONEY"
482 PRINT"BET AGAIN":PRINT:GOTO470
490 BM=BM-BT
900 R1=INT(RND(1)*18)+1
910 R2=INT(RND(1)*18)+1
920 R3=INT(RND(1)*18)+1
930 REM'GET A RANDOM NUMBER OF ROLLS FOR EACH DIE'
1000 V=53248:POKEV+21,7
1010 REM'SET BEGINNING OF VIDEO CHIP, ACTIVATE SPRITES 0,1,AND 2'
1050 POKE2040,192
1060 REM'SET POINTERS FOR DIE 1'
1100 FORG1=0TO62:READD1:POKE12288+G1,D1:NEXTG1:GOSUB10000
1120 REM'READ AND POKE IN DATA FOR DIE 1, GO POSITION THE SPRITE'
1130 IFR1=0THENS1=1:OOT012000
1135 REM'IF THE ROLLS RAN OUT,SET VALUE OF DIE 1 AT 1, GO TO A DIFFERENT
DIE'
1140 R1=R1-1
1145 REM'SUBTRACT ONE ROLL FROM TOTAL'
1150 POKE2040,193

```

cont. on next page

Listing 4-3—cont. Dice Games

```
1160 REM SAME PROCEDURE FOR FACE 2 OF DIE 1
1200 FORG2=0T062:READD2:POKE12352+G2,D2:NEXTG2:GOSUB10000
1230 IFR1=0THENS1=2:GOTO12000
1240 R1=R1-1
1250 POKE2040,194
1300 FORG3=0T062:READD3:POKE12416+G3,D3:NEXTG3:GOSUB10000
1330 IFR1=0THENS1=3:GOTO12000
1340 R1=R1-1
1350 POKE2040,195
1360 FORG4=0T062:READD4:POKE12480+G4,D4:NEXTG4:GOSUB10000
1367 IFR1=0THENS1=4:GOTO12000
1368 R1=R1-1
1370 POKE2040,196
1380 FORG5=0T062:READD5:POKE12544+G5,D5:NEXTG5:GOSUB10000
1387 IFR1=0THENS1=5:GOTO12000
1388 R1=R1-1
1390 POKE2040,197
1400 FORG6=0T062:READD6:POKE12608+G6,D6:NEXTG6:GOSUB10000
1430 IFR1=0THENS1=6:GOTO12000
1440 R1=R1-1
2500 RESTORE:GOTO1050
2510 REM'RESTORE DATA, GO ROLL DIE 1 AGAIN'
2550 POKE2041,192
2560 REM'START SAME PROCEDURE FOR DIE 2'
2600 FORH1=0T062:READE1:POKE12288+H1,E1:NEXTH1:GOSUB10250
2630 IFR2=0THENS2=1:GOTO12100
2640 R2=R2-1
2700 POKE2041,193
2800 FORH2=0T062:READE2:POKE12352+H2,E2:NEXTH2:GOSUB10250
2830 IFR2=0THENS2=2:GOTO12100
2840 R2=R2-1
2900 POKE2041,194
3000 FORH3=0T062:READE3:POKE12416+H3,E3:NEXTH3:GOSUB10250
3005 IFR2=0THENS2=3:GOTO12100
3007 R2=R2-1
3010 POKE2041,195
3020 FORH4=0T062:READE4:POKE12480+H4,E4:NEXTH4:GOSUB10250
3025 IFR2=0THENS2=4:GOTO12100
3027 R2=R2-1
3030 POKE2041,196
3040 FORH5=0T062:READE5:POKE12544+H5,E5:NEXTH5:GOSUB10250
3042 IFR2=0THENS2=5:GOTO12100
3043 R2=R2-1
3045 POKE2041,197
3050 FORH6=0T062:READE6:POKE12608+H6,E6:NEXTH6:GOSUB10250
3055 IFR2=0THENS2=6:GOTO12100
3057 R2=R2-1
3060 RESTORE:GOTO2550
3100 POKE2042,192
3105 REM'START SAME PROCEDURE FOR DIE 3'
3110 FORI1=0T062:READQ1:POKE12288+I1,Q1:NEXTI1:GOSUB11250
3115 IFR3=0THENS3=1:GOTO12200
3117 R3=R3-1
3120 POKE2042,193
3130 FORI2=0T062:READQ2:POKE12352+I2,Q2:NEXTI2:GOSUB11250
3135 IFR3=0THENS3=2:GOTO12200
3137 R3=R3-1
3140 POKE2042,194
3150 FORI3=0T062:READQ3:POKE12416+I3,Q3:NEXTI3:GOSUB11250
3155 IFR3=0THENS3=3:GOTO12200
3157 R3=R3-1
3160 POKE2042,195
3170 FORI4=0T062:READQ4:POKE12480+I4,Q4:NEXTI4:GOSUB11250
3175 IFR3=0THENS3=4:GOTO12200
```

```

3177 R3=R3-1
3180 POKE2042,196
3190 FOR I5=0 TO 62: READ Q5: POKE12544+I5, Q5: NEXT I5: GO SUB 11250
3195 IFR3=0 THEN S3=5: GO TO 12200
3197 R3=R3-1
3200 POKE2042,197
3210 FOR I6=0 TO 62: READ Q6: POKE12600+I6, Q6: NEXT I6: GO SUB 11250
3220 IFR3=0 THEN S3=6: GO TO 12200
3225 R3=R3-1
4500 RESTORE: GO TO 03100
5000 DATA 0, 0, 0, 63, 255, 252, 127, 255, 254, 255, 255, 255, 255, 255, 255, 255, 255, 255
5010 DATA 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 195, 255, 255, 195, 255, 255
5020 DATA 255, 195, 255, 255, 195, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255
5030 DATA 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 127, 255, 254, 63, 255, 252
5040 DATA 0, 0, 0, 63, 255, 252, 127, 255, 254, 255, 255, 135, 255, 255, 135, 255, 255, 135
5050 DATA 255, 255, 135, 255, 255, 135, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255
5060 DATA 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255
5070 DATA 225, 255, 255, 225, 255, 255, 225, 255, 255, 255, 225, 255, 255, 225, 255, 255
5080 DATA 127, 255, 254, 63, 255, 252
5090 DATA 0, 0, 0, 63, 255, 252, 127, 255, 254, 225, 255, 255, 225, 255, 255
6000 DATA 225, 255, 255, 225, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255
6010 DATA 255, 195, 255, 255, 195, 255, 255, 195, 255, 255, 195, 255
6020 DATA 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 135, 255, 255, 135, 255, 255, 135
6030 DATA 255, 255, 135, 127, 255, 254, 63, 255, 252
6050 DATA 0, 0, 63, 255, 252, 127, 255, 254, 240, 255, 15, 240, 255, 15
6070 DATA 240, 255, 15, 240, 255, 15, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255
6080 DATA 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255
6090 DATA 240, 255, 15, 240, 255, 15, 240, 255, 15, 240, 255, 15, 127, 255, 254
7000 DATA 63, 255, 252
7010 DATA 0, 0, 0, 63, 255, 252, 127, 255, 254, 240, 255, 15, 240, 255, 15
7020 DATA 240, 255, 15, 240, 255, 15, 255, 255, 255, 255, 255, 255
7030 DATA 255, 195, 255, 255, 195, 255, 255, 195, 255, 255, 195, 255
7040 DATA 255, 255, 255, 255, 255, 255, 240, 255, 15, 240, 255, 15, 240, 255, 15
7050 DATA 240, 255, 15, 127, 255, 254, 63, 255, 252
7060 DATA 0, 0, 0, 63, 255, 252, 127, 255, 254, 248, 126, 31, 248, 126, 31
7070 DATA 248, 126, 31, 248, 126, 31, 255, 255, 255, 255, 255, 255, 248, 126, 31
7080 DATA 248, 126, 31, 248, 126, 31, 248, 126, 31, 255, 255, 255, 255, 255, 255
7090 DATA 248, 126, 31, 248, 126, 31, 248, 126, 31, 248, 126, 31, 127, 255, 254, 63, 255, 252
10000 POKEV,50:POKEV+1,60:RETURN
10010 REM POSITION DIE 1, KEEP IT ROLLING'
10250 POKEV+40,1
10300 REM COLOR DIE 2 WHITE'
10500 POKEV+2,150:POKEV+3,60
11000 RETURN
11250 POKEV+41,1
11270 REM COLOR DIE 3 WHITE'
11300 POKEV+4,250
11350 POKEV+5,60:RETURN
11370 REM POSITION DIE 3'
12000 IFS1=NMTHENNHNH+1
12010 REM IF THE VALUE OF DIE 1 IS EQUAL TO THE NUMBER BET ON, ADD TO # OF HITS'
12050 RESTORE: GO TO 2550
12070 REM RESTORE DATA, GO ROLL DIE 2'
12100 IFS2=NMTHENNHNH+1
12110 IFOA=2 THEN I3000
12130 REM IF PLAYER CHOSE HIGH-LOW, DON'T ROLL DIE 3'
12150 RESTORE: GO TO 3100
12200 IFS3=NMTHENNHNH+1
12300 IFNH=0 THEN 12410
12305 REM IF PLAYER DID NOT SCORE IN GAME 1, DON'T ADD TO HIS BET'
12310 IFNH=1 THEN BT=BT#2: GO TO 12400
12320 IFNH=2 THEN BT=BT#3: GO TO 12400
12330 IFNH=3 THEN BT=BT#4: GO TO 12400
12340 REM ADD MONEY TO WINNER OF GAME 1 ACCORDING TO ODDS'
12400 BM=BM+BT
12402 REM THIS IS THE PLAYER'S NEW TOTAL BETTING MONEY
12403 PRINT:PRINT

```

cont. on next page

Listing 4-3—cont. Dice Games

```

12405 PRINT"YOU HIT";NH;"TIMES."
12410 PRINT:PRINT
12420 INPUT"WANT TO PLAY AGAIN";A1$
12430 IFA1$="Y"THENRESTORE:NH=0:GOTO300
12500 END
13000 RESTORE:SC=S1+S2
13005 REM'SET SUM OF DIE 1 AND 2 FOR HIGH-LOW'
13010 IFNB=1THEN13100
13020 IFNB=2THEN13200
13030 IFNB=3THEN13300
13050 REM'GO TO SUBROUTINES ACCORDING TO EVENT BET ON'
13100 IFSC<7ORSC=7THEN13400
13105 REM'IF THIS IS NOT TRUE THEN THE PLAYER DID NOT WIN'
13110 BM=BM+(BT#2)
13115 REM'SET NEW TOTAL BETTING MONEY'
13120 GOTO13400
13200 IFSC>7ORSC=7THEN13400
13210 BM=BM+(BT#2)
13220 GOTO13400
13300 IFSC<>7THEN13400
13310 BM=BM+(BT#3)
13320 GOTO13400
13400 PRINT:PRINT"YOU NOW HAVE";BM;"DOLLARS."
13450 IF BM=0 THEN PRINT "SORRY YOU'RE BUSTED! GAME IS OVER!":END
13500 INPUT"WANT TO PLAY AGAIN";A2$
13510 IFA2$="Y"THEN400
13520 IFA2$="N"THENEND

```

Important Variables in Program

BM = Betting money
 NB = Number bet on in chuck-a-luck
 NM = Choice bet on in high-low
 BT = Amount of money bet
 D1, D2, D3 = Random values for three dice
 NH = Number of "hits" in chuck-a-luck
 X\$ = Input to play again
 SC = Sum of first two dice
 Y\$ = Input to play again

Explanation of Important Line Numbers in Program

50-180 Ask player to choose a game.
 300-380 Ask player's bet on chuck-a-luck.
 400-490 Ask how the player wishes to bet in high-low.
 900-920 Get random number of rolls from each die.
 1000-4500 Form six possible faces on dice.
 5000-7090 Data for sprites.
 10000-11350 Position three sprites.
 12000-12200 Add up "hits" scored in chuck-a-luck.

12300-12405 Pay off player in chuck-a-luck and add winnings, if any, to total.

13000-13510 See if player wins in high-low and add winnings, if any, to total.

If you don't want to type in all of the extra graphics features, a "no-frills" version of the program is given in Listing 4-4.

Listing 4-4. No-Frills Dice

```

50 BM=100
60 REM'SET TOTAL BETTING MONEY TO $100'
100 PRINTCHR$(147):PRINT:PRINT
110 PRINTSPC(5)"1. CHUCK-A-LUCK"
120 PRINTSPC(5)"2. HIGH-LOW"
130 PRINT:PRINT
140 INPUT"PLEASE CHOOSE A GAME";CH
150 PRINT"YOU HAVE";BM;"DOLLARS"
160 IFCH=2THEN200
170 INPUT"WHICH NUMBER DO YOU WISH TO BET ON";NB
180 INPUT"HOW MUCH DO YOU WISH TO BET";BT
183 IFBT<=BMTHEN190
185 IFBT>BMTHEN PRINT"YOU DON'T HAVE THAT MUCH MONEY BET AGAIN":PRINT:GOTO100
187 REM'IF HE BETS MORE THAN HE HAS, GO ASK AGAIN'
190 GOTO300
195 REM'GO ROLL THE DICE'
200 PRINT:PRINT"1. HIGHER THAN SEVEN."
210 PRINT"2. LOWER THAN SEVEN."
220 PRINT"3. EXACTLY SEVEN."
230 PRINT:PRINT
240 PRINT"WHICH OF THE ABOVE DO YOU BET YOU WILL"
250 INPUT"ROLL";NM
260 INPUT"HOW MUCH DO YOU WISH TO BET";BT
265 IF BT<=BM THEN 300
270 IFBT>BMTHEN PRINT"YOU DON'T HAVE THAT MUCH MONEY BET AGAIN":PRINT:GOTO 260
300 PRINTCHR$(147):PRINT:PRINT
310 D1=INT(RND(1)*6)+1
320 D2=INT(RND(1)*6)+1
330 D3=INT(RND(1)*6)+1
335 REM D1-D3 ARE THE VALUES FOR THE 3 DICE
340 BM=BM-BT
343 REM'SUBTRACT BET FROM TOTAL HOLDINGS'
345 PRINTD1:PRINTD2
350 IFCH=2THEN600
355 REM'IF PLAYING GAME 2, DON'T DISPLAY THE VALUE OF DIE 3'
360 PRINTD3:PRINT
400 IFD1=NBTHENNH=NH+1
410 IFD2=NBTHENNH=NH+1
420 IFD3=NBTHENNH=NH+1
425 REM'ADD NUMBER OF TIMES GAME 1 PLAYER SCORES A HIT'
430 PRINT"YOU HIT";NH;"TIMES."
435 IFNH=0THEN480
437 REM'IF THE PLAYER GOT NO HITS, DON'T ADD TO HIS WINNINGS'
440 IFNH=1THENBT=BT*2
450 IFNH=2THENBT=BT*3
460 IFNH=3THENBT=BT*4
465 REM'SET WINNINGS ACCORDING TO ODDS'
470 BM=BM+BT
475 REM'ADD WINNINGS TO PLAYER'S TOTAL HOLDINGS'
480 PRINT"YOU NOW HAVE";BM;"DOLLARS."
490 PRINT:PRINT

```

cont. on next page

Listing 4-4—cont. No-Frills Dice

```
495 IFBM=0THEN PRINT"SORRY, YOU'RE BUSTED! GAME'S OVER!":END
500 INPUT"WANT TO TRY AGAIN(Y/N)";X$
505 NH=0
507 REM'SET NUMBER OF HITS BACK TO ZERO'
510 IFX$="Y"THEN150
520 END
600 SC=D1+D2
605 REM SC IS THE TOTAL OF DIE 1 AND DIE 2
610 IFSC>7THEN700
620 IFSC<7THEN720
630 IFSC=7THEN740
640 REM'GO ACCORDING TO WHAT THE DICE TOTAL'
700 IFNM=1THENBT=BT#2:GOTO760
705 REM'IF THE PLAYER GUESSED CORRECTLY, ADD TO HIS BET'
710 GOTO770
715 REM'HE DID NOT GUESS CORRECTLY, SKIP ADDING TO HIS HOLDINGS'
720 IFNM=2THENBT=BT#2:GOTO760
730 GOTO770
740 IFNM=3THENBT=BT#3:GOTO760
750 GOTO770
760 BM=BM+BT
765 REM'ADD WINNINGS TO TOTAL HOLDINGS'
770 PRINT"YOU NOW HAVE";BM;"DOLLARS."
780 PRINT:PRINT
790 IF BM=0THEN PRINT"SORRY, YOU'RE BUSTED! GAME'S OVER!":END
800 INPUT"WANT TO PLAY AGAIN";Y$
810 IFY$="Y"THEN150
```

Sample Run of Program

A sample run of the Chuck-A-Luck program is shown in Fig. 4-2.

Suggested Modifications and Projects

1. Create some sprites that look like they are being “thrown across” the screen, just as real dice would appear.
2. Find some of your favorite dice games and create them using some of the routines within this program.

PROGRAM: SIMON SAYS

This graphics and sound game tests your memory (Listing 4-5). Simply type in the different colors you would like displayed, tell the computer how many different colors you think you can remember, and then enjoy the game!

Important Variables in Program

- S = Beginning of sound chip
B = Skill level



Fig. 4-2. Sample run of the Chuck-A-Luck program.

Listing 4-5. Simon Says

```

10 S=54272
20 REM'SET THE BEGINNING OF SOUND CHIP'
100 PRINTCHR$(147):PRINT:PRINT
110 PRINT"YOU MAY GET ACQUAINTED WITH THE SOUNDS"
120 PRINT"AND COLORS (1-4) AS MUCH AS YOU LIKE."
130 INPUT"ENTER 5 TO START GAME";P
135 POKES+24,15
137 REM'SET THE VOLUME'
140 IFP=1THEN2000
150 IFP=2THEN2100
160 IFP=3THEN2200
170 IFP=4THEN2300
175 REM'GO TO THE CHOSEN COLOR/SOUND'
180 P=0
200 POKE53281,6
205 REM'SET BACKGROUND BACK TO BLUE'
210 PRINT"HOW MANY EVENTS DO YOU WISH TO TRY TO PLACE IN ORDER"
220 INPUT B
300 FORX=1TOB
310 RN(X)=INT(RND(1)*4)+1
315 REM'GET "B" RANDOM NUMBERS'
320 IFRN(X)=1THEN2000
330 IFRN(X)=2THEN2100
340 IFRN(X)=3THEN2200
350 IFRN(X)=4THEN2300
355 REM'GO THE RANDOMLY SELECTED COLOR/SOUND'
360 IFX=BTHEN300
370 NEXTX
380 POKES+24,0
385 REM'TURN OFF SOUND'
390 POKE53281,6

```

cont. on next page

Listing 4-5 - cont. Simon Says

```
400 PRINT:PRINT:PRINT
410 PRINT"1. BLACK"
420 PRINT"2. RED"
430 PRINT"3. GREEN"
440 PRINT"4. ORANGE"
450 FORA=1TOB
455 REM'ASK "B" QUESTIONS'
460 INPUT"ENTER COLOR OF SCREEN NO.,";R
470 IFR<>RN<A>THENPRINT"WRONG":GOTO500
480 SC=SC+1
490 REM'ADD TO SCORE'
500 NEXTA
510 PRINT"YOUR SCORE IS";SC
520 NP=NP+1
525 REM'ADD TO THE NUMBER OF PLAYS'
530 IFNP=3THENEND
535 REM'END AFTER 3 PLAYS'
540 INPUT"HIT ANY KEY TO CONTINUE";T$
550 POKES+24,15
555 REM'TURN VOLUME BACK ON'
560 GOTO300
570 REM'DISPLAY ANOTHER SEQUENCE'
2000 PRINTCHR$(147)
2010 POKE53281,0
2020 REM'SET BACKGROUND TO BLACK'
2030 POKES+5,0:POKES+6,200
2035 REM'SET A/D/S/R'
2040 POKES+4,33
2045 REM'SET WAVEFORM'
2050 POKES+1,3:POKES,155
2055 REM'SET HIGH AND LOW FREQUENCY'
2060 FORD=1TO500:NEXTD
2065 REM'SET TIME DELAY'
2070 POKES+1,0:POKES,0
2075 REM'TURN OFF FREQUENCIES'
2080 POKES+4,0:IFP=1THEN130
2085 REM'TURN OFF WAVEFORM. GO TO 130 IF ONLY PRACTICING'
2090 GOTO360
2095 REM'GO TO NEXT X'
2100 PRINTCHR$(147)
2110 POKE53281,2
2115 REM'SET BACKGROUND TO RED'
2130 POKES+5,0:POKES+6,200
2140 POKES+4,17
2150 POKES+1,8:POKES,23
2160 FORD=1TO500:NEXTD
2170 POKES+1,0:POKES,0
2180 POKES+4,0:IFP=2THEN130
2190 GOTO360
2200 PRINTCHR$(147)
2210 POKE53281,5
2220 REM'SET BACKGROUND TO GREEN'
2230 POKES+5,50:POKES+6,50
2240 POKES+4,33
2250 POKES+1,8:POKES,147
2260 FORD=1TO500:NEXTD
2270 POKES+1,0:POKES,0
2280 POKES+4,0:IFP=3THEN130
2290 GOTO360
2300 PRINTCHR$(147)
2320 POKES+5,50:POKES+6,50
2330 POKES+4,17
2340 POKE53281,8
2345 REM'SET BACKGROUND TO ORANGE'
```

```

2350 POKES+1,19:POKES,63
2360 FORD=1T0500:NEXTD
2370 POKES+1,0:POKES,0
2380 POKES+4,0:IFP=4THEN130
2390 GOTO360

```

RN(x) = Array for random numbers
 X = Loop to change colors and sounds
 A = Loop to ask the screen color of "B" events
 SC = Player's score
 NP = Number of players (up to three)
 D = Delay loop for sustaining sound and colors
 T\$ = Input for continuing game

Explanation of Important Line Numbers Within Program

100-180 Determine sounds and colors player wants.
 200-220 Ask player if he wants to play the game.
 300-370 Direct the program to randomly selected sounds and colors.
 380-390 Turn off sound and change screen color back to blue.
 400-500 Ask player to repeat order of colors.
 510 Displays player's score.
 520-560 Check number of times player has played.
 2000-2260 Data to form different sounds and background colors.

Sample Run of Program

1. BLACK
2. RED
3. GREEN
4. ORANGE

ENTER COLOR OF SCREEN NO. 1? 3
 ENTER COLOR OF SCREEN NO. 2? 2
 ENTER COLOR OF SCREEN NO. 3? 4

YOUR SCORE IS 2
 HIT ANY KEY TO CONTINUE?

Suggested Modifications and Projects

1. Try to divide the screen into four sections, one section for each color, so that the whole screen is not filled with each color for the game play.

- Use skill levels to vary the speed with which the colors and sounds are shown to the player.

PROGRAM: 21 (BLACKJACK)

Here's a classic card game in which you try to get cards that add up to 21. Don't go over 21, or you "bust." However, do try to get as close to 21 as you can, since the computer is your shifty-eyed dealer! See Listing 4-6.

Listing 4-6. 21 (Blackjack)

```

40 NH=0:NC=1
50 DIMD$(53):DIMZ(53)
60 BM=100
70 REM'SET AMOUNT OF BETTING MONEY'
90 PRINT"SHUFFLING"
100 FORX=1TO52
110 REM'BEGIN SHUFFLING ROUTINE'
130 R1=INT(RND(1)*53)+0
155 REM'GET A RANDOM NUMBER FOR THE LOOP IN 200'
160 FORY=1TO52
165 REM'BEGIN ROUTINE THAT MAKES CERTAIN THAT EACH R1 VALUE IS USED ONLY ONCE'
170 IFZ(Y)=R1THEN150
180 NEXTY
190 Z(X)=R1
195 REM'KEEP RECORD OF THIS R1 SO IT WILL NOT BE USED AGAIN'
200 FORLA=1TOR1
210 READD$:NEXTLA
215 REM'READ A RANDOM AMOUNT OF DATA'
220 READD$(X)
225 REM THIS WILL BE CARD X IN THE SHUFFLED DECK
230 RESTORE:NEXTX:GOTO5500
240 REM'RESTORE DATA, GET NEXT CARD, GOTO 5500 WHEN SHUFFLING COMPLETE'
300 IFLEFT$(D$(CP),2)="AC"THEN9000
303 REM'LOOK AT THE FIRST 2 LETTERS IN THE NAME OF THE CARD AND GO SET VALUE'
305 IFLEFT$(D$(CP),2)="KI"THEN9020
310 IFLEFT$(D$(CP),2)="QU"THEN9020
315 IFLEFT$(D$(CP),2)="JA"THEN9020
320 IFLEFT$(D$(CP),2)="TE"THEN9020
325 IFLEFT$(D$(CP),2)="NI"THEN9100
330 IFLEFT$(D$(CP),2)="EI"THEN9120
335 IFLEFT$(D$(CP),2)="SE"THEN9140
340 IFLEFT$(D$(CP),2)="SI"THEN9160
345 IFLEFT$(D$(CP),2)="FI"THEN9180
350 IFLEFT$(D$(CP),2)="FO"THEN9200
355 IFLEFT$(D$(CP),2)="TH"THEN9220
360 IFLEFT$(D$(CP),2)="TW"THEN9240
5000 DATAND,ACE OF SPADES,KING OF SPADES,QUEEN OF SPADES,JACK OF SPADES
5005 REM'THE FIRST PIECE OF DATA WILL NOT BE USED'
5010 DATATEN OF SPADES,NINE OF SPADES,EIGHT OF SPADES,SEVEN OF SPADES
5020 DATASIX OF SPADES,FIVE OF SPADES,FOUR OF SPADES
5030 DATATHREE OF SPADES,TWO OF SPADES
5040 DATARACE OF DIAMONDS,KING OF DIAMONDS,QUEEN OF DIAMONDS,JACK OF DIAMONDS
5050 DATATEN OF DIAMONDS,NINE OF DIAMONDS,EIGHT OF DIAMONDS
5060 DATASEVEN OF DIAMONDS,SIX OF DIAMONDS,FIVE OF DIAMONDS
5070 DATAFOUR OF DIAMONDS,THREE OF DIAMONDS,TWO OF DIAMONDS
5080 DATARACE OF HEARTS,KING OF HEARTS,QUEEN OF HEARTS,JACK OF HEARTS
5090 DATATEN OF HEARTS,NINE OF HEARTS,EIGHT OF HEARTS,SEVEN OF HEARTS
5100 DATASIX OF HEARTS,FIVE OF HEARTS,FOUR OF HEARTS
5110 DATATHREE OF HEARTS,TWO OF HEARTS

```

```

5120 DATAACE OF CLUBS,KING OF CLUBS,QUEEN OF CLUBS,JACK OF CLUBS
5130 DATATEN OF CLUBS,NINE OF CLUBS,EIGHT OF CLUBS
5140 DATASEVEN OF CLUBS,SIX OF CLUBS,FIVE OF CLUBS
5150 DATAFOUR OF CLUBS,THREE OF CLUBS,TWO OF CLUBS
5500 PRINT:PRINT"YOU HAVE";BM;"DOLLARS."
5501 PRINT
5503 IFBM=0THEN PRINT"SORRY, YOU'RE BUSTED! GAME'S OVER!":END
5504 REM'SEE IF THE PLAYER LOST ALL OF HIS MONEY'
5505 IFNC>40THENPRINT"RE-SHUFFLING":GOTO12110
5510 INPUT"PLEASE PLACE A BET";BT
5512 PRINT
5515 IF BT<=BM THEN 6000
5520 IF BT>BM THEN PRINT"YOU DON'T HAVE THAT MUCH MONEY"
5525 PRINT"BET AGAIN":PRINT:GOTO 5510
5530 REM'IF HE BETS MORE THAN HE HAS, GO ASK AGAIN'
6000 PRINTCHR$(147):PRINT:PRINT
6010 P1=0:P2=0:NH=0:H2=0
6015 A1=0:A2=0
6020 PRINT"YOUR CARDS ARE:"
6030 PRINTD$(NC):CP=NC:GOSUB300
6035 REM'DEAL THE FIRST CARD IN THE HAND,GO CHECK ITS VALUE'
6040 PRINTD$(NC+1):CP=NC+1:GOSUB300
6045 REM'DEAL THE SECOND ONE'
6050 IFP1=21THEN13000
6055 REM'SEE IF THE PLAYER GOT A BLACKJACK'
6100 PRINT"I HAVE:"
6110 PRINTD$(NC+2):CP=NC+2:C2=1:GOSUB300
6115 REM'DEAL AND SHOW THE DEALER'S FIRST CARD'
6120 PRINT"??????"
6130 REM THE DEALER'S SECOND CARD IS NOT SHOWN YET
6140 NC=NC+3
6145 REM 3 CARDS HAVE BEEN DEALT
6150 INPUT"DO YOU WANT A HIT(Y/N)";I1$
6160 IFI1$<"Y"THEN8500
6210 PRINTD$(NC+NH):NH=NH+1
6215 REM'DEAL NEXT CARD, MARK THAT ANOTHER CARD WAS DEALT'
6220 CP=NC+NH-1:GOSUB300
6230 REM'GO FIND AND ADD THIS CARD'S VALUE TO THE TOTAL, P1'
6240 IFP1>21THEN7500
6241 REM'SEE IF THE PLAYER BUSTED OR IF HE HAS ACES'
6242 GOTO6150
7500 IFA1=0THEN7600
7505 REM'IF HE HAS NO ACES HE BUSTED'
7510 IFA1>0THENP1=P1-10:A1=A1-1:GOTO6240
7600 PRINT"YOU BUSTED"
7610 BM=BM-BT
7620 REM'SUBTRACT HIS BET'
7800 NC=NC+NH:GOTO5500
8500 PRINT"I ALSO HAVE:"
8510 NC=NC+NH
8520 REM'GO TO THE NEXT CARD'
8540 PRINTD$(NC):CP=NC:C2=1:GOSUB300
8543 REM'SHOW CARD, GET ITS VALUE, MARK THAT THE CARD-'
8544 REM'BEING VALUED IS THE DEALER'S'
8545 H2=H2+1
8547 REM'MARK THAT THE DEALER TOOK ANOTHER CARD'
8550 IFP2<P1ORP2=P1THEN8600
8555 REM'SEE IF THE DEALER NEEDS A HIT'
8560 PRINT"I WIN."
8565 BM=BM-BT
8570 IFNC>40THEN12000
8575 REM'SEE IF THERE ARE ENOUGH CARDS LEFT'
8580 NC=NC+H2:GOTO5500
8600 NC=NC+H2
8610 PRINTD$(NC):CP=NC:H2=H2+1:C2=1:GOSUB300
8613 REM'SHOW THE CARD, ETC.'

```

cont. on next page

Listing 4-6 - cont. 21 (Blackjack)

```
8615 IFP2>21THEN8800
8617 REM/SEE IF THE DEALER BUSTED OR IF HE HAS ACES'
8620 IFP2>16THEN10000
8625 REM THE DEALER MUST STAND ON 17 OR HIGHER
8630 GOTO8550
8800 IFA2=0THEN8900
8805 REM/IF THE DEALER HAS NO ACES HE BUSTED'
8810 IFA2>0THENP2=P2-10
8815 REM/SET THE VALUE OF ONE ACE FROM 11 TO 1'
8820 A2=A2-1:GOTO8620
8825 REM ONE ACE WAS USED
8900 PRINT"YOU WIN"
8905 BM=BM+BT
8910 REM/ADD PLAYER'S BET TO HIS TOTAL'
8920 NC=NC+H2:GOTO5500
9000 IFC2=1THEN9015
9005 REM/IF A DEALER'S CARD IS BEING VALUED SKIP THE NEXT LINE'
9010 P1=P1+11:A1=A1+1:RETURN
9013 REM/AN ACE MEANS 11 POINTS ADDED TO THE TOTAL'
9014 REM/MARK THAT THE DEALER HAS AN ACE,SET OTHER MARKER BACK TO 0'
9015 P2=P2+11:A2=A2+1:C2=0:RETURN
9020 IFC2=1THEN9040
9030 P1=P1+10:RETURN
9040 P2=P2+10:C2=0:RETURN
9100 IFC2=1THEN9115
9110 P1=P1+9:RETURN
9115 P2=P2+9:C2=0:RETURN
9120 IFC2=1THEN9135
9130 P1=P1+8:RETURN
9135 P2=P2+8:C2=0:RETURN
9140 IFC2=1THEN9155
9150 P1=P1+7:RETURN
9155 P2=P2+7:C2=0:RETURN
9160 IFC2=1THEN9175
9170 P1=P1+6:RETURN
9175 P2=P2+6:C2=0:RETURN
9180 IFC2=1THEN9195
9190 P1=P1+5:RETURN
9195 P2=P2+5:C2=0:RETURN
9200 IFC2=1THEN9215
9210 P1=P1+4:RETURN
9215 P2=P2+4:C2=0:RETURN
9220 IFC2=1THEN9235
9230 P1=P1+3:RETURN
9235 P2=P2+3:C2=0:RETURN
9240 IFC2=1THEN9255
9250 P1=P1+2:RETURN
9255 P2=P2+2:C2=0:RETURN
10000 IFP2<P1 OR P2=P1THEN10100
10010 PRINT"I WIN"
10015 BM=BM-BT
10020 NC=NC+H2
10040 GOTO5500
10100 PRINT"YOU WIN"
10110 BM=BM+BT
10300 NC=NC+H2:GOTO5500
12000 PRINT"SORRY, THIS HAND IS INVALID."
12100 PRINT"I MUST RE-SHUFFLE."
12110 FORX2=1TO52
12120 Z(X2)=0
12125 REM/SET ALL Z(X) VALUES BACK TO 0'
12130 NEXTX2
```

```
12150 NC=1
12175 REM 'START BACK AT CARD ONE AFTER SHUFFLING'
12200 GOTO100
13000 PRINT:PRINT
13100 PRINT"BLACKJACK!"
13200 NC=NC+2
13250 REM 2 CARDS WERE USED
13300 BM=BM+(BT*1.5)
13350 REM'MULTIPLY THE PLAYER'S BET'
13400 GOTO5500
```

Important Variables in Program

NH = Number of hits

NC = Number of card being played

D\$(x) = Card

Z(x) = Used when shuffling to make sure random number was not used more than once

X = Loop to read cards randomly

R1 = Random number to read random amount of data

Y = Loop used to check Z(x)

Z(y) = Used in the "Y" loop

LA = Loop to read R1 data

BM = Money with which to bet

H2 = Number of hits for dealer

BT = The bet

P1 = Total value of player's hand

P2 = Total value of the dealer's hand

A1 = Total number of aces in player's hand

A2 = Total number of aces in dealer's hand

CP = Set to NC to find the value of the card

J1\$ = Player's decision for a hit or not

Explanation of Important Lines Within Program

40-60 Set up variables.

100-230 Card shuffling routine.

300-360 Define the value of each card.

5000-5150 Data.

5500-5530 Ask the player for his bet.

6010-6015 Set certain variables.

6020-6055 Deal the player's first two cards.

6100-6110 Display the dealer's first card.

6150-6240 Give player any hits he wants.

7500-7800 Check if player got over 21 or if he has aces.

8500-8545 Get the dealer's other card.

8550-8630 Give dealer any needed hits.

8800-8820 Check if the dealer needs ace values.

9000-9255 Set a value for each card.

10000-13400 Check to see who won.

Sample Run of Program

```
YOUR CARDS ARE  
SIX OF SPADES  
KING OF SPADES  
I HAVE:  
JACK OF CLUBS  
????????????  
DO YOU WANT A HIT? Y  
QUEEN OF HEARTS  
YOU BUSTED  
YOU HAVE 70 DOLLARS  
PLEASE PLACE A BET?
```

Suggested Changes and Modifications

1. See if you can make graphics work with this program so that cards are actually displayed on the screen.
2. Try to program your own favorite card game into your Commodore 64, using some of the routines found in this program.

Chapter 5

Educational Programs

Computers are being used more for education than anyone really expected. Because the computer is fast, efficient, has a large amount of memory, and doesn't hurt a person's feelings, the computer is praised for being an excellent teaching device. In this chapter there are five programs for the Commodore 64 that teach subjects from Geometry to French Cuisine vocabulary.

PROGRAM: HISTORY

This program not only teaches a person about history, but also gives quizzes. This is accomplished by quizzing a person about certain events and the order in which they occurred, and then giving the right answer in case the person misses the question. See Listing 5-1.

Listing 5-1. History

```
10 SC=0:NB=0
15 REM'SET VARIABLES'
100 PRINT"Q":PRINT:PRINT
110 PRINT"  PLACE THESE EVENTS IN CHRONOLOGICAL ORDER."
120 READA$,B$,C$
```

cont. on next page

Listing 5-1—cont. History

```

125 REM/READ FIRST 3 PIECES OF DATA'
130 PRINT:PRINT
140 RN=INT(RND(1)*6)+1
150 IFRN=1THENPRINTA$:PRINTB$:PRINTC$
160 IFRN=2THENPRINTB$:PRINTC$:PRINTA$
170 IFRN=3THENPRINTC$:PRINTA$:PRINTB$
180 IFRN=4THENPRINTA$:PRINTC$:PRINTB$
190 IFRN=5THENPRINTB$:PRINTA$:PRINTC$
200 IFRN=6THENPRINTC$:PRINTB$:PRINTA$
210 REM/PICK A RANDOM ORDER TO DISPLAY THE DATA'
250 PRINT:PRINT
300 INPUT"WHICH HAPPENED FIRST";D$
305 NB=NB+1
310 IFD$<>A$THEN800
315 REM NOTICE THAT 330 WOULD BE SKIPPED
320 PRINT:PRINT"CORRECT"
330 SC=SC+1
335 REM/ADD ANOTHER POINT'
340 INPUT"WHICH HAPPENED SECOND";E$
345 NB=NB+1
350 IFE$<>B$THEN840
360 PRINT:PRINT"CORRECT"
370 SC=SC+1
375 FORT1=1T0600:NEXTT1
380 GOTO700
390 REM/WAIT, GO ON TO 700'
700 READF$,G$,H$
701 REM/READ THE NEXT 3 PIECES OF DATA ( THE DATES )'
703 PRINT"?"
707 PRINTSPC(5)"HERE IS THE CORRECT ORDER WITH"
708 PRINT"RESPECTIVE DATES."
710 PRINT:PRINT:PRINT
720 PRINTA$:PRINTF$
730 PRINTB$:PRINTG$
740 PRINTC$:PRINTH$
745 REM/SHOW THE PLAYER THE DATES'
750 PRINT:PRINT:PRINT
755 FORT2=1T0900:NEXTT2
757 IFNB=40THEN900
758 REM THERE ARE ONLY 40 QUESTIONS
760 INPUT"DO YOU WISH TO CONTINUE(Y/N)";I$
770 IFI$="Y"THEN100
775 REM/IF YES, PRINT ANOTHER QUESTION'
780 GOTO900
785 'IF NO, GO TO 900'
800 PRINT:PRINT:PRINT
810 PRINTSPC(3)"WRONG. ";A$;" OCCURED FIRST."
820 GOTO340
830 REM/ASK THE SECOND PART OF THE QUESTION'
840 PRINTSPC(5)"WRONG. ";B$;" OCCURED SECOND."
845 FORT3=1T0900:NEXTT3
850 GOTO700
860 REM/GO SHOW THE CORRECT ANSWERS AND DATES'
900 PRINT"?:":PRINT:PRINT:PRINT
910 PRINT"YOU GOT";SC;"QUESTIONS CORRECT OUT OF";NB;".
920 PRINT:PRINT:PRINT
930 PRINT"YOUR PERCENTAGE IS ";(SC/NB)*100;".
5000 DATA FRENCH AND INDIAN WAR,AMERICAN REVOLUTION,WAR OF 1812
5005 DATA 1755,1776,1812
5010 DATA CIVIL WAR,EMANCIPATION PROCLAMATION,FOURTEENTH AMENDMENT
5015 DATA 1861,1863,1866
5020 DATA LOUISIANA PURCHASE,WAR OF 1812,JACKSON ADMINISTRATION
5025 DATA 1803,1812,1820
5030 DATA MEXICAN WAR,SPANISH-AMERICAN WAR,WORLD WAR I

```

5035 DATA1846,1898,1914
 5040 DATADECLARATION OF INDEPENDENCE,ARTICLES OF CONFEDERATION,CONSTITUTION
 5045 DATA1776,1781,1787
 5050 DATABATTLE OF YORKTOWN,BATTLE OF THE MARNE,BATTLE OF THE BULGE
 5055 DATA1781,1914,1944
 5060 DATATREATY OF GHENT,WORLD WAR I,TREATY OF VERSAILLES
 5065 DATA1814,1914,1919
 5070 DATABOSTON TEA PARTY,INTOLERABLE ACTS,DECLARATION OF INDEPENDENCE
 5075 DATA1773,1774,1776
 5080 DATACARTHAGE FOUNDED,FIRST OLYMPIC GAMES,ROME FOUNDED
 5085 DATA800 BC,776 BC,753 BC
 5090 DATADEATH OF JULIUS CAESAR,JESUS BORN,BRITAIN TAKEN BY ROME
 5095 DATA44 BC,4 BC,84 BC
 5100 DATADEATH OF ATTILA THE HUN,FALL OF ROME,BIRTH OF MUHAMMAD
 5105 DATA453,455,570
 5110 DATACLOVIS-FIRST KING OF FRANCE,CHARLEMAGNE'S EMPIRE
 5120 DATA EGBERT BECOMES FIRST KING OF ENGLAND
 5125 DATA481,800,828
 5130 DATANDROMA INVASION OF ENGLAND,JENGIS KHAN TAKES CHINA
 5140 DATA KUBLAI KHAN BECOMES GREAT KHAN
 5145 DATA1066,1214,1260
 5150 DATASIR WALTER RALEIGH EXPLORES VIRGINIA,JAMESTOWN FOUNDED,MAYFLOWER
 5155 DATA1583,1607,1620
 5160 DATASEVEN YEARS' WAR BEGINS,BRITISH GET CANADA FROM FRANCE
 5170 DATA AMERICAN REVOLUTION BEGINS
 5175 DATA1755,1763,1776
 5180 DATACONSTITUTION SIGNED,RAID OF THE BASTILLE,LOUIS XVI BEHEADED
 5185 DATA1787,1789,1793
 5190 DATANAPOLEAN FIGHTS IN EGYPT,NAPOLEAN BECOMES EMPEROR
 5200 DATA NAPOLEAN FIGHTS IN RUSSIA
 5205 DATA1798,1804,1812
 5210 DATAABDICATION OF NAPOLEAN,WATERLOO,TREATY OF VIENNA
 5215 DATA1814,1815,1815
 5220 DATA CRIMEAN WAR,CIVIL WAR,BOER WAR
 5225 DATA1854,1861,1899
 5230 DATARUSSO-JAPANESE WAR,WORLD WAR I,RUSSIAN REVOLUTION
 5235 DATA1904,1914,1917

Important Variables in Program

SC = Number of questions right

NB = Number of questions answered

RN = Random number to determine the order the data is to be printed

A\$,B\$,C\$ = History data

D\$ = Input for "what happened first?"

E\$ = Input for "what happened second?"

F\$, G\$, H\$ = Date data

T1, T2, T3 = Delay loops

Explanation of Important Line Numbers Within Program

10-120 Read three pieces of data.

130-200 Display data in random order.

210-390 Ask what event happened first and which happened second.

700-740 Read and print dates after each event, printed in the correct order.

800-850 Tell player the answer if incorrectly answered.

900-930 Tell player his or her score.

5000-5235 Data of events.

Sample Run of Program

**WAR OF 1812
FRENCH AND INDIAN WAR
AMERICAN REVOLUTION**

**WHICH HAPPENED FIRST? FRENCH AND INDIAN WAR
CORRECT**

**WHICH HAPPENED SECOND? AMERICAN REVOLUTION
CORRECT**

Suggested Modifications and Projects

1. Add some other events in history to the data in this program.
2. Tell the user to input a skill level, from novice to expert, which makes the computer read a certain set of data. The novice should have the simple events of history, while the expert would be presented the lesser-known events.

PROGRAM: PLACES & CAPITALS

If you have never really learned which capitals are in which states and countries, this program should help. See Listing 5-2.

Listing 5-2. Places & Capitals

```
3 DIM Z(77)
4 REM A DIMENSION IS NEEDED FOR Z(X)
5 PRINTCHR$(147):PRINTSPC(6)"PLACES AND THEIR CAPITALS"
7 SC=0
8 REM SC WILL BE USED TO KEEP THE SCORES IN QUIZES
10 PRINT:PRINT:PRINTSPC(16)"MENU"
15 PRINT:PRINT
20 PRINTSPC(5)"1. FIND THE CAPITAL"
25 PRINT
30 PRINTSPC(5)"2. FIND THE STATE OR COUNTRY"
35 PRINT
40 PRINTSPC(5)"3. QUIZ - GIVEN THE PLACE, NAME"
43 PRINTSPC(8)"THE COUNTRY."
45 PRINT
50 PRINTSPC(5)"4. QUIZ - GIVEN THE CAPITAL, NAME"
53 PRINTSPC(8)"THE PLACE."
55 PRINT:PRINT:PRINT:PRINT
60 INPUT"PLEASE CHOOSE 1,2,3 OR 4";I1
63 RESTORE
65 IF I1=1 THEN 300
```

```

70 IF I1=2 THEN 500
75 IF I1=3 THEN 700
80 IF I1=4 THEN 700
85 REM GO TO THE USER'S CHOICE
90 GOTO 60
95 REM IF I1 NOT 1,2,3, OR 4 THEN ASK AGAIN
100 DATA ALABAMA, BIRMINGHAM, ALASKA, JUNEAU
103 DATA ARIZONA, PHOENIX, ARKANSAS, LITTLE ROCK
105 DATA CALIFORNIA, SACRAMENTO, COLORADO, DENVER, CONNECTICUT, HARTFORD
110 DATA DELAWARE, DOVER, FLORIDA, TALLAHASSEE
113 DATA GEORGIA, ATLANTA, HAWAII, HONOLULU
115 DATA IDAHO, BOISE, ILLINOIS, SPRINGFIELD, INDIANA
117 DATA INDIANAPOLIS, IOWA, DES MOINES
120 DATA KANSAS, TOPEKA, KENTUCKY, FRANKFORT, LOUISIANA
123 DATA BATON ROUGE, MAINE, AUGUSTA
125 DATA MARYLAND, ANNAPOLIS, MASSACHUSETTS, BOSTON
127 DATA MICHIGAN, LANSING, MINNESOTA
130 DATA ST. PAUL, MISSISSIPPI, JACKSON, MISSOURI
133 DATA JEFFERSON CITY, MONTANA, HELENA
135 DATA NEBRASKA, LINCOLN, NEVADA, CARSON CITY
137 DATA NEW HAMPSHIRE, CONCORD, NEW JERSEY
140 DATA TRENTON, NEW MEXICO, SANTA FE, NEW YORK
143 DATA ALBANY, NORTH CAROLINA, RALEIGH
145 DATA NORTH DAKOTA, BISMARCK, OHIO, COLUMBUS
147 DATA OKLAHOMA, OKLAHOMA CITY, OREGON, SALEM
150 DATA PENNSYLVANIA, HARRISBURG, RHODE ISLAND
153 DATA PROVIDENCE, SOUTH CAROLINA, COLUMBIA
155 DATA SOUTH DAKOTA, PIERRE, TENNESSEE, NASHVILLE
157 DATA TEXAS, AUSTIN, UTAH, SALT LAKE CITY
160 DATA VERMONT, MONTPELIER, VIRGINIA, RICHMOND, WASHINGTON, OLYMPIA
165 DATA WEST VIRGINIA, CHARLESTON, WISCONSIN
170 DATA MADISON, WYOMING, CHEYENNE
180 DATA AFGANISTAN, KABUL, ARGENTINA, BUENOS AIRES
195 DATA AUSTRALIA, CANBERRA, AUSTRIA, VIENNA
200 DATA BELGIUM, BRUSSELS, BRAZIL, BRASILIA
205 DATA CANADA, OTTAWA, COLOMBIA, BOGOTA
210 DATA CUBA, HAVANA, DENMARK, COPENHAGEN
215 DATA FRANCE, PARIS, EAST GERMANY, BERLIN
220 DATA WEST GERMANY, BONN, GREECE, ATHENS
225 DATA IRAN, TEHRAN, IRAQ, BAGHDAD
230 DATA IRELAND, DUBLIN, ITALY, ROME
235 DATA JAPAN, TOKYO, NORWAY, OSLO
240 DATA PORTUGAL, LISBON, SOVIET UNION, MOSCOW
245 DATA SPAIN, MADRID, GREAT BRITAIN, LONDON
300 PRINT CHR$(147):PRINT:PRINT:PRINT
310 PRINT "NAME THE STATE OR COUNTRY AND I'LL FIND"
320 INPUT "ITS CAPITAL"; I1$
330 FOR L1=1 TO 74: READD1$, D2$
340 IF D1$=I1$ THEN 400
350 NEXT L1
355 REM THE L1 LOOP READS DATA, 2 AT A TIME, AND SEES IF THE FIRST PIECE-
356 REM OF DATA IS THE PLACE NAMED
360 PRINT:PRINT:PRINT "I AM SORRY, THAT PLACE DOES NOT EXIST"
370 PRINT "IN MY MEMORY BANKS."
375 REM THE PLACE NAMED IS NOT IN THE DATA IF THE L1 LOOP IS COMPLETED
380 FOR T1=1 TO 1500: NEXT T1: GOTO 5
390 REM WAIT A WHILE, RETURN TO MENU
400 PRINT "THE CAPITAL OF "; I1$; " IS "; D2$
405 REM NOW THE USER HAS THE INFO HE WANTED
410 FOR T2=1 TO 1500: NEXT T2: GOTO 5
420 REM WAIT, RETURN TO MENU
500 PRINT CHR$(147):PRINT:PRINT:PRINT
505 REM THIS IS THE BEGINNING OF SELECTION #2
510 PRINT "NAME THE CAPITAL AND I'LL FIND THE"
515 INPUT "PLACE"; A$
520 FOR L3=1 TO 74: READD3$, D4$

```

Listing 5-2—cont. Places & Capitals

```
530 IFD4$=AA$THEN600
540 NEXTL3
545 REM THIS LOOP HAS THE SAME PURPOSE AS THE ONE AT 330
550 PRINT:PRINT:PRINT"I AM SORRY. THAT CAPITAL DOES NOT EXIST"
560 PRINT"IN MY MEMORY BANKS."
570 FORT3=1TO1500:NEXTT3:GOTO5
600 PRINTAA$;" IS THE CAPITAL OF ";D3$
610 FORT4=1TO1500:NEXTT4:GOTO5
620 REM END OF SELECTION 2 SUBROUTINE. IT IS SIMILAR TO LINES 300-420
700 FORL4=1TO74:RESTORE
703 REM THIS SUBROUTINE SERVES SELECTIONS 3 AND 4
705 IFL4=74THEN1200
707 REM THERE ARE ONLY 74 PAIRS OF DATA
710 R1=INT(RND(1)*74)+0:D5=(R1*2)
713 REM A PAIR WILL BE READ AT RANDOM. D5 IS USED IN 730
715 FORL6=1TO74
720 IFR1= Z(L6)THEN710
722 REM IF THIS RANDOM NUMBER WAS ALREADY USED, GO BACK AND GET ANOTHER
723 NEXTL6
724 REM THIS LOOP CHECKS ALL POSSIBLE VALUES FOR R1
725 Z(L4)=R1
726 REM THIS R1 IS STORED SO IT WILL NOT BE USED AGAIN
730 FORL5=1TO D5
740 READD7$:NEXTL5
745 REM 'R1*2' DATA STATEMENTS ARE SHOVED ASIDE
750 READD5$,D6$
753 REM THIS IS THE DATA WE WANT
755 IFI1=4THEN1010
757 REM 'IF THE SECOND QUIZ WAS CHOSEN, GOTO1010'
760 PRINTCHR$(147)
765 PRINT"TO END INPUT 'Q'"
767 PRINT:PRINT:PRINT
770 PRINT"WHAT IS THE CAPITAL OF ";D5$;"?"
780 INPUT " ";AB$
783 REM THE QUESTION IS ANSWERED HERE
785 IFAB$="Q"THEN1200
787 REM THE PLAYER OPTED TO QUIT
790 IFAB$<>D6$THEN020
795 REM 'IF THE ANSWER IS INCORRECT GOTO 020'
800 PRINT:PRINT:PRINT"CORRECT."
810 FORT5=1TO400:NEXTT5:SC=SC+1:NEXTL4
815 REM WAIT, ADD POINT TO SCORE, GO BACK FOR ANOTHER QUESTION
820 PRINT:PRINT:PRINT"WRONG."
830 PRINT:PRINT"THE CAPITAL OF ";D5$;" IS ";D6$
835 REM THE CORRECT ANSWER IS GIVEN
840 FORT6=1TO600:NEXTT6:NEXTL4
850 REM WAIT A WHILE, GO BACK FOR ANOTHER QUESTION
1000 REM THE SECOND QUIZ WAS CHOSEN. GO GET SOME RANDOM DATA
1010 PRINTCHR$(147)
1013 REM SELECTION 4 BEGINS ITS SUBROUTINE HERE
1014 REM IT IS VERY SIMILAR TO THE LAST ONE AT 700
1015 PRINT"TO END INPUT 'Q'."
1017 PRINT:PRINT:PRINT
1020 PRINTD6$;" IS THE CAPITAL OF WHAT PLACE?"
1025 INPUT " ";AC$
1030 IFAC$="Q"THEN1200
1040 IFAC$<>D5$THEN1070
1050 PRINT:PRINT:PRINT"CORRECT"
1060 FORT6=1TO600:NEXTT6:SC=SC+1:NEXTL4
1070 PRINT:PRINT:PRINT"WRONG"
1080 PRINTD6$;" IS THE CAPITAL OF ";D5$;"."
1090 FORT7=1TO600:NEXTT7:NEXTL4
1200 PRINTCHR$(147):PRINT:PRINT:PRINT
```

```
1210 PRINT"YOU GOT";SC;"QUESTIONS CORRECT OUT OF";(L4-1);"."
1220 PRINT:PRINT"YOUR PERCENTAGE IS";SC/(L4-1)*100;".%"
1225 REM(L4-1) IS THE TOTAL NUMBER OF QUESTIONS ATTEMPTED
1230 FORT0=1T02100:NEXTT0:RUN
1235 REM 1200-1230 SERVES BOTH QUIZZES
```

Important Variables in Program

Z(X) Used to make sure random number is not read twice in a quiz

SC = Score for quizzes

D1\$, D2\$, D3\$, D4\$ = Data

L1 = Loop to find capital

T1,T2 = Delay loop

AA\$ = Input

L3 = Loop to find state or country

R1 = Random scheme for quizzes one and two

L6 = Loop to see if random number has been used already

AB\$ = Input in first quiz

AC\$ = Input for second quiz

Explanation of Important Lines Within Program

3-60 Set variables and print menu.

65-90 Go to the desired subroutine.

100-245 Data.

300-410 Find and display requested data for choice one.

500-610 Find and display requested data for choice two.

700-750 Read data at random.

760-810 Ask questions for the first quiz using the data from 700-750.

820-840 Show the correct response for the first quiz.

900-940 Same function as 700-740, but for quiz two.

1010-1060 Same function as 760-810, but for quiz two.

1070-1090 Same function as 820-840, but for quiz two.

1210-1230 Give the player's score for any quiz.

Sample Run of Program

MENU

1. FIND THE CAPITAL
 2. FIND THE STATE OR COUNTRY
 3. QUIZ — GIVEN THE PLACE, NAME THE COUNTRY.
 4. QUIZ — GIVEN THE CAPITAL, NAME THE PLACE.
- PLEASE CHOOSE 1, 2, 3, OR 4?

Suggested Modifications and Projects

1. If there are some countries not included in the data which you would like to be there, add those countries with their capitals, and make the appropriate changes to the program.
2. See if you can make a sprite library of high-resolution images of countries, states, and the respective locations of their capitals.

PROGRAM: METRIC CONVERTER

Because the metric system is a more efficient, logical, and international system than our own "normal" measurement system, people should begin learning more about it and what the terms of metric measurement mean. This program should help, since it converts units of measurement both ways, from English to Metric as well as Metric to English. See Listing 5-3.

Listing 5-3. Metric Converter

```

100 PRINTCHR$(147):PRINT:PRINT
110 PRINTSPC(11)"METRIC CONVERTER"
120 PRINT:PRINT
130 PRINT"1. ENGLISH TO METRIC"
140 PRINT"2. ENGLISH FROM METRIC"
150 PRINT:PRINT
160 INPUT"PLEASE CHOOSE ONE";CH
170 IFCH=1THENA$="TO":GOTO210
200 A$="FROM"
210 PRINTCHR$(147):PRINT:PRINT
220 PRINT"1. INCHES ";A$;" CENTIMETERS"
230 PRINT"2. FEET ";A$;" METERS"
240 PRINT"3. MILES ";A$;" KILOMETERS"
250 PRINT"4. OUNCES ";A$;" GRAMS"
270 PRINT:PRINT
280 INPUT"YOUR CHOICE";C2
290 IFCH=1THEN400
300 IFCH=2THEN500
350 REM'GO TO THE USER'S CHOICE OF SYSTEM TO CONVERT TO'
400 PRINTCHR$(147):PRINT:PRINT
410 IFC2=1THEN600
420 IFC2=2THEN650
430 IFC2=3THEN700
440 IFC2=4THEN750
450 REM'GO TO THE USER'S CHOICE OF UNIT TO CONVERT TO'
500 PRINTCHR$(147):PRINT:PRINT
510 IFC2=1THEN900
520 IFC2=2THEN950
530 IFC2=3THEN1000
540 IFC2=4THEN1050
600 PRINT"ENTER THE AMOUNT IN INCHES TO BE"
610 INPUT"CHANGED TO CENTIMETERS";E1
620 M1=E1#2.54
625 REM THIS FORMULA FINDS THE METRIC EQUIVALENT OF THE ENGLISH VALUE
630 PRINT:PRINTE1;"INCHES IS EQUAL TO";M1;"CENTIMETERS"

```

```

640 GOTO2000
650 PRINT"ENTER THE AMOUNT IN FEET TO BE"
660 INPUT"CHANGED TO METERS";E2
670 M2=E2/3.281
680 PRINT:PRINT E2;"FEET IS EQUAL TO";M2;"METERS"
690 GOTO2000
700 PRINT"ENTER THE AMOUNT IN MILES TO BE"
710 INPUT"CHANGED TO KILOMETERS";E3
720 M3=E3*1.6093
730 PRINT:PRINT E3;"MILES IS EQUAL TO";M3;"KILOMETERS"
740 GOTO2000
750 PRINT"ENTER THE AMOUNT IN OUNCES TO BE"
760 INPUT"CHANGED TO GRAMS";E4
770 M4=E4*28.35
780 PRINT:PRINT E4;"OUNCES IS EQUAL TO";M4;"GRAMS"
790 GOTO2000
900 PRINT"ENTER THE AMOUNT IN CENTIMETERS"
910 INPUT"TO BE CHANGED TO INCHES";M1
920 E1=M1*.3937
925 REM THIS FORMULA FINDS THE ENGLISH EQUIVALENT OF THE METRIC VALUE
930 PRINT:PRINT M1;"CENTIMETERS IS EQUAL TO";E1;"INCHES"
940 GOTO2000
950 PRINT"ENTER THE AMOUNT IN METERS TO BE"
960 INPUT"CHANGED TO FEET";M2
970 E2=M2*3.218
980 PRINT:PRINT M2;"METERS IS EQUAL TO";E2;"FEET"
990 GOTO2000
1000 PRINT:PRINT"ENTER THE AMOUNT IN KILOMETERS"
1010 INPUT"CHANGED TO MILES";M3
1020 E3=M3*.6215
1030 PRINT:PRINT M3;"KILOMETERS IS EQUAL TO";E3;"MILES"
1040 GOTO2000
1050 PRINT"ENTER THE AMOUNT IN GRAMS TO BE"
1060 INPUT"CHANGED TO OUNCES";M4
1070 E4=M4/28.35
1080 PRINT:PRINT M4;"GRAMS IS EQUAL TO";E4;"OUNCES"
1090 GOTO2000
2000 PRINT:PRINT
2010 INPUT"HIT ANY KEY TO CONTINUE";TL$
2020 GOTO100
2030 REM WHEN THE USER IS FINISHED READING GO TO THE BEGINNING'

```

Important Variables in Program

- CH = Choice of conversion
- C2 = Choice of specific unit of measure to convert
- E1 = Amount for inches
- M1 = Amount for centimeters
- E2 = Amount for feet
- M2 = Amount for meters
- E3 = Amount for miles
- M3 = Amount for kilometers
- E4 = Amount for ounces
- M4 = Amount for grams
- TL\$ = 'Continue when any key is hit' string

Explanation of Line Numbers Within Program

100-150 Print the menu on the screen.

- 160 Gets input.
- 170-200 Set A\$ to "to" or "from."
- 210-250 Print a second menu.
- 290-540 Direct program to subroutine according to the user's choices.
- 600-1090 Do various calculations to give the user the information needed.

Sample Run of Program

```
ENTER AMOUNT IN INCHES TO BE CHANGED TO CENTI-  
METERS? 10  
10 INCHES IS EQUAL TO 25.4 CENTIMETERS  
HIT ANY KEY TO CONTINUE?
```

Suggested Modifications and Projects

1. There are other metric measures for things such as volume and temperature. Try adding routines for these conversions.
2. See if you can use graphics to make the metric conversion more visible — for instance, you could show a "ruler" marked off in meters, and for comparison, another "ruler" marked off in feet.

PROGRAM: FRENCH TUTOR

If you want to impress your friends by ordering a meal in French, try out this program. By using this program on your Commodore 64, you can become fluent with the phrases for some popular items of French cuisine. Regardless of whether this impresses your friends or not, they will probably be impressed because you learned your words with a computer! See Listing 5-4.

Listing 5-4. French Tutor

```
50 RESTORE  
100 PRINTCHR$(147):PRINT  
110 PRINTSPC(9)"FRENCH FOOD READER"  
120 PRINT:PRINT  
130 PRINTSPC(3)"1. LESSON 1-MEAT AND POULTRY"  
140 PRINTSPC(3)"2. LESSON 2-FRUITS"  
150 PRINTSPC(3)"3. LESSON 3-VEGETABLES"  
160 PRINTSPC(3)"4. LESSON 4-FISH"  
170 PRINTSPC(3)"5. LESSON 5-VARIOUS TERMS"  
180 PRINTSPC(3)"6. QUIZ ON LESSON 1"  
190 PRINTSPC(3)"7. QUIZ ON LESSON 2"
```

```

200 PRINTSPC(3)"8. QUIZ ON LESSON 3"
210 PRINTSPC(3)"9. QUIZ ON LESSON 4"
220 PRINTSPC(3)"10. QUIZ ON LESSON 5"
230 PRINT PRINT
240 INPUT"YOUR CHOICE PLEASE";CH
250 IFCH=1THEN500
260 IFCH=2THEN600
270 IFCH=3THEN700
280 IFCH=4THEN800
290 IFCH=5THEN900
300 IFCH>10THEN240
350 REM'IF CHOICE GREATER THAN 10, GO ASK AGAIN'
400 GOTO1000
450 REM ALL QUIZES START AT 1000
500 PRINTCHR$(147):PRINT
510 FORLA=1TO21
520 READD$,E$
525 REM'READ 21 PAIRS OF DATA'
530 PRINTD$;" MEANS ";E$
535 REM'DISPLAY THIS DATA'
540 NEXTLA
550 GOTO6000
600 PRINTCHR$(147):PRINT:PRINT
605 FORCL=1TO21
610 READD$,E$
615 REM'CLEAR AWAY DATA TO GET TO WHAT THE USER WANTS'
620 NEXTCL
630 FORLB=1TO9
640 READD$,E$
650 PRINTD$;" MEANS ";E$
660 NEXTLB
670 GOTO6000
700 PRINTCHR$(147):PRINT
710 FORCL=1TO30
720 READD$,E$
730 NEXTCL
740 FORLC=1TO11
750 READD$,E$
760 PRINTD$;" MEANS ";E$
770 NEXTLC
780 GOTO6000
800 PRINTCHR$(147):PRINT
810 FORCL=1TO41
820 READD$,E$
830 NEXTCL
840 FORLD=1TO8
850 READD$,E$
860 PRINTD$;" MEANS ";E$
870 NEXTLD
880 GOTO6000
900 PRINTCHR$(147):PRINT
910 FORCL=1TO49
920 READD$,E$
930 NEXTCL
940 FORLE=1TO15
950 READD$,E$
960 PRINTD$;" MEANS ";E$
970 NEXTLE
980 GOTO6000
1000 PRINTCHR$(147)
1010 IFCH=6THENCL=0:LT=20:GOTO1530
1015 REM'DON'T CLEAR ANY DATA'
1020 IFCH=7THENCL=20:LT=8:GOTO1500
1030 IFCH=8THENCL=29:LT=10:GOTO1500
1040 IFCH=9THENCL=40:LT=7:GOTO1500
1050 IFCH=10THENCL=48:LT=14:GOTO1500

```

Listing 5-4—cont. French Tutor

```

1500 FORX=0TOCL
1510 READD$,E$
1515 REM'CLEAR CL PAIRS OF DATA AWAY'
1520 NEXTX
1530 FORY=0TOLT
1535 REM'ASK LT+1 QUESTIONS'
1540 READD$,E$
1550 PRINTD$
1555 REM'DISPLAY THE FRENCH WORD'
1560 PRINT"WHAT IS THIS WORD IN ENGLISH?"
1570 PRINT"(YOU NEED NOT USE THE ARTICLE)"
1575 INPUT A$
1580 IFA$C=E$THEN2000
1585 REM'SEE IF THE RESPONSE WAS NOT CORRECT'
1590 PRINT"CORRECT"
1600 SC=SC+1
1603 REM'ADD ONE TO THE SCORE'
1605 PRINT:PRINT
1610 NEXTY:GOTO3000
1620 REM'GO TO 3000 WHEN QUIZ OVER'
2000 PRINT"WRONG. IT IS ";E$
2010 REM'GIVE THE CORRECT ANSWER'
2050 PRINT:PRINT
2100 NEXTY
3000 PRINT"YOUR SCORE ON THIS SECTION IS";(SC/(LT+1))*100
3100 REM'GIVE THE PERCENTILE SCORE'
5000 DATA VIADE,MEAT,L'AGNEAU,LAMB,LE BIFTECK,BEEFSTEAK,LE BEOUF,BEEF
5010 DATA COTELETTE,CHOP,L'ENTRECOTE,RIB STEAK,L'ESCARROT,SNAIL
5020 DATA FOIE,LIVER,LE JAMBON,HAM,LE LAPIN,RABBIT,LE RAOOUT,STEW
5030 DATA ROTI,ROAST,LA SAUCISSE,SAUSAGE,LE VEAU,VEAL
5040 DATA VOLAILLE,POULTRY,LE CANARD,DUCK,LE FAISAN,PHASANT
5050 DATA POULET,CHICKEN,A POINT,MEDIUM
5060 DATA BIEN CUIRE,WELL DONE,SAIGNANT,RARE
5070 DATA 'ABRICOT,APRICOT,L'AMANDE,ALMOND,LA CERISE,CHERRY
5080 DATA CITRON,LEMON,LA FRAISE,STRAWBERRY
5090 DATA FRAMBOISE,RASPBERRY,LE PAMPLEMOUSSE,GRAPEFRUIT
5100 DATA PECHE,PEACH,LE RAISIN,GRAPE
5105 DATA LEGUME,VEGETABLE
5110 DATA 'AIL,GARLIC,L'ARTICHAUT,ARTICHOKE,L'ASPERGE,ASPARAGUS
5120 DATA CHAMPIGNON,MUSHROOM,LE CHOU,CABBAGE,LES EPINARDS,SPINACH
5130 DATA HARICOTS VERTS,STRINGBEANS,LA LAITUE,LETTUCE
5140 DATA 'OIGNON,ONION,LE RADIS,RADISH
5150 DATA POISSON,FISH,LA CREVETTE,SHRIMP,LE HARENG,HERRING
5160 DATA HOMARD,LOBSTER,L'HUITRE,OYSTER,LA MORUE,COD
5170 DATA SAUMON,SALMON,LA TRUITE,TROUT
5180 DATA BOISSON,DRINK,LE LAIT,MILK,LE VIN,WINE,LE PAIN,BREAD
5190 DATA 'OEUF,EGG,LE RIZ,RICE,L'EPICE,SPICE,LE SEL,SALT
5200 DATA POIVRE,PEPPER,LA MOUTARDE,MUSTARD,LE MIEL,HONEY
5210 DATA CREPE,PANCAKE,LE FLAN,CUSTARD,LA PATISSERIE,PASTRY
5220 DATA RECETTE,RECIPE
6000 PRINT
6100 INPUT"HIT ANY KEY TO CONTINUE";T$
6200 GOTO50

```

Important Variables in French Tutor

CH = Input user's choice from menu

D\$,E\$ = Data read

CL = Loop to clear away unneeded data

A\$ = Response to quizzes

SC = Score

T\$ = Input for continuing the program

Explanation of Lines Within Program

100-220 Print choices.

250-300 Direct program to subroutines according to the choice.

500-980 Display desired information (vocabulary).

1000-1050 Direct program to the desired quiz and set necessary variables.

1500-1520 Clear away data that is not needed for the quiz.

1530-2100 Perform the desired quiz.

3000-3100 Calculate user's score.

5000-5210 Data.

Sample Run of Program

FRENCH FOOD READER

- 1. LESSON 1 — MEAT AND POULTRY**
- 2. LESSON 2 — FRUITS**
- 3. LESSON 3 — VEGETABLES**
- 4. LESSON 4 — FISH**
- 5. LESSON 5 — VARIOUS TERMS**
- 6. QUIZ ON LESSON 1**
- 7. QUIZ ON LESSON 2**
- 8. QUIZ ON LESSON 3**
- 9. QUIZ ON LESSON 4**
- 10. QUIZ ON LESSON 5**

YOUR CHOICE PLEASE? 3

Suggested Modifications and Projects

1. You probably have realized there are many, many French words to be learned. Think of other words that fit in particular categories (clothes, common nouns, simple verbs) and modify this program to accommodate them.
2. Try making use of sentences within the quizzes, so that hints may be attained from seeing the context of the sentences used. For example, you give the user of this program an English sentence with a French word in it, and ask for the definition of the French word.

PROGRAM: GEOMETRY

Mathematics is simple for the Commodore 64, but for those struggling through Geometry, it may not seem so easy. This program, however, will automatically calculate the area of two-dimensional objects and the area of three-dimensional objects. (Listing 5-5).

Listing 5-5. Geometry

```
100 PRINTCHR$(147)
110 PRINTSPC(10)"GEOMETRY"
120 PRINT:PRINT
130 PRINT"DO YOU WISH TO CALCULATE FOR A PLANE"
140 INPUT"FIGURE OR FOR A SOLID? (P OR S)";C$
150 IFC$="S"THEN700
200 PRINTCHR$(147):PRINT:PRINT
210 PRINT"1. RECTANGLE "
220 PRINT"2. TRIANGLE"
230 PRINT"3. TRAPEZOID"
240 PRINT"4. CIRCLE"
250 PRINT:PRINT
260 PRINT"ENTER THE OBJECT OF WHICH YOU WANT TO"
270 INPUT"FIND THE AREA";C2
275 PRINTCHR$(147):PRINT:PRINT
280 IFC2=1THEN400
290 IFC2=2THEN400
300 IFC2=3THEN500
310 IFC2=4THEN600
320 REM GO TO SPECIFIC SUBROUTINES'
400 PRINT"ENTER THE LENGTH OF THE FIGURE'S"
410 INPUT"BASE";B
420 INPUT"ENTER THE ALTITUDE";H
430 A=B*H
435 REM THE FORMULA FOR THE AREA OF A RECTANGLE
440 IFC2=1THEN460
450 A=A/2
455 REM AREA OF A TRIANGLE
460 PRINT"THE AREA IS";A
470 GOTO2000
500 INPUT"ENTER THE LENGTH OF BASE 1";B1
510 INPUT"ENTER THE LENGTH OF BASE 2";B2
520 INPUT"ENTER THE ALTITUDE";H
530 A=(H/2)*(B1+B2)
535 REM AREA OF A TRAPEZOID
540 PRINT"THE AREA IS";A
550 GOTO2000
600 INPUT"ENTER THE RADIUS";R
610 A=π*(R^2)
615 REM AREA OF A CIRCLE
620 PRINT"THE AREA IS";A
630 GOTO2000
700 PRINTCHR$(147):PRINT:PRINT
710 PRINT"1. RECTANGULAR SOLID"
720 PRINT"2. CYLINDER"
730 PRINT"3. CONE"
740 PRINT"4. SPHERE"
750 PRINT:PRINT
760 INPUT"YOUR CHOICE PLEASE";C3
770 IFC3=1THEN820
780 GOTO970
820 PRINTCHR$(147):PRINT:PRINT
```

```

850 INPUT"ENTER THE LENGTH";L
860 INPUT"ENTER THE WIDTH";W
870 INPUT"ENTER THE HEIGHT";H
880 V=L*W*H
885 REM VOLUME OF RECTANGULAR SOLID
890 INPUT"ENTER THE DEPTH";D
900 S=(L*W)+(W*D)+(D*W)
910 A=S*2
915 REM SURFACE AREA OF RECTANGULAR
920 GOTO1500
930 REM'00 DISPLAY THE INFORMATION'
970 PRINTCHR$(147):PRINT:PRINT
975 IFC3=4THEN990
976 REM'SKIP ASKING THE ALTITUDE FOR A SPHERE'
980 INPUT"ENTER THE ALTITUDE";H
990 INPUT"ENTER THE RADIUS";R
1000 IFC3=2THEN1030
1010 IFC3=3THEN1100
1020 IFC3=4THEN1200
1030 V=π*H*(R↑2)
1040 A=(2*π)*R*(H+R)
1045 REM FORMULAS FOR A CYLINDER
1050 GOTO1500
1100 INPUT"ENTER THE SLANT HEIGHT";SL
1105 REM THIS IS NEEDED FOR A CONE
1110 A=π*R*(R+SL)
1120 V=(π/3)*H*(R↑2)
1125 REM FORMULAS FOR A CONE
1130 GOTO1500
1200 A=π*4*(R↑2)
1210 V=(π/6)*R*(R↑3)
1220 REM FORMULAS FOR A SPHERE
1500 PRINT:PRINT
1510 PRINT"THE SURFACE AREA IS";A
1520 PRINT"THE VOLUME IS";V
2000 PRINT:PRINT
2050 INPUT"HIT ANY KEY TO CONTINUE";T$
2100 GOTO100

```

Important Variables in Program

- C\$ = Choice of plane figure or solid
- C2 = Plane object to be chosen
- B = Length of base
- H = Altitude
- A = Area
- V = Volume
- B1,B2 = Two bases for trapezoids
- R = Radius input
- C3 = Choice of solid
- W = Width
- D = Depth
- SL = Slant height for cone
- T\$ = 'Press any key to continue' string

Explanation of Important Line Numbers in Program

- 100-150 Find out figure the user wants to work with.
- 200-270 Print menu for plane figures.
- 280-310 Direct program to appropriate subroutine.
- 400-620 Calculations for plane figures.
- 700-750 Menu for solids.
- 820-915 Calculations for a rectangular solid.
- 970-990 Ask for information about solids.
- 1000-1020 Go to specific subroutine.
- 1030-1210 Make necessary calculations for solids.
- 1510-1520 Display the data that the user wanted.

Sample Run of Program

- 1. RECTANGLE
 - 2. TRIANGLE
 - 3. TRAPEZOID
 - 4. CIRCLE
- ENTER THE OBJECT OF WHICH YOU WANT TO
FIND THE AREA? 2

Suggested Modifications and Projects

- 1. Utilize the sprites to draw high-resolution shapes of the objects in this program.
- 2. Set up a quiz whereby the user must input formulas for different types of shapes. This will teach the user the methods used to attain the areas and volumes of different figures.

As you can see from these programs, computers can make learning less difficult and more fun. There is a lot to learn, though, both about computers and education, but hopefully the two can grow up together.

Chapter 6

The Computer Tutor

Teaching basic education with a computer is an excellent concept in itself, but what other things can a computer teach people besides arithmetic, spelling, and the other subjects commonly taught by the home computer? How about using a computer to teach people *about* computers?

If this idea seems strange to you, try to think of anyone or anything more capable of instructing someone about computers than a computer. Obviously, the machine is very effective at educating other people about itself, and this very application is covered in this chapter. The Commodore 64 will be used as a computer tutor, so to speak, to enhance anyone's knowledge about the machine itself.

PROGRAM: COMPUTER INTRODUCTION

The purpose of this program, obviously enough, is to introduce a person to the power of the computer. More importantly, this program takes advantage of the Commodore 64's features to show the features built into that particular computer. The program is basically a demonstration and offers six different options from which the user may choose. Once one of these options has been

selected, the computer begins the routine, demonstrating things such as sounds, graphics, and the BASIC language (Listing 6-1).

Listing 6-1. Computer Introduction

```

5 PRINT"J"
10 REM INTRODUCTION TO COMPUTERS
20 REM BY TIM KNIGHT
30 PRINTTAB(3)"INTRODUCTION TO THE COMMODORE 64"
35 PRINT:FOR I = 1 TO 40:PRINTCHR$(209);:NEXT:PRINT
40 PRINT"  THE COMMODORE 64 IS ONE OF THE MOST"
50 PRINT"POWERFUL COMPUTERS EVER MADE FOR THE"
60 PRINT"HOME COMPUTER USER. THIS PROGRAM WILL"
70 PRINT"DEMONSTRATE JUST A FEW OF ITS "
80 PRINT"CAPABILITIES AS A MACHINE FOR LEARNING,"
90 PRINT"PLAYING GAMES, AND EXPLORING COMPUTERS."
95 PRINT:FOR I = 1 TO 40:PRINTCHR$(210);:NEXT
100 PRINT"  TO GET THE MENU FOR THE COMPUTER"
110 PRINT"DEMONSTRATIONS YOU CAN SEE, JUST PRESS"
120 PRINT"THE LETTER 'C' TO CONTINUE."
125 FOR I = 1 TO 40:PRINTCHR$(210);:NEXT:PRINT
130 GET A$:IF A$<>"C" THEN 130
140 PRINT"J"
150 PRINT"INTRODUCTION TO THE COMMODORE 64 - DEMOS";
160 PRINT:FOR I =1 TO 40:PRINT CHR$(185);:NEXT:PRINT
170 PRINT"(1)  COLORS ON THE COMMODORE"
180 PRINT"(2)  COMMODORE CHARACTERS"
190 PRINT"(3)  SUPER SPRITE!"
200 PRINT"(4)  MUSIC MAKER"
210 PRINT"(5)  SOUNDS OF THE SIXTY-FOUR"
220 PRINT"(6)  A SIMPLE AND BASIC GAME"
230 PRINT"(7)  END PROGRAM"
240 PRINT:INPUT"PLEASE ENTER YOUR CHOICE";CH
250 IF CH<1 OR CH>7 THEN GOTO 240
260 ON CH GOSUB 1000,2000,3000,4000,5000,6000,7000
270 GOTO 140
1000 REM COLORS ON THE COMMODORE
1010 PRINT"J"
1020 PRINT"  ONE OF THE COMMODORE'S MOST POWERFUL"
1030 PRINT"FEATURES IS ITS WIDE VARIETY OF COLORS,"
1040 PRINT"YOU MAY CHANGE THE COLOR OF THE BORDER,"
1050 PRINT"(THE AREA AROUND THE TEXT YOU SEE HERE)"
1060 PRINT"THE BACKGROUND, AND EVEN THE TEXT"
1070 PRINT"ITSELF! TO SEE SOME OF THE NICE COLORS"
1080 PRINT"AVAILABLE ON THE COMMODORE, JUST WATCH"
1090 PRINT"THIS.....":FOR K = 1 TO 1000:NEXT
1100 S=53248: REM VIC CHIP REGISTER
1110 FOR I = 1 TO 20
1120 POKE S+32,INT(RND(1)*15): REM BORDER
1130 POKE S+33,INT(RND(1)*15): REM BACKGROUND
1131 FOR Z = 1 TO 500:NEXT
1135 NEXT
1140 GOSUB 10000
1150 POKE S+32,254:POKE S+33,246
1999 RETURN
2000 REM COMMODORE CHARACTERS
2010 PRINT"J"
2020 PRINTTAB(10)"COMMODORE CHARACTERS":PRINT
2030 PRINT"  THE COMMODORE 64 IS CAPABLE OF "
2040 PRINT"PRODUCING A VARIETY OF CHARACTERS ON"
2050 PRINT"THE VIDEO SCREEN. SOME OF THE"
2060 PRINT"CHARACTERS AVAILABLE ARE:":PRINT
2070 PRINT"SYMBOLS AND NUMBERS:"
2080 FOR I = 33 TO 64:PRINTCHR$(I);" ";:NEXT:PRINT
2090 PRINT:PRINT"THE UPPER-CASE ALPHABET:"

```

```

2100 FOR I = 65 TO 90:PRINTCHR$(I);" ";:NEXT:PRINT
2110 PRINT:PRINT"GRAPHICS CHARACTERS:"
2120 FOR I = 96 TO 127:PRINTCHR$(I);" ";:NEXT:FOR I = 161 TO 191:PRINT
CHR$(I);" ";:NEXT
2125 PRINT:PRINT
2130 GOSUB 10000
2999 RETURN
3000 REM SUPER SPRITE!
3001 RESTORE
3010 PRINT"☺"
3020 PRINT"    THE MOST POWERFUL GRAPHICS "
3030 PRINT"CAPABILITY ON THE SIXTY-FOUR IS THE"
3040 PRINT"POWER TO CREATE AND DISPLAY SPRITES,"
3050 PRINT"HIGH RESOLUTION CHARACTERS WHICH MAY"
3060 PRINT"BE MOVED TO AND AROUND ANY PART OF THE"
3070 PRINT"VIDEO SCREEN. THE COMPUTER CAN SET"
3080 PRINT"UP SINGLE-COLOR OR MULTI-COLOR SPRITES,"
3090 PRINT"BOTH HAVING HIGH RESOLUTION. THE "
3100 PRINT"COMMODORE CAN ALSO MANIPULATE UP TO"
3110 PRINT"EIGHT OF THEM, MOVING THEM AROUND THE"
3120 PRINT"SCREEN, DETECTING WHEN THEY COLLIDE "
3130 PRINT"WITH ONE ANOTHER, AND EVEN DETECTING"
3140 PRINT"WHEN ONE COLLIDES WITH THE BORDER OF"
3150 PRINT"THE SCREEN. TO SEE A SIMPLE SPRITE"
3160 PRINT"MOVING ACROSS THE SCREEN, JUST PRESS "
3170 PRINT"THE 'S' KEY FOR 'SPRITE'."
3180 GET A$:IF A$<>"S" THEN 3180
3190 PRINT"☺"
3200 V=13*4096: REM VIC REGISTER
3210 POKE V+21,4: REM SPRITE TWO ENABLED
3220 POKE V+23,4: REM EXPAND IN V-DIRECTION
3240 POKE 2042,13: REM SPRITE TWO POINTER
3250 FOR N= 0 TO 62:READ Q:POKE 830+N,Q:NEXT
3255 FOR TT=1 TO 3
3260 FOR X= 0 TO 200
3270 POKE V+23,4:POKE V+29,4
3280 POKE V+4,X:POKE V+5,X
3290 NEXT
3330 NEXT
3380 DATA 0,0,36,60,120
3390 DATA 36,32,72,36,56
3400 DATA 72,61,34,122,60
3410 DATA 32,120,0,0,0
3420 DATA 0,24,0,15,255
3430 DATA 240,63,255,252,249
3440 DATA 231,159,63,255,252
3443 DATA 7,255,224,0,24
3450 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3460 GOSUB 10000
3470 POKE V+21,0: REM DISABLE SPRITE
3999 RETURN
4000 REM MUSIC MAKER
4001 RESTORE:PRINT"☺"
4002 FOR Q = 0 TO 62:READ II:NEXT
4003 READ NN
4010 S=54272
4020 FOR L = S TO S+24:POKE L,0:NEXT
4030 POKE S+5,9:POKE S+6,0
4040 POKE S+24,15
4045 PRINT"    MUSIC ON THE COMMODORE IS EASY TO"
4046 PRINT"PLAY, AND CAN BE PRODUCED IN AN AMAZING"
4047 PRINT"NUMBER OF VARIATIONS. HERE'S A SHORT"
4048 PRINT"ONE-VOICE PIECE..":FOR T= 1 TO 2000:NEXT
4050 READ H,L,D
4060 IF H=333 THEN GOTO 4200
4070 POKE S+1,H:POKE S,L
4080 POKE S+4,33:FOR T=1 TO D:NEXT

```

Listing 6-1—cont. Computer Introduction

```

4090 POKE S+4,32:FOR T= 1 TO 40:NEXT:GOTO 4050
4100 DATA 12,143,250,16,195,500
4110 DATA 12,143,250,10,143,250
4120 DATA 9,104,250
4130 DATA 0,97,250,7,233,500
4140 DATA 14,24,250,12,143,500
4150 DATA 12,143,250,18,209,500
4160 DATA 15,210,250,12,143,250
4170 DATA 11,48,250,9,104,250,8,97,500
4180 DATA 14,24,250,12,143,500
4190 DATA 12,143,250,21,31,500
4200 DATA 16,195,250,12,143,250,10,143,250
4210 DATA 8,97,250
4220 DATA 11,48,500,14,24,250
4230 DATA 16,195,500,15,210,225
4240 DATA 14,24,225,12,143,450
4250 DATA 15,210,350,18,209,350
4260 DATA 16,195,250,15,210,250
4270 DATA 16,195,750,333,333,333
4280 FOR L = 8 TO S+24:POKE L,0:NEXT
4290 GOSUB 10000
4999 RETURN
5000 REM SOUNDS OF THE SIXTY-FOUR
5010 PRINT"□"
5020 PRINT" BY UTILIZING THE POWERFUL "
5030 PRINT"SOUND CAPABILITIES OF THE COMMODORE"
5040 PRINT"COMPUTER, YOU CAN GENERATE AN AMAZING"
5050 PRINT"VARIETY OF DIFFERENT SOUNDS. HERE IS"
5060 PRINT"JUST ONE 'SIMPLE SAMPLE' OF A SOUND"
5070 PRINT"POSSIBLE ON THE COMMODORE 64..."
5090 S=54272
5100 FOR II=1 TO 2000:NEXT: REM DELAY
5110 FOR L = 0 TO 24:POKE S+L,0:NEXT
5120 POKE S,150:POKE S+1,10
5130 POKE S+5,24
5140 POKE S+22,104
5150 POKE S+23,1
5160 POKE S+24,79
5170 FOR N=1 TO 10
5180 POKE S+4,129
5190 FOR T= 1 TO 250:NEXT:POKE S+4,120
5200 FOR T = 1 TO 30:NEXT:NEXT
5210 POKE S+24,0
5500 GOSUB 10000
5999 RETURN
6000 REM A SIMPLE AND BASIC GAME
6010 PRINT"□"
6020 PRINTTAB(7)"A SIMPLE AND 'BASIC' GAME"
6030 PRINT
6040 PRINT" HERE IS A GAME WRITTEN IN THE BASIC"
6050 PRINT"LANGUAGE TO SHOW YOU HOW THE COMPUTER"
6060 PRINT"IS GIVEN INSTRUCTIONS TO PLAY EVEN"
6070 PRINT"THE SIMPLEST GAME. THIS PARTICULAR"
6080 PRINT"GAME IS CALLED 'HIGH-LOW'. ALL YOU HAVE"
6090 PRINT"TO DO IS GUESS A NUMBER BETWEEN 1 AND"
6100 PRINT"100. THE COMPUTER WILL TELL YOU IF"
6110 PRINT"YOU ARE 'HIGH' OR 'LOW', AND YOU SHOULD"
6120 PRINT"MAKE ANOTHER GUESS ACCORDINGLY, UNTIL"
6130 PRINT"YOU REACH THE CORRECT NUMBER. AT THAT"
6140 PRINT"POINT, THE COMMODORE 64 WILL TELL YOU"
6150 PRINT"HOW MANY GUESSES YOU TOOK TO GUESS"
6160 PRINT"THE RIGHT NUMBER. AT THAT TIME, YOU"
6170 PRINT"WILL BE SHOWN THE ACTUAL PROGRAM "
6180 PRINT"WHICH WAS MADE TO PLAY THE GAME..."
6190 PRINT:GOSUB 10000
6200 PRINT"□":N=INT(RND(1)*100)+1

```

```

6210 INPUT"ENTER YOUR GUESS";G
6220 C=C+1: REM COUNTS HOW MANY GUESSES
6225 IF G<N THEN PRINT"LOW":GOTO 6210
6230 IF G>N THEN PRINT"HIGH":GOTO 6210
6240 PRINT"CONGRATULATIONS! YOU GOT THE NUMBER"
6250 PRINT"RIGHT IN ";C;" GUESSES!"
6260 PRINT:INPUT"HIT RETURN TO SEE THE PROGRAM ITSELF";A:PRINT
6270 PRINT"10 N=INT(RND(1)*100)+1"
6280 PRINT"20 INPUT'ENTER YOUR GUESS';G"
6290 PRINT"30 C=C+1: REM COUNTS HOW MANY GUESSES"
6295 PRINT"40 IF G<N THEN PRINT'LOW':GOTO 20"
6300 PRINT"50 IF G>N THEN PRINT'HIGH':GOTO 20"
6310 PRINT"60 PRINT'CONGRATULATIONS! YOU GOT THE NUMBER'"
6320 PRINT"70 PRINT'RIGHT IN ';C;' GUESSES!'"!PRINT
6330 GOSUB 10000
6999 RETURN
7000 REM END PROGRAM
7010 PRINT"J"
7020 PRINT" I HOPE THAT I SHOWED YOU SOME OF THE"
7030 PRINT"POWER THE COMMODORE HAS 'UNDER ITS"
7040 PRINT"KEYBOARD', NOW IT'S YOUR TURN TO "
7050 PRINT"EXPERIMENT WITH THE COMPUTER AND LEARN"
7060 PRINT"TO MASTER THIS MACHINE WITH YOUR"
7070 PRINT"EXPERIENCE! GOOD LUCK!"
7080 PRINT:PRINT:END
10000 REM PRESS C TO CONTINUE
10010 PRINTTAB(9)"PRESS 'C' TO CONTINUE"
10020 GET A$:IF A$<>"C" THEN 10020
10030 RETURN

```

Explanation of Lines Within Program

- 5-130 Introduce program and wait for key to be pressed.
- 140-240 Show options and ask for choice.
- 250-260 Make sure choice is within limits, then go to choice.
- 270 When through with subroutine, go back to options menu.
- 1000-1999 Demonstration of colors.
- 2000-2999 Demonstration of characters.
- 3000-3999 Create sprite and display for demonstration.
- 4000-4999 Play short song for music demonstration.
- 5000-5999 Play unusual sound for sound demonstration.
- 6000-6999 Play HIGH/LOW game and list it as a program.
- 7000-7080 Say good-bye and end program.
- 10000-10030 Subroutine to wait for 'C' key to be pressed.

Important Variables Within Program

- CH = Choice selected from options menu
- S = Beginning of VIC chip
- A\$ = Variable for "GET" statement
- H = High frequency
- L = Low frequency
- D = Duration of a note

- S = Beginning of sound chip (as used in program lines past 4000)
- N = Random number
- G = Guess made by player in HIGH/LOW game

Sample Run From a Portion of the Program

A SIMPLE AND 'BASIC' GAME

HERE IS A GAME WRITTEN IN THE BASIC LANGUAGE TO SHOW YOU HOW THE COMPUTER IS GIVEN INSTRUCTIONS TO PLAY EVEN THE SIMPLEST GAME. THIS PARTICULAR GAME IS CALLED 'HIGH/LOW'. ALL YOU HAVE TO DO IS GUESS A NUMBER BETWEEN 1 AND 100. THE COMPUTER WILL TELL YOU IF YOU ARE 'HIGH' OR 'LOW', AND YOU SHOULD MAKE ANOTHER GUESS ACCORDINGLY, UNTIL YOU REACH THE CORRECT NUMBER. AT THAT POINT, THE COMMODORE 64 WILL TELL YOU HOW MANY GUESSES YOU TOOK TO GUESS THE RIGHT NUMBER. AT THAT TIME, YOU WILL BE SHOWN THE ACTUAL PROGRAM WHICH WAS MADE TO PLAY THE GAME . . .

PRESS 'C' TO CONTINUE

ENTER YOUR GUESS? 54

HIGH

ENTER YOUR GUESS?

Suggested Changes and Projects

1. Try and expand the program so it covers even more areas of computing and the Commodore 64. Some areas might be data input and output, a demonstration of how the computer can balance a checkbook, and a sample of the multiple-voice music the Commodore can play.

PROGRAM: COMPUTER TUTOR 1

This is a program designed to make people more familiar with the most important BASIC keywords that the Commodore has, from ABS to ON. Once the computer is told the keyword that the operator would like to know more about, it will give a brief explanation, and sometimes a demonstration, of that keyword. See Listing 6-2.

Listing 6-2. Computer Tutor 1

```

10 REM COMPUTER TUTOR #1
20 REM BY TIM KNIGHT
30 REM THIS PROGRAM IS MADE TO ENHANCE
40 REM YOUR KNOWLEDGE OF THE BASIC
50 REM LANGUAGE
60 PRINT": REM CLEAR SCREEN
70 PRINTTAB(10)"COMPUTER TUTOR ONE":PRINT
80 PRINT"WHICH OF THE FOLLOWING WOULD YOU LIKE"
90 PRINT"TO KNOW MORE ABOUT:":PRINT
100 PRINT"(1) ABS      (2) ASC      (3) CHR$"
110 PRINT"(4) CONT     (5) DATA    (6) FOR/NEXT"
120 PRINT"(7) GET      (8) GOSUB    (9) GOTO"
130 PRINT"(10) IF/THEN (11) INPUT   (12) LEFT$"
140 PRINT"(13) LEN     (14) MID$    (15) ON"
150 PRINT:INPUT"PLEASE ENTER YOUR CHOICE":CH
160 IF CH<1 OR CH>15 THEN GOTO 150
170 IF CH<6 THEN ON CH GOTO 1000,2000,3000,4000,5000
180 IF CH<11 THEN ON CH-5 GOTO 6000,7000,8000,9000,10000
190 ON CH-10 GOTO 14000,11000,12000,13000,15000
1000 REM ABSOLUTE VALUE FUNCTION
1010 PRINT":
1020 PRINT TAB(6)"ABS=ABSOLUTE VALUE FUNCTION":PRINT
1030 PRINT" THE ABSOLUTE VALUE FUNCTION RETURNS"
1040 PRINT"THE ABSOLUTE VALUE OF A NUMBER. THIS"
1050 PRINT"MEANS THAT THE NUMBER YOU GIVE THE"
1060 PRINT"COMPUTER WILL RETURN TO YOU WITHOUT"
1070 PRINT"ANY SIGNS. IF YOU GIVE THE COMPUTER"
1080 PRINT"A POSITIVE NUMBER, IT WILL GIVE YOU BACK";
1090 PRINT"A POSITIVE NUMBER. IF YOU GIVE THE "
1100 PRINT"COMPUTER A NEGATIVE NUMBER, HOWEVER,"
1110 PRINT"YOU WILL GET A POSITIVE NUMBER BACK IN"
1120 PRINT"RETURN. FOR INSTANCE, THE STATEMENT"
1130 PRINT"[PRINT ABS(X)] WILL PRINT A POSITIVE"
1140 PRINT"NUMBER REGARDLESS OF WHETHER THE VALUE"
1150 PRINT"OF 'X' IS POSITIVE OR NEGATIVE"
1160 GOTO 30000
2000 REM ASCII VALUE FUNCTION
2010 PRINT":
2020 PRINTTAB(8)"ASC=ASCII VALUE FUNCTION":PRINT
2030 PRINT" TO DETERMINE THE ASCII VALUE"
2040 PRINT"OF A CHARACTER, THE ASC FUNCTION MUST"
2050 PRINT"BE USED. THE ASCII VALUE IS A NUMBER"
2060 PRINT"THAT CORRESPONDS TO THE COMPUTER'S"
2070 PRINT"CODE FOR THAT PARTICULAR CHARACTER."
2080 PRINT"EACH CHARACTER ON THE COMMODORE 64 HAS"
2090 PRINT"A NUMBER THAT CORRESPONDS TO ITSELF. IF";
2100 PRINT"YOU TYPED [PRINT ASC('A')], YOU WOULD"
2110 PRINT"GET BACK A VALUE OF 65, SINCE THE LETTER";
2120 PRINT"'A' HAS AN ASCII VALUE OF 65."
2130 GOTO 30000
3000 REM CHARACTER STRING
3005 PRINT":
3010 PRINTTAB(4)"CHR$=CHARACTER STRING FUNCTION"
3020 PRINT
3030 PRINT" THE OPPOSITE OF THE ASCII FUNCTION"
3040 PRINT"IS THE CHR$ COMMAND. BY USING THE "
3050 PRINT"KEYWORD CHR$, YOU MAY DETERMINE WHAT "
3060 PRINT"CHARACTER A CERTAIN ASCII NUMBER "
3070 PRINT"CORRESPONDS TO. FOR EXAMPLE, IF YOU "
3080 PRINT"TYPED [PRINT CHR$(65)], THE COMPUTER"
3090 PRINT"WOULD RESPOND BY PRINTING THE LETTER"
3100 PRINT"'A', SINCE THE ASCII CODE 65 IS THE "
3110 PRINT"COMMODORE 64'S EQUIVALENT FOR THE 'A'."
3120 PRINT"WHEN USING CHR$, YOU MUST USE A VALUE"
3130 PRINT"BETWEEN 0 AND 255."

```

cont. on next page

Listing 6-2—cont. Computer Tutor 1

```

3140 GOTO 30000
4000 REM CONTINUE FUNCTION
4010 PRINT":)"
4020 PRINTTAB(8)"CONT=CONTINUE PROGRAM":PRINT
4030 PRINT"  THE CONT COMMAND SIMPLY RESUMES THE"
4040 PRINT"EXECUTION OF A PROGRAM THAT HAS BEEN"
4050 PRINT"STOPPED, USUALLY BY THE 'RUN/STOP' KEY."
4060 PRINT"BY TYPING [CONT] THEN [RETURN], YOUR"
4070 PRINT"BASIC PROGRAM WILL BEGIN WHERE IT LEFT"
4080 PRINT"OFF. TO GIVE 'CONT' A TRY, HIT THE "
4090 PRINT"RUN/STOP KEY, THEN TYPE [CONT] AND"
4100 PRINT"[RETURN]. THE PROGRAM WILL KEEP GOING"
4110 PRINT"AS IF YOU HAD NEVER STOPPED IT!"
4120 GOTO 30000
5000 REM DATA FUNCTION
5010 PRINT":)"
5020 PRINTTAB(10)"DATA=DATA TO BE READ"
5030 PRINT
5040 PRINT"  THE DATA STATEMENT IS USED TO STORE"
5050 PRINT"A LIST OF CONSTANTS. THESE CONSTANTS"
5060 PRINT"MAY BE IN THE FORM OF NUMBERS OR WORDS"
5070 PRINT"OR EVEN BOTH. EACH CONSTANT MUST BE"
5080 PRINT"SEPARATED BY A COMMA (,) SO THE"
5090 PRINT"COMPUTER WILL KNOW ONE CONSTANT FROM"
5100 PRINT"ANOTHER. TO RETRIEVE THE DATA, THE "
5110 PRINT"COMMAND 'READ' SHOULD BE USED ELSEWHERE"
5120 PRINT"IN THE PROGRAM. FOR INSTANCE, THE "
5130 PRINT"FOLLOWING PROGRAM WILL READ THE NUMBER"
5140 PRINT"TEN FROM THE DATA STATEMENT, "
5150 PRINT"THEN PRINT IT ON THE SCREEN:"
5160 PRINT:PRINT"10 READ A"
5170 PRINT"20 PRINT A"
5180 PRINT"30 DATA 10
5190 GOTO 30000
6000 REM FOR...TO...NEXT
6010 PRINT":)"
6020 PRINTTAB(6)"FOR..NEXT=FOR/NEXT LOOP"
6030 PRINT
6040 PRINT"  THE FOR/NEXT LOOP IS SIMPLY A"
6050 PRINT"COUNTING DEVICE. IF YOU WANT TO COUNT"
6060 PRINT"TO THE NUMBER FIVE, YOU WOULD SAY 'ONE,"
6070 PRINT"TWO, THREE, FOUR, FIVE.' FOR A COMPUTER"
6080 PRINT"TO COUNT TO FIVE, YOU HAVE TO TELL IT"
6090 PRINT"WHAT NUMBER TO START COUNTING AT, WHAT"
6100 PRINT"NUMBER TO STOP COUNTING, AND WHAT STEP"
6110 PRINT"IT SHOULD COUNT AT. THE 'STEP' IS JUST"
6120 PRINT"HOW FAST THE COMPUTER COUNTS - IT CAN "
6130 PRINT"COULD COUNT 1,2,3,4,5, OR GO TWICE AS"
6140 PRINT"FAST BY COUNTING 1,3,5. THE FIRST "
6150 PRINT"EXAMPLE WOULD BE STEP 1, SINCE THE "
6160 PRINT"COMPUTER COUNTS IN STEPS OF ONE, WHILE"
6170 PRINT"THE SECOND EXAMPLE WOULD BE STEP 2, FOR"
6180 PRINT"THE COMPUTER COUNTS IN STEPS OF TWO,"
6190 PRINT"COUNTING EVERY OTHER NUMBER."
6200 PRINT:PRINT"  HIT ANY KEY TO CONTINUE..."
6210 GETA$:IFA$=""THEN 6210
6220 PRINT":)"
6230 PRINT"  TO TELL THE COMPUTER WHERE TO START"
6240 PRINT"COUNTING AND WHERE TO STOP COUNTING, "
6250 PRINT"MERELY SPECIFY THE NUMBER IN YOUR FOR/"
6260 PRINT"NEXT LOOP. IF YOU WANTED THE COMPUTER"
6270 PRINT"TO COUNT FROM 2 TO 8, FOR INSTANCE, YOU"
6280 PRINT"WOULD TYPE [FOR A=2 TO 8]. YOU CAN USE"
6290 PRINT"ANY VARIABLE YOU WANT ('A' IN THIS CASE)";
6300 PRINT"TO DO THE COUNTING. IF YOU WANTED THE"

```

```

6310 PRINT"COMPUTER TO COUNT IN STEPS OF TWO (2,4"
6320 PRINT"6,8) INSTEAD OF IN STEPS OF 1 (2,3,4,5,"
6330 PRINT"6,7,8), YOU WOULD USE THE STEP COMMAND,"
6340 PRINT"AND TYPE (FOR A=2 TO 8 STEP 2). YOU "
6350 PRINT"MAY USE ANY STEP YOU DESIRE, AND IF YOU"
6360 PRINT"COUNT BACKWARDS (LIKE FROM 10 TO 1) YOU"
6370 PRINT"MUST HAVE A NEGATIVE STEP (LIKE -1)."
```

```

6400 GOTO 30000
7000 REM GET FUNCTION
7010 PRINT"?"
7020 PRINTTAB(4)"GET=GET A VARIABLE FROM KEYBOARD"
7030 PRINT
7040 PRINT" THE GET STATEMENT IS MOST OFTEN USED"
7050 PRINT"TO GET A NUMBER OR LETTER AS TYPED INTO"
7060 PRINT"THE KEYBOARD. FOR INSTANCE, YOU MIGHT"
7070 PRINT"PUT INTO YOUR PROGRAM THIS LINE:"PRINT
7080 PRINT"10 GETA$:IF A$='' THEN 10":PRINT
7090 PRINT"THIS WOULD CAUSE THE COMPUTER TO WAIT"
7100 PRINT"UNTIL SOMETHING WAS TYPED INTO THE KEY-"
7110 PRINT"BOARD. IF THE LETTER 'A' WAS TYPED, THE";
7120 PRINT"STRING A$ WOULD THEN BE THE LETTER 'A'."
7130 PRINT"BECAUSE THE GET STATEMENT DOES NOT"
7140 PRINT"WAIT FOR A KEY TO BE PRESSED, THE "
7150 PRINT"COMMAND 'IF A$='' THEN 10' MUST BE USED"
7160 PRINT"TO MAKE THE COMPUTER WAIT UNTIL A$ DOES"
7170 PRINT"INDEED HAVE A VALUE FROM THE GET A$"
7180 PRINT"COMMAND."
```

```

7190 GOTO 30000
8000 REM GOSUB FUNCTION
8010 PRINT"?"
8020 PRINTTAB(6)"GOSUB=GO TO A SUBROUTINE"
8030 PRINT
8040 PRINT" THE GOSUB IS A COMMAND TO THE"
8050 PRINT"COMPUTER TO GO TO A DIFFERENT PART"
8060 PRINT"OF THE PROGRAM TO PERFORM A SPECIAL"
8070 PRINT"TASK, OR SUBROUTINE, AND THEN RETURN"
8080 PRINT"TO THE PLACE JUST AFTER THE SUBROUTINE"
8090 PRINT"WAS ORDERED. THE FOLLOWING PROGRAM"
8100 PRINT"WOULD INSTRUCT THE COMPUTER TO GO TO"
8110 PRINT"LINE 40, PRINT A NUMBER, THEN RETURN"
8120 PRINT"TO LINE 20, THE LINE FOLLOWING THE "
8130 PRINT"GOSUB COMMAND:"PRINT
8140 PRINT"10 A=10:GOSUB 40"
8150 PRINT"20 PRINT"PROGRAM FINISHED'"
8160 PRINT"30 END"
8170 PRINT"40 PRINT A"
8180 PRINT"50 RETURN: REM RETURN TELLS PROGRAM"
8190 PRINT"60 REM TO RETURN TO 'GOSUB' STATEMENT."
```

```

8200 GOTO 30000
9000 REM GOTO FUNCTION
9010 PRINT"?"
9020 PRINTTAB(4)"GOTO=GO TO ANOTHER PART OF PROGRAM"
9030 PRINT
9040 PRINT" THE GOTO STATEMENT TELLS THE COMPUTER";
9050 PRINT"TO GO TO ANOTHER PART OF A PROGRAM,"
9060 PRINT"THE GOTO STATEMENT IS FOLLOWED BY THE"
9070 PRINT"LINE NUMBER THE PROGRAM SHOULD GO TO."
9080 PRINT"THE FOLLOWING PROGRAM WOULD PRINT '64'"
9090 PRINT"OVER AND OVER, SINCE LINE 30 TELLS THE"
9100 PRINT"COMPUTER TO GO TO THE BEGINNING"
9110 PRINT"OF THE PROGRAM, LINE 10:"PRINT
9120 PRINT"10 A$='64'"
9130 PRINT"20 PRINT A$"
9140 PRINT"30 GOTO 10"
9150 GOTO 30000
10000 REM IF/THEN STATEMENTS
```

Listing 6-2—cont. Computer Tutor 1

```
10010 PRINT"␣"
10020 PRINTTAB(3)"IF/THEN=LOGICAL DECISION BY COMPUTER"
10030 PRINT
10040 PRINT"    FOR THE COMPUTER TO MAKE A DECISION"
10050 PRINT"AND IMPLEMENT A RESULT, THE IF/THEN"
10060 PRINT"STATEMENT IS USED. THE COMMAND MERELY"
10070 PRINT"TELLS THE COMPUTER TO DO SOMETHING"
10080 PRINT"('THEN') IF ('IF') SOMETHING IS TRUE."
10090 PRINT"FOR EXAMPLE, IF YOU WANTED THE COMPUTER"
10100 PRINT"TO GO TO LINE 100 IF THE VALUE OF THE "
10110 PRINT"VARIABLE 'A' WAS 10, YOU WOULD TYPE":PRINT
10120 PRINT"IF A=10 THEN GOTO 100.":PRINT
10130 PRINT"YOU CAN TYPE IN JUST ABOUT ANY CONDITION";
10140 PRINT"AND RESULT USING THE IF/THEN COMMAND."
10150 GOTO 30000
11000 REM LEFT$ FUNCTION
11010 PRINT"␣"
11020 PRINTTAB(5)"LEFT$=LEFT PART OF STRING"
11030 PRINT
11040 PRINT"    LEFT$ IS USED TO 'CHOP OFF' THE LEFT"
11050 PRINT"PART OF A STRING FOR WHATEVER REASON."
11060 PRINT"THE FORMAT FOR LEFT$ IS:"
11070 PRINT"LEFT$(STRING,NUMBER)'. YOU TELL "
11080 PRINT"THE COMPUTER THE STRING AND THE "
11090 PRINT"NUMBER OF CHARACTERS TO 'CHOP OFF' THE"
11100 PRINT"LEFT SIDE. IF YOU WANTED TO FIND OUT"
11110 PRINT"THE FIRST LETTER OF A PERSON'S NAME,"
11120 PRINT"FOR EXAMPLE, YOU NEED ONLY TELL THE "
11130 PRINT"COMPUTER, 'PRINT LEFT$(NAME$,1)' IF "
11140 PRINT"NAME$ WERE THE STRING FOR THAT PERSON'S"
11150 PRINT"NAME."
11160 GOTO 30000
12000 REM LEN FUNCTION
12010 PRINT"␣"
12020 PRINTTAB(8)"LEN=LENGTH OF STRING"
12030 PRINT
12040 PRINT"    THE LEN FUNCTION IS USED TO FIND"
12050 PRINT"THE LENGTH OF A STRING. THE STRING"
12060 PRINT"'CLOCKWORK', FOR INSTANCE, WOULD HAVE"
12070 PRINT"A VALUE OF NINE IF YOU TYPED 'PRINT"
12080 PRINT"LEN ('CLOCKWORK'), SINCE THE STRING HAS"
12090 PRINT"NINE CHARACTERS IN IT. IF A STRING"
12100 PRINT"CALLED 'ACD$' HAD TEN CHARACTERS IN "
12110 PRINT"IT, AND YOU TYPED L=LEN(ACD$), THE "
12120 PRINT"VALUE OF L WOULD BE 10."
12130 GOTO 30000
13000 REM MIDDLE OF STRING FUNCTION
13010 PRINT"␣"
13020 PRINTTAB(5)"MID$=MIDDLE OF STRING FUNCTION"
13025 PRINT
13030 PRINT"    THE MID$ FUNCTION FIND A PIECE OF A"
13040 PRINT"STRING WITHIN A STRING. THE FORMAT"
13050 PRINT"FOR MID$ IS 'MID$(STRING, #1, #2)'."
13060 PRINT"STRING SPECIFIES THE STRING, NUMBER"
13070 PRINT"ONE TELLS THE COMPUTER AT WHAT POINT"
13080 PRINT"IN THE STRING YOU WANT TO EXTRACT A"
13090 PRINT"PIECE OF THE STRING, AND NUMBER TWO"
13100 PRINT"TELLS THE COMPUTER HOW LONG YOU WANT "
13110 PRINT"THAT PIECE TO BE. SUPPOSE A STRING"
13120 PRINT"EXISTED CALLED T$ WHICH WAS 'QUENTIN"
13130 PRINT"COOK DELARGE'. TYPING THE STATEMENT"
13140 PRINT"[PRINT MID$(T$,8,4)] WOULD TELL THE"
13150 PRINT"COMPUTER TO PRINT 'COOK', SINCE "
13160 PRINT"'COOK' IS THE FOUR-CHARACTER LONG"
13170 PRINT"PIECE OF THE STRING BEGINNING AT"
```

```

13190 PRINT"LOCATION 8 OF T$."
13200 GOTO 30000
14000 REM INPUT STATEMENT
14010 PRINT"?"
14020 PRINTTAB(8)"INPUT=INPUT VARIABLES"
14030 PRINT
14040 PRINT" THE INPUT STATEMENT LETS A COMPUTER"
14050 PRINT"USER ENTER DATA INTO THE COMPUTER, IN"
14060 PRINT"THE FORM OF LETTERS, NUMBERS, OR BOTH."
14070 PRINT"FOR THE COMPUTER TO ACCEPT NUMERIC "
14080 PRINT"INFORMATION, A NUMBER VARIABLE MUST"
14090 PRINT"BE USED ALONG WITH INPUT (I.E. INPUT A).";
14100 PRINT"FOR ALPHANUMERIC DATA TO BE ACCEPTED"
14110 PRINT"BY THE COMPUTER, A COMMAND LIKE [INPUT"
14120 PRINT"A$] SHOULD BE USED, SEVERAL PIECES"
14130 PRINT"OF DATA MAY BE INPUT IN A ROW, BY "
14140 PRINT"USING THE INPUT COMMAND IN THIS WAY":PRINT
14150 PRINT"10 INPUT A,B$,B,A$":PRINT
14160 PRINT"THIS COMMAND WOULD ALLOW THE INPUT OF"
14170 PRINT"A NUMBER, A STRING, A NUMBER, AND"
14180 PRINT"ANOTHER STRING."
14190 GOTO 30000
15000 REM ON STATEMENT
15010 PRINT"?"
15020 PRINTTAB(6)"ON STATEMENT=MULTIPLE GOTO/GOSUB"
15030 PRINT
15040 PRINT" THE ON STATEMENT EVALUATES THE VALUE"
15050 PRINT"OF A VARIABLE, AND GOES TO A LINE NUMBER";
15060 PRINT"AS DETERMINED FOR THAT NUMBER. FOR"
15070 PRINT"EXAMPLE:"
15080 PRINT:PRINT"10 ON A GOTO 40,50,60":PRINT
15090 PRINT"IF THE VARIABLE A WAS EQUAL TO ONE,"
15100 PRINT"THE PROGRAM WOULD GO TO LINE 40. IF "
15110 PRINT"VARIABLE A WAS EQUAL TO TWO, THE PROGRAM";
15120 PRINT"WOULD GO TO LINE 50. IF VARIABLE A WAS"
15130 PRINT"EQUAL TO 3, THE PROGRAM WOULD GO TO "
15140 PRINT"LINE 60. A 'GOSUB' MAY BE USED IN PLACE";
15150 PRINT"OF THE 'GOTO' IF YOU WANT TO GO TO A "
15160 PRINT"SUBROUTINE IN THE PROGRAM."
15170 GOTO 30000
20000 REM SEE IF ANOTHER BASIC KEYWORD SHOULD BE DEFINED
20005 PRINT"?"
20010 PRINT:PRINT:PRINT' WOULD YOU LIKE TO SEE ANOTHER BASIC "
20020 INPUT"KEYWORD DEMONSTRATED(Y/N)";ANS$
20030 IF LEFT$(ANS$,1)="Y" THEN GOTO 60
20040 PRINT:PRINT"OK! HAVE A NICE DAY!":END
30000 REM WAIT FOR LETTER "C" TO BE PRESSED
30001 PRINT
30005 PRINTSPC(9)"PRESS 'C' TO CONTINUE"
30010 GET A$:IF A$="" THEN 30010
30020 IF A$="C" THEN GOTO 20000
30030 GOTO 30010

```

Explanation of Lines Within Program

- 10-150 Show keywords and ask which one to explain.
- 160-190 Make sure choice is all right, then go to that keyword.
- 1000-1160 ABS function.
- 2000-2130 ASC function.
- 3000-3140 CHR\$ function.
- 4000-4120 CONT command.
- 5000-5190 DATA statement.

6000-6400 FOR/NEXT loop.
7000-7190 GET statement.
8000-8200 GOSUB statement.
9000-9150 GOTO statement.
10000-10150 IF/THEN statement.
11000-11160 LEFT\$ function.
12000-12130 LEN function.
13000-13200 MID\$ function.
14000-14190 INPUT statement.
15000-15170 ON statement.
20000-20040 Ask if another keyword should be demonstrated.
30000-30030 Wait for 'C' to be pressed.

Important Variables in Program

CH = Choice of keyword
A\$ = Choice of keyword

Sample Run of a Portion of the Program

ABS = ABSOLUTE VALUE FUNCTION

THE ABSOLUTE VALUE FUNCTION RETURNS THE ABSOLUTE VALUE OF A NUMBER. THIS MEANS THAT THE NUMBER YOU GIVE THE COMPUTER WILL RETURN TO YOU WITHOUT ANY SIGNS. IF YOU GIVE THE COMPUTER A POSITIVE NUMBER, IT WILL GIVE YOU BACK A POSITIVE NUMBER. IF YOU GIVE THE COMPUTER A NEGATIVE NUMBER, HOWEVER, YOU WILL GET A POSITIVE NUMBER BACK IN RETURN. FOR INSTANCE, THE STATEMENT [PRINT ABS(X)] WILL PRINT A POSITIVE NUMBER REGARDLESS OF WHETHER THE VALUE OF 'X' IS POSITIVE OR NEGATIVE. PRESS 'C' TO CONTINUE

Suggested Changes and Projects

1. Try modifying the program so it allows a user to actually try out the BASIC keyword as it might be used in a program.

PROGRAM: COMPUTER TUTOR 2

Computer Tutor 2 has the same purpose as Tutor 1 except that it covers the BASIC keywords from PEEK to WAIT. See Listing 6-3.

Listing 6-3. Computer Tutor 2

```

10 REM COMPUTER TUTOR #2
20 REM BY TIM KNIGHT
30 REM THIS PROGRAM IS MADE TO ENHANCE
40 REM YOUR KNOWLEDGE OF THE BASIC
50 REM LANGUAGE
60 PRINT": REM CLEAR SCREEN
70 PRINTTAB(10)"COMPUTER TUTOR TWO":PRINT
80 PRINT"WHICH OF THE FOLLOWING WOULD YOU LIKE"
90 PRINT"TO KNOW MORE ABOUT:":PRINT
100 PRINT"(1) PEEK (2) POKE (3) PRINT"
110 PRINT"(4) READ (5) REM (6) RIGHT$"
120 PRINT"(7) RND (8) STR$ (9) TAB"
130 PRINT"(10) TIME$ (11) VAL (12) WAIT"
150 PRINT:INPUT"PLEASE ENTER YOUR CHOICE";CH
160 IF CH<1 OR CH>12 THEN GOTO 150
170 IF CH<6 THEN ON CH GOTO 1000,2000,3000,4000,5000
180 ON CH-5 GOTO 6000,7000,8000,9000,10000,11000,12000
1000 REM PEEK FUNCTION
1010 PRINT":
1020 PRINTTAB(6)"PEEK=PEEK AT A BYTE OF MEMORY"
1030 PRINT
1040 PRINT" BY USING THE PEEK COMMAND, YOU CAN"
1050 PRINT"FIND OUT THE CONTENTS OF ANY BYTE"
1060 PRINT"(PIECE OF MEMORY) WITHIN YOUR COMPUTER."
1070 PRINT"THERE ARE 65,536 BYTES IN THE COMMODORE"
1080 PRINT"64, AND YOU MAY FIND OUT THE CONTENTS"
1090 PRINT"OF ANY ONE OF THEM BY TYPING (PRINT"
1100 PRINT"PEEK(X)], X BEING THE BYTE NUMBER YOU"
1110 PRINT"ARE CURIOUS ABOUT. IF YOU WANT TO KNOW"
1120 PRINT"THE CONTENTS OF BYTE 3,420, FOR INSTANCE";
1130 PRINT"YOU ONLY HAVE TO TYPE:":PRINT
1140 PRINT"PRINT PEEK(3420)"
1150 GOTO 30000
2000 REM POKE FUNCTION
2010 PRINT":
2020 PRINTTAB(4)"POKE=PUT VALUE INTO MEMORY LOCATION"
2030 PRINT
2040 PRINT" IF YOU WANT TO PUT A CERTAIN VALUE"
2050 PRINT"INTO A MEMORY LOCATION, YOU SHOULD USE"
2060 PRINT"THE POKE STATEMENT. THE POKE COMMAND"
2070 PRINT"TELLS THE COMPUTER TO PUT A VALUE INTO"
2080 PRINT"THE BYTE YOU SPECIFY. IF YOU WANTED"
2090 PRINT"TO PUT THE VALUE 64 INTO BYTE 7000."
2100 PRINT"FOR EXAMPLE, YOU WOULD GIVE THE COMPUTER";
2110 PRINT"THE COMMAND:":PRINT
2120 PRINT"POKE 7000,64"
2130 GOTO 30000
3000 REM PRINT COMMAND
3010 PRINT":
3020 PRINTTAB(4)"PRINT=PRINT SOMETHING ON THE SCREEN"
3030 PRINT
3040 PRINT" THE PRINT COMMAND MAY BE USED TO "
3050 PRINT"EITHER PRINT VARIABLES ON THE SCREEN"
3060 PRINT"OR PRINT MESSAGES. IF YOU WANTED TO"
3070 PRINT"PRINT THE VALUE FOR THE VARIABLE 'A',"
3080 PRINT"YOU WOULD TYPE:"
3090 PRINT:PRINT"PRINT A":PRINT
3100 PRINT" IF, HOWEVER, YOU WANTED A MESSAGE"
3110 PRINT"PUT ON THE SCREEN, YOU WOULD TYPE:"
3120 PRINT:PRINT"PRINT 'THIS IS A MESSAGE'"
3130 GOTO 30000
4000 REM READ FUNCTION
4010 PRINT":
4020 PRINTTAB(10)"READ=READ DATA"

```

cont. on next page

Listing 6-3—cont. Computer Tutor 2

```
4030 PRINT
4040 PRINT" THE READ STATEMENT IS USED ALONG"
4050 PRINT"WITH THE DATA STATEMENT. READ IS USED"
4060 PRINT"TO READ DATA FROM ANOTHER PART OF THE"
4070 PRINT"PROGRAM, WHETHER THAT DATA IS IN THE"
4080 PRINT"FORM OF NUMBERS OR WORDS. THIS EXAMPLE"
4090 PRINT"WOULD READ THE VARIABLE A, READ THE"
4100 PRINT"WORD A$, THEN PRINT BOTH OF THEM:"
4110 PRINT
4120 PRINT"10 READ A,A$"
4130 PRINT"20 PRINT A,A$"
4140 PRINT"30 DATA 12,COMMODORE"
4150 GOTO 30000
5000 REM REM STATEMENT
5010 PRINT"?"
5020 PRINTTAB(9)"REM=REMARK STATEMENT"
5030 PRINT
5040 PRINT" IF YOU WISH TO MAKE A REMARK WITHIN"
5050 PRINT"A PROGRAM, EITHER FOR YOUR OWN PURPOSES"
5060 PRINT"OR FOR THE USE OF OTHERS, THE REM"
5070 PRINT"STATEMENT IS USED. WHENEVER THE"
5080 PRINT"COMPUTER ENCOUNTERS A REM STATEMENT,"
5090 PRINT"IT SIMPLY GOES ON TO THE NEXT PROGRAM"
5100 PRINT"LINE, IGNORING EVERYTHING FOLLOWING"
5110 PRINT"THE REM STATEMENT ON THAT LINE. YOU "
5120 PRINT"CAN THEREFORE PUT ANYTHING YOU WISH "
5130 PRINT"AFTER A REM STATEMENT:"
5140 PRINT:PRINT"10 REM THIS IS A REMARK"
5150 GOTO 30000
6000 REM RIGHT$ FUNCTION
6010 PRINT"?"
6020 PRINTTAB(4)"RIGHT$=TAKE RIGHT PART OF STRING"
6030 PRINT
6040 PRINT" TO REMOVE A PIECE OF A STRING FROM"
6050 PRINT"ITS RIGHT SIDE, USE THE RIGHT$ STATE-"
6060 PRINT"MENT. THE FORMAT FOR RIGHT$ IS:"
6070 PRINT:PRINT"RIGHT$(STRING,NUMBER)":PRINT
6080 PRINT"THE 'STRING' IS THE STRING YOU WANT TO"
6090 PRINT"PERFORM THE RIGHT$ STATEMENT ON, WHILE"
6100 PRINT"THE NUMBER IS THE NUMBER OF CHARACTERS"
6110 PRINT"YOU WANT TO TAKE OFF THAT STRING. THE"
6120 PRINT"RIGHT$ STATEMENT DOES NOT REMOVE ANY"
6130 PRINT"PART OF THE ORIGINAL STRING FROM ITSELF"
6140 PRINT"BUT RATHER ALLOWS YOU TO 'COPY' A "
6150 PRINT"PIECE OF THE ORIGINAL STRING INTO "
6160 PRINT"ANOTHER STRING OR ONTO THE SCREEN."
6170 GOTO 30000
7000 REM RANDOM FUNCTION
7010 PRINT"?"
7020 PRINTTAB(6)"RND=GENERATE A RANDOM NUMBER"
7030 PRINT
7040 PRINT" RND IS A COMMAND THAT GENERATES"
7050 PRINT"A RANDOM NUMBER BETWEEN 0 AND 1. A"
7060 PRINT"RANDOM NUMBER MIGHT BE .6483 OR .5392"
7070 PRINT"OR ANY OTHER NUMBER BETWEEN 0 AND 1."
7080 PRINT" IF YOU WANT THE RANDOM NUMBER TO"
7090 PRINT"BE A WHOLE NUMBER, YOU MUST MULTIPLY"
7100 PRINT"THE RANDOM NUMBER BY THE 'UPPER RANGE'"
7110 PRINT"OF THE RANDOM NUMBER YOU WANT, ADD 1"
7120 PRINT"TO IT, THEN REMOVE ANY DECIMALS BY "
7130 PRINT"USING THE 'INT' FUNCTION. FOR INSTANCE,;"
7140 PRINT"IF YOU WANTED A RANDOM NUMBER BETWEEN"
7150 PRINT"1 AND 50, YOU WOULD TELL THE COMPUTER:"
```

```

7160 PRINT:PRINT"A= INT(RND(0)*50)+1":PRINT
7170 PRINT"THE NUMBER WITHIN THE PARENTHESES"
7180 PRINT"OF THE RND FUNCTION [RND(0)] IS NOT"
7190 PRINT"IMPORTANT, SO JUST USE ZERO."
7200 GOTO 30000
8000 REM STR$ FUNCTION
8010 PRINT"J"
8020 PRINTTAB(2)"STR$=STRING REPRESENTATION OF NUMBER"
8030 PRINT
8040 PRINT"    IF YOU HAVE A NUMBER AND WOULD LIKE"
8050 PRINT"TO MAKE A STRING OUT OF IT, YOU MUST"
8060 PRINT"USE THE STR$ COMMAND. ASSUME THAT A "
8070 PRINT"VARIABLE CALLED 'T' HAD THE VALUE 67"
8080 PRINT"AND YOU WANTED TO ASSIGN A STRING CALLED";
8090 PRINT"T$ THE VALUE OF 'T', YOU WOULD HAVE"
8100 PRINT"TO TELL THE COMPUTER:"
8110 PRINT:PRINT"T$=STR$(T)":PRINT
8120 PRINT"    ONCE THIS HAD BEEN DONE, T WOULD "
8130 PRINT"STILL EQUAL 67, BUT T$ WOULD EQUAL THE"
8140 PRINT"STRING '67'."
8150 GOTO 30000
9000 REM TAB STATEMENT
9010 PRINT"J"
9020 PRINTTAB(6)"TAB=MOVE CURSOR TO THE RIGHT"
9030 PRINT
9040 PRINT"    JUST LIKE ON A TYPEWRITER, THE TAB"
9050 PRINT"FUNCTION MOVES WHATEVER IS TO BE PRINTED";
9060 PRINT"A CERTAIN NUMBER OF SPACES TO THE RIGHT,;"
9070 PRINT"THIS IS GOOD FOR MAKING SCREENS APPEAR"
9080 PRINT"NEAT. THE FORMAT FOR TAB IS:"
9090 PRINT:PRINT"PRINT TAB(X)":PRINT
9100 PRINT"WITH 'X' BEING THE NUMBER OF SPACES"
9110 PRINT"TO MOVE TO THE RIGHT, A STATEMENT"
9120 PRINT"LIKE [PRINT TAB(20) 'HI'] WOULD HAVE"
9130 PRINT"THIS RESULT:"PRINT
9140 PRINTTAB(20)"HI"
9150 GOTO 30000
10000 REM TIME$ FUNCTION
10010 PRINT"J"
10020 PRINTTAB(8)"TIME$=COMPUTER CLOCK"
10030 PRINT
10040 PRINT"    THE COMMODORE 64 HAS ITS OWN"
10050 PRINT"BUILT-IN CLOCK WHICH WORKS AS LONG"
10060 PRINT"AS THE COMPUTER IS ON. YOU CAN SET"
10070 PRINT"THE TIME BY SETTING TI$ EQUAL TO A"
10080 PRINT"STRING OF NUMBERS. IF THE TIME WAS"
10090 PRINT"EIGHT-THIRTY IN THE MORNING AND 30"
10100 PRINT"SECONDS, YOU WOULD TYPE [TI$='083030']."
10110 PRINT"TO FIND OUT THE TIME IN HOURS, MINUTES"
10120 PRINT"AND SECONDS, SIMPLY TYPE [PRINT TI$]"
10130 GOTO 30000
11000 REM VAL FUNCTION
11010 PRINT"J"
11020 PRINT"VAL=VALUE OF A STRING"
11030 PRINT
11040 PRINT"    TO FIND OUT WHAT THE VALUE OF A "
11050 PRINT"STRING IS, YOU SHOULD USE THE VAL"
11060 PRINT"STATEMENT. STRINGS ARE NOT NUMBERS"
11070 PRINT"THEMSELVES, SO THIS 'CONVERSION' "
11080 PRINT"STATEMENT MUST BE USED. SUPPOSE A "
11090 PRINT"STRING CALLED NM$ WAS '435'. TO "
11100 PRINT"FIND OUT THE VALUE OF THAT STRING, YOU"
11110 PRINT"WOULD TYPE:"
11120 PRINT:PRINT"PRINT VAL(NM$)":PRINT
11130 PRINT"    THE COMPUTER WOULD RETURN TO YOU"
11140 PRINT"A VALUE OF 435."
11150 GOTO 30000

```

cont. on next page

Listing 6-3—cont. Computer Tutor 2

```
12000 REM WAIT STATEMENT
12010 PRINT":)"
12020 PRINTTAB(6)"WAIT=WAIT FOR A BYTE TO CHANGE"
12030 PRINT
12040 PRINT" THE WAIT STATEMENT WILL CAUSE ALL"
12050 PRINT"OF THE COMPUTER'S OPERATIONS TO STOP"
12060 PRINT"UNTIL A BYTE IN MEMORY CHANGES TO"
12070 PRINT"A CERTAIN VALUE. THE FORMAT, FOR OUR"
12080 PRINT"PURPOSES, IS:"
12090 PRINT:PRINT"WAIT LOCATION,MASK 1":PRINT
12100 PRINT"LOCATION' IS THE BYTE NUMBER YOU WANT"
12110 PRINT"TO WAIT FOR, WHILE 'MASK 1' IS THE"
12120 PRINT"VALUE WHICH THAT BYTE HAS TO CHANGE"
12130 PRINT"TO FOR THE COMPUTER TO RESUME OPERATION.;"
12140 PRINT"IF, FOR EXAMPLE, YOU WANTED THE COMPUTER";
12150 PRINT"TO WAIT UNTIL BYTE 4 WAS EQUAL TO "
12160 PRINT"THE VALUE 54, YOU WOULD TYPE:":PRINT
12170 PRINT"WAIT 4,54"
12180 GOTO 30000
20000 REM SEE IF ANOTHER BASIC KEYWORD SHOULD BE DEFINED
20005 PRINT":)"
20010 PRINT"WOULD YOU LIKE TO SEE ANOTHER BASIC "
20020 INPUT"KEYWORD DEMONSTRATED(Y/N)";AN$
20030 IF LEFT$(AN$,1)="Y" THEN GOTO 60
20040 PRINT:PRINT"OK! HAVE A NICE DAY!":END
30000 REM CHECK FOR 'C' TO BE PRESSED
30010 PRINT
30020 PRINTTAB(9)"PRESS 'C' TO CONTINUE"
30030 GETA$:IF A$=""THEN 30030
30040 IF A$<"C" THEN 30030
30050 GOTO 20000
```

Explanations of Lines Within Program

- 10-150 Display keywords and ask which one is wanted.
- 160-180 Make sure choice is all right, then go to keyword.
- 1000-1150 PEEK function.
- 2000-2130 POKE command.
- 3000-3130 PRINT command.
- 4000-4150 READ function.
- 5000-5150 REM statement.
- 6000-6170 RIGHT\$ statement.
- 7000-7200 RND function.
- 8000-8150 STR\$ command.
- 9000-9150 TAB command.
- 10000-10130 TIS function.
- 11000-11150 VAL function.
- 12000-12180 WAIT command.
- 20000-20040 Ask if another keyword is desired.
- 30000-30050 Wait for 'C' to continue.

Important Variables in Program

CH = Choice of keyword

A\$ = Variable used with GET statement

Sample Run of a Portion of the Program

PEEK=PEEK AT A BYTE OF MEMORY

BY USING THE PEEK COMMAND, YOU CAN FIND OUT THE CONTENTS OF ANY BYTE (PIECE OF MEMORY) WITHIN YOUR COMPUTER. THERE ARE 65,536 BYTES IN THE COMMODORE 64, AND YOU MAY FIND OUT THE CONTENTS OF ANY ONE OF THEM BY TYPING [PRINT PEEK (X)], X BEING THE BYTE NUMBER YOU ARE CURIOUS ABOUT. IF YOU WANT TO KNOW THE CONTENTS OF BYTE 3,420, FOR INSTANCE YOU ONLY HAVE TO TYPE:

PRINT PEEK(3420)

Suggested Changes and Projects

1. Modify the program so that the user may use the keyword described as it would be used in a real program. In this way, a further understanding of the BASIC keyword may be attained.

[The text on this page is extremely faint and illegible. It appears to be a list or a series of entries, possibly a table of contents or a list of names and dates. The text is too blurry to transcribe accurately.]

Chapter 7

Programs for the Home

When you first purchased your computer, the person selling the machine to you probably noted its uses for the home, such as “checkbook balancing, figuring out mortgages, and even helping with recipes.” However, up to now, you may have had a little trouble bringing your computer to do these things. In this chapter, though, these three programs, ready to run and work with, supply you with the tools necessary to make your home more organized and efficient. There are, of course, countless applications for computers within the home, but these programs should nevertheless give you a good idea of how easily a Commodore 64 can accomplish some of these tasks.

PROGRAM: CHECKBOOK

Balancing a checkbook can be a monotonous, frustrating, and inaccurate job with pencil and paper. This program, though, will help you determine just how “balanced” your account is, and even suggest where a mistake was made if the figures don’t work out properly (Listing 7-1).

Listing 7-1. Checkbook

```
10 REM OMIT ALL DOLLAR SIGNS
500 PRINTCHR$(147):PRINT:PRINT
510 INPUT"1. ENTER YOUR BEGINNING BALANCE";BB
515 IFFA=1THEN600
517 REM 'WAS THE PLAYER ONLY CORRECTING A MISTAKE? SEE 610'
520 PRINTCHR$(147):PRINT:PRINT
530 INPUT"2. ENTER YOUR TOTAL DEPOSITS";TD
535 IFFA=1THEN600
540 PRINTCHR$(147):PRINT:PRINT
550 INPUT"3. ENTER TOTAL WITHDRAWALS";TW
555 IFFA=1THEN600
560 PRINTCHR$(147):PRINT:PRINT
570 INPUT"4. ENTER ANY SERVICE CHARGES";SC
575 IFFA=1THEN600
576 PRINTCHR$(147):PRINT:PRINT
577 INPUT"5. ENTER ANY INTEREST PAID BY BANK";IN
578 IFFA=1THEN600
580 PRINTCHR$(147):PRINT:PRINT
590 INPUT"6. ENTER THE BANK'S ENDING BALANCE";EB
593 IFEB<<(BB+TD)-(TW+SC)+IN)THEN5000
594 REM'DOES THE ENDING BALANCE NOT EQUAL THESE SUMS?'
595 IFFA=1THEN600
600 PRINTCHR$(147):PRINT:PRINT
610 INPUT"DID YOU MAKE ANY MISTAKES<Y/N>";OU$
620 IFOU$="Y"THEN700
630 PRINT"DO YOU WISH TO ADD UP ALL THE CHECKS"
635 INPUT"THAT ARE LISTED ON THE BANK STATEMENT<Y/N>";I1$
640 IF I1$="Y"THEN900
643 PRINTSPC(3)"HOW MANY CHECKS HAVE YOU WRITTEN?"
645 INPUT"(INCLUDING OUTSTANDING ONES)";I2
647 GOTO3000
649 PRINTCHR$(147):PRINT:PRINT
650 PRINT"HOW MANY OUTSTANDING CHECKS DO YOU HAVE"
655 INPUT" ";O1
660 IF O1=0THEN O1=0:GOTO3300
665 REM'IF THERE ARE NO OUTSTANDING CHECKS, SKIP THE ROUTINE AT 3200'
670 GOTO3200
700 INPUT"UNDER WHICH HEADING NUMBER";HN
710 FA=1
715 REM THIS WILL MARK THAT CORRECTIONS ARE BEING MADE
720 IFHN=1THEN500
730 IFHN=2THEN520
740 IFHN=3THEN540
750 IFHN=4THEN560
760 IFHN=5THEN576
770 IFHN=6THEN580
780 REM 720-770 'GO TO WHERE THE ERROR IS'
900 PRINTCHR$(147):PRINT:PRINT
910 INPUT"HOW MANY CHECKS ARE THERE";NC
920 FORLA=1TONC
930 INPUT"ENTER AMOUNT OF CHECK";TX
935 REM'ENTER AMOUNTS NC TIMES'
940 TY=TY+TX
945 REM TY IS THE TOTAL AMOUNT THE LIST OF CHECKS
950 NEXTLA
960 PRINT:PRINT
970 IFTY<>TWTHEN2000
975 REM'IF THIS TOTAL IS NOT EQUAL TO THE TOTAL WITHDRAWALS GO TO 2000
980 GOTO643
990 REM'GO TO THE NEXT QUESTION'
2000 PRINT"AN ERROR HAS BEEN FOUND."
2010 PRINT"THE BANK REPORTS TOTAL WITHDRAWALS OF";TW;"DOLLARS"
2020 PRINT"YOUR COMPUTATIONS FOUND A TOTAL OF";TY;"DOLLARS"
2030 PRINT:PRINT
```

```

2040 PRINT"TO CONTINUE YOU MUST SUPPORT EITHER FIGURE 1 (THE BANK) OR 2 (YOU)"
2050 INPUT " ";CA
2060 IFCA=1THENTW=TW:GOTO643
2070 IFCA=2THENTW=TY:GOTO643
2080 REM'RESET TW IF '2' IS CHOSEN'
3000 PRINTCHR$(147):PRINT:PRINT
3020 FORLB=1TOI2
3030 INPUT"ENTER AMOUNT OF CHECK";PD
3040 TP=TP+PD
3050 NEXTLB
3070 GOTO649
3200 FOR LC=1TOO1
3210 INPUT"ENTER AMOUNT OF CHECK";O2
3220 OT=OT+O2
3230 NEXTLC
3300 PRINTCHR$(147):PRINT:PRINT
3400 TW=TP-OT
3410 REM'THE NEW TW FIGURE IS TP MINUS THE AMOUNT OUTSTANDING'
3500 RB=BB-SC-TW+IN+TD
3510 REM'SET THE FIGURE FOR THE USER'S ENDING BALANCE'
3600 PRINT"THE BANK'S ENDING BALANCE IS";EB;"DOLLARS."
3610 PRINT
3620 PRINT"YOUR ENDING BALANCE IS";RB;"DOLLARS."
3700 IFRB=EBTHEN4000
3710 REM'SEE IF THESE ENDING FIGURES ARE EQUAL'
3800 PRINT:PRINTSPC(6)"SORRY. SOMEONE HAS MADE A MISTAKE SOMEWHERE.";END
3900 REM THIS IS PRINTED IF THEY ARE NOT EQUAL
4000 PRINT:PRINT"THEY BALANCE!"
4100 END
5000 PRINT"AN ERROR HAS BEEN MADE":GOTO610
5100 REM THIS IS PRINTED IF THE AMOUNT IN NO.6 DOES NOT EQUAL THE SUM -
5200 REM OF THE PREVIOUS AMOUNTS

```

Important Variables in Program

- BB = Input for beginning balance
- FA = Used to see if user was correcting a mistake
- TD = Total deposits
- TW = Total withdrawals
- SC = Service charges
- IN = Interest
- EB = Bank's ending balance
- I2 = Number of checks written
- O1 = Outstanding checks
- HN = Heading number (to correct mistakes)
- NC = Number of checks listed on bank statement
- LA = Loop to add number of listed checks
- TX = Amount of single check
- TY = Amount of listed checks
- TP = Total amount in written checks
- LC = Loop to add up outstanding checks
- O2 = Individual amount of outstanding check
- OT = Total amount outstanding
- RB = Ending balance according to the user

Explanation of Important Line Numbers in Program

- 500-595 Get necessary data from the user.
630-670 Ask user to add up certain groups of checks.
700-780 Correct mistakes that user thinks have been made.
900-950 Add up checks on bank statement.
970 Checks if this sum is equal to what the bank reported.
2000-2080 Inform user that either the bank or the user made some mistakes in addition.
3000-3050 Add up checks in the user's checkbook.
3200-3230 Add up outstanding checks.
3300-4000 Check to see if the bank's and the user's figures balance.
5000 Informs user that an error has been made when the figures in the bank statement were added.

Sample Run of Program

```
HOW MANY CHECKS ARE THERE? 3
ENTER AMOUNT OF CHECK? 345.23
ENTER AMOUNT OF CHECK? 335.69
ENTER AMOUNT OF CHECK? 183.43
```

Suggested Modifications and Projects

1. If you have a printer running on your Commodore 64, try to get the computer to print out the information to the printer instead of the monitor.
2. Modify the program so that check numbers will be accepted, thereby making checks easily identifiable by their numbers.
3. Make a subroutine to this program to calculate the interest you would earn on an interest-earning checking account over a period of months.

PROGRAM: RECIPE CONVERTER

There are many recipes, but most of them are made for a large number of people. If you are making the household's favorite meal tonight, but the recipe is for 15 people, here's a program to convert recipes made for a certain number of people into recipes for your

group of people. Of course, the program works both ways, in case you have an army coming over to your house for dinner (Listing 7-2).

Listing 7-2. Recipe Converter

```

50 PRINTCHR$(147)
75 PRINT SPC(8) "RECIPE CONVERTER"
80 PRINT:PRINT
100 INPUT"ENTER THE NAME OF THIS RECIPE";A$
110 INPUT"HOW MANY INGREDIENTS ARE THERE";NI
120 FOR X=1 TO NI
125 REM GET DATA FOR EACH INGREDIENT
130 PRINT"ENTER INGREDIENT NO.";X
140 INPUT " ";B$(X)
150 INPUT"WHAT UNIT OF MEASURE IS USED";C$(X)
160 INPUT"WHAT AMOUNT IN THIS UNIT IS USED";D(X)
170 NEXT X
180 PRINT:PRINT
190 PRINT"WHAT IS THE SUGGESTED NUMBER"
195 INPUT"OF SERVINGS";E
200 PRINT:PRINT
210 PRINT"TO WHAT NUMBER OF SERVINGS DO YOU"
220 INPUT"WISH TO CONVERT";F
230 PRINTCHR$(147)
233 PRINTSPC(9)A$
234 REM 'PRINT THE RECIPE NAME'
235 PRINT:PRINT
240 FOR Y=1 TO NI
250 G(Y)=(D(Y)*F)/E
255 REM G(Y) IS NEW AMOUNT NEEDED
260 PRINT"USE";G(Y);C$(Y);" OF ";B$(Y)
270 NEXT Y
    
```

Important Variables in Program

- X, Y = Loop variables for number of ingredients
- NI = Number of ingredients
- A\$ = Name of recipe
- E = Suggested number of servings
- F = Your number of servings
- B\$(x) = Ingredients
- C\$(x) = Units of measure used
- D(x) = Amount of ingredient used
- G(y) = New amount of ingredients needed

Explanation of Important Line Numbers Within Program

- 50-80 Title page.
- 100-170 Get name of recipe, ingredients, units, amounts.
- 180-195 Input number of suggested servings.
- 200-220 Input number you want served.
- 230-270 Print out all new information.

Sample Run of Program**TOGODRINK****USE 1 CUP OF VINEGAR****USE 2 OUNCES OF TOMATO JUICE****USE .5 TABLESPOONS OF JUCUMMA HERB****Suggested Modifications and Projects**

1. Try making a program in which you can type your recipes, and then store them onto disk for later reference.

PROGRAM: MORTGAGE MACHINE

As we have seen before, the Commodore 64 is very good at math. Figuring out mortgage payments is pretty tedious in itself, but by using this program, you may find out your monthly payments over any 12 month period for a particular loan (Listing 7-3).

Listing 7-3. Mortgage Machine

```

1 REM DIMENSION ARRAY FOR AMORTIZATION
5 DIM A(500,2)
10 REM MORTGAGE MACHINE
20 PRINT"?"
30 REM BY TIM KNIGHT
40 PRINTTAB(10)"THE MORTGAGE MACHINE"
50 PRINT
55 REM LIST CHOICES
60 PRINT"PLEASE SELECT:"
70 PRINT:PRINT"(1) AMORTIZE MORTGAGE"
80 PRINT"(2) END MORTGAGE PROGRAM"
90 PRINT:PRINT"ENTER YOUR CHOICE";
95 REM WAIT UNTIL 1 OR 2 IS PRESSED
100 GET A$:IF A$="1" THEN 1000
110 IF A$="2" THEN 2000
120 GOTO 100
999 REM SET UP INTRODUCTION
1000 PRINT"?:PRINTSPC(2)"-----AMORTIZE MORTGAGE-----"
1010 PRINT:PRINT" YOU MAY FIND OUT HOW MUCH PRINCIPAL"
1020 PRINT"AND INTEREST YOU MUST PAY ON A LOAN"
1030 PRINT"DURING ANY TWELVE MONTH PERIOD BY USING"
1040 PRINT"THIS PROGRAM. SIMPLY ENTER THE VALUES"
1050 PRINT"REQUESTED, AND THE COMMODORE 64 WILL DO"
1060 PRINT"THE REST. ENTER TOTAL AMOUNT OF"
1065 REM INPUT TOTAL AMOUNT OF LOAN
1070 INPUT"LOAN -->";A
1075 REM INPUT PERCENTAGE RATE
1080 PRINT:INPUT"ENTER PERCENT INTEREST YOU MUST PAY";I
1085 REM MAKE SURE INTEREST IS POSITIVE
1090 IF I<1 THEN 1080
1100 R=I/1200
1105 REM FIND LENGTH OF LOAN
1110 PRINT"HOW LONG DOES THIS LOAN LAST?":INPUT"(IN YEARS)";Y

```

```

1115 REM CONVERT YEARS INTO MONTHS
1120 N=Y*12
1130 GOSUB 10000
1135 REM DETERMINE MONTHLY PAYMENT 1165 REM FOR/NEXT LOOP TO DETERMINE PAYM
1140 PF=INT((PF+5.000001E-03)*100)/100
1150 PRINT"PAYMENTS EACH MONTH=";PF
1160 A(0,1)=A
1165 REM FIND OUT PAYMENTS FOR EVERY MONTH (IN INTEREST) AND BALANCE
1170 FOR I = 1 TO N
1175 REM INTEREST PAYMENT
1180 A(I,2)=INT((A(I-1,1)*R+5.000001E-03)*100)/100
1185 REM BALANCE
1190 A(I,1)=A(I-1,1)-PF+A(I,2)
1200 NEXT
1205 REM INPUT BEGINNING MONTH
1210 PRINT"NOW ENTER THE BEGINNING PAYMENT NUMBER"
1220 INPUT"ANY 12-MONTH PERIOD";PD
1230 PRINT"Q":PRINT"PAYMENT----PRINCIPAL--INTEREST--BALANCE"
1235 REM SET TOTAL INTEREST TO ZERO
1240 TN=0
1245 REM FOR/NEXT LOOP FOR 12 PERIODS
1250 FOR I = PD TO PD+11
1260 IF I>N THEN I=PD+11:GOTO 1290
1265 REM SUM TOTAL INTEREST
1270 TN=TN+A(I,2)
1275 REM DISPLAY DATA
1280 PRINT I,"$"INT(PF-A(I,2)),"$"INT(A(I,2)),"$"INT(A(I,1))
1290 NEXT
1295 REM SHOW TOTAL INTEREST FOR THAT YEARLY PERIOD
1300 PRINT:PRINT"INTEREST FOR 12 PERIODS=";TN
1310 PRINT:PRINT"PRESS 'C' TO CONTINUE"
1320 GET A$:IF A$<"C" THEN 1320
1325 REM RUN PROGRAM AGAIN
1330 RUN
2000 PRINT"Q":PRINT"THANKS FOR USING THE MORTGAGE MACHINE!"
2005 REM STOP PROGRAM
2010 END
9999 REM CALCULATE PRINCIPAL PAYMENT
10000 PF=A*(R/(1-(1/(1+R)^N))):RETURN

```

Important Variables in Mortgage Machine

- A\$ = Decision to amortize mortgage or end mortgage program
- A = Amount of loan
- N = Number of months
- PF = Payments each month
- A(x,y) = Array for payments in interest and principal
- TN = Total interest
- I = FOR/NEXT loop variable

Explanation of Important Lines Within Program

- 1-55 Set up array, print title.
- 60-120 Display menu and get choice.
- 999-1070 Show introduction, get loan.
- 1075-1120 Input interest, length, calculate months.
- 1130-1160 Calculate and show monthly payment.
- 1165-1200 Calculate all payments each month and balance.

1205-1290 Input any 12 month period; show information.

1295-1330 Show total interest; run program again if requested.

2000-2010 End program.

9999-10000 Calculate principal payment.

Sample Run of Portion of Program

YOU MAY FIND OUT HOW MUCH PRINCIPAL AND INTEREST YOU MUST PAY ON A LOAN DURING ANY TWELVE MONTH PERIOD BY USING THIS PROGRAM. SIMPLY ENTER THE VALUES REQUESTED, AND THE COMMODORE 64 WILL DO THE REST. ENTER TOTAL AMOUNT OF LOAN -- > ? 10000

ENTER PERCENT INTEREST YOU MUST PAY 13.9

HOW LONG DOES THIS LOAN LAST (IN YEARS)? 5

PAYMENTS EACH MONTH = 232.16

Suggested Modifications and Projects

1. See if you can create a program to compare different mortgage payments from different banks, so you can get the best deal in getting a loan.

These three programs show that there is more to a computer than just games. Try to find out how a computer could help you most in the home, make a program for that application, and make your life a little easier. Perhaps the program will be good enough to sell — making the lives of others a little easier and making your wallet a little fatter.

Chapter 8

Utility Programs

A utility is a program especially designed to make programming easier. In this chapter there are three utilities which will make it easier for you to develop sounds, create sprites, and make music. The computer does the hard work, saving you from the more difficult programming.

PROGRAM: SOUND DEVELOPER

Sound Developer allows you to instruct the Commodore to use different waveforms, frequencies, and ASDR (attack/sustain/decay/release) parameters to create your own sounds. By having such a wide variety of factors from which to choose, the number of sounds you can make is amazing . . . as are some of the sounds! See Listing 8-1.

Listing 8-1. Sound Developer

```
10 PRINT"☺"  
20 REM SOUND DEVELOPMENT PROGRAM  
30 REM BY TIM KNIGHT  
40 REM SOUND CHIP ADDRESS  
50 S=54272
```

cont. on next page

Listing 8-1 — cont. Sound Developer

```
60 REM CLEAR SOUND CHIP
70 FOR L = 5 TO S+24:POKE L,0:NEXT
75 PRINT"J":PRINTTAB(10)"SOUND MAKER":PRINT
80 REM SET SOUND VOLUME
90 INPUT"WHAT SOUND VOLUME DO YOU WANT(1-15)";V
100 IF V<0 OR V>15 THENPRINT:PRINT"MUST BE BETWEEN 0 AND 15! TRY AGAIN":PRINT:GOTO 90
110 POKE S+24,V
115 REM DETERMINE WAVEFORM DESIRED
120 PRINT"J": REM CLEAR SCREEN AGAIN
130 PRINT"ENTER WAVEFORM DESIRED:";PRINT
140 PRINT"(1) SAWTOOTH WAVEFORM"
150 PRINT"(2) TRIANGULAR WAVEFORM"
160 PRINT"(3) VARIABLE PULSE WAVEFORM"
170 PRINT"(4) WHITE NOISE WAVEFORM"
180 PRINT:INPUT "ENTER CHOICE";WF
190 IF WF<1 OR WF>4 THENPRINT:PRINT"OUT OF RANGE! TRY AGAIN!":PRINT:GOTO 180
200 PRINT"J"
210 ON WF GOTO 220, 230, 240, 250
220 SR=33:SP=32:GOTO 260
230 SR=17:SP=16:GOTO 260
240 SR=65:SP=64:GOTO 260
250 SR=129:SP=128:GOTO 260
260 REM FOR TRIANGULAR WAVEFORM:
270 IF WF=3 THEN POKE S+3,0:POKE S+2,0
280 REM DETERMINE ATTACK, DECAY, SUSTAIN, AND RELEASE VALUES
290 INPUT"ENTER ATTACK VALUE DESIRED (0-15)";AT
300 IF AT<0 OR AT>15 THEN PRINT:PRINT"OUT OF RANGE!":PRINT:GOTO 290
310 INPUT"ENTER DECAY VALUE DESIRED (0-15)";DC
320 IF DC<0 OR DC>15 THEN PRINT:PRINT"OUT OF RANGE!":PRINT:GOTO 310
330 REM ENTER ATTACK AND DECAY INTO MEMORY
340 POKE S+5,(AT*16)+DC
350 INPUT"ENTER SUSTAIN VALUE DESIRED (0-15)";SS
360 IF SS<0 OR SS>15 THEN PRINT:PRINT"OUT OF RANGE!":PRINT:GOTO 350
370 INPUT"ENTER RELEASE VALUE DESIRED (0-15)";RL
380 IF RL<0 OR RL>15 THEN PRINT"OUT OF RANGE!":PRINT:GOTO 370
390 REM ENTER SUSTAIN AND RELEASE INTO MEMORY
400 POKE S+6,(SS*16)+RL
402 REM DETERMINE HIGH AND LOW FREQUENCY
403 INPUT"ENTER HIGH FREQUENCY(1-255)";HF
404 IF HF<0 THEN 403
405 INPUT"ENTER LOW FREQUENCY(1-255)";LF
406 IF LF<0 THEN 405
410 PRINT"J":PRINT"HOW MANY TIMES WOULD YOU LIKE THIS "
420 INPUT"SOUND TO REPEAT";NT
425 IF (NT>0) AND (NT<10) THEN 470
430 IF NT<1 THEN PRINT"SURELY YOU WANT TO HEAR IT ONE TIME!":PRINT:GOTO 410
440 IF NT>10 THEN INPUT"ARE YOU SURE????";ANS
450 IF LEFT$(ANS,1)="Y" THEN PRINT"OK!!!":GOTO 470
460 PRINT:GOTO 410
470 PRINT"ABOUT HOW MANY 1/100'S OF A SECOND"
480 INPUT"WOULD YOU LIKE THE SOUND TO LAST";NS
490 IF NS<1 THEN PRINT:PRINT"NOT POSSIBLE!":PRINT:GOTO 470
500 NS=NS*2.5
510 PRINT"J"
520 PRINT"YOUR SOUND ";PRINT
530 PRINT"WAVEFORM TYPE =";WF
540 PRINT"ATTACK LENGTH =";AT
550 PRINT"DECAY LENGTH =";DC
560 PRINT"SUSTAIN LENGTH =";SS
570 PRINT"RELEASE RATE =";RL
580 PRINT"VOLUME =";V
590 PRINT"HIGH FREQUENCY =";HF
```

```

600 PRINT"LOW FREQUENCY =":LF
610 PRINT:PRINT"NOW PLAYING....."
1000 REM ROUTINE FOR PLAYING SOUND
1005 FOR XX= 1 TO NT
1010 POKE S+1,HF:POKE S,LF
1020 POKE S+4,SR
1030 FOR T=1 TO NS:NEXT
1040 POKE S+4,SP
1050 NEXT
1199 REM CLEAR SOUND CHIP AGAIN
1200 FOR L = S TO S+24:POKE L,0:NEXT
1250 INPUT"WOULD YOU LIKE ANOTHER SOUND(Y/N)";ANS$
1260 IF LEFT$(ANS$,1)="Y" THEN GOTO 75
1270 PRINT":PRINT"SEE YOU LATER!!!":END

```

Explanation of Line Numbers Within Program

- 10-50 Clear screen and set S equal to sound chip address.
- 60-75 Clear sound chip and put title of program onto screen.
- 80-110 Routine to set volume of sound.
- 115-250 Routine to establish waveform.
- 260-270 Special routine for variable pulse waveform.
- 280-400 Set up attack, sustain, decay, and release.
- 402-406 Input high and low frequencies.
- 410-450 Input number of times sound should play.
- 460-500 Set up duration of sound.
- 510-610 Display all parameters of sound.
- 1000-1200 Play sound and clear sound chip of all values.
- 1250-1270 See if another sound desired and act accordingly.

Important Variables in Program

- S = Sound chip address for VIC chip
- V = Volume
- WF = Waveform
- SR,ST = Start and stop values for waveform
- AT = Attack
- DC = Decay
- SS = Sustain
- RL = Release
- HF = High frequency
- LF = Low frequency
- NT = Number of times sound should repeat
- NS = 1/100 of a second sound should last
- ANS\$ = Answer to question "would you like another sound"

Special Note: Many of the VIC chip's functions are utilized in this program. Pay careful attention to all of the pokes involved with the variable 'S' so you may understand just what sections of the VIC chip are being used.

A Sample Run of a Portion of the Program

YOUR SOUND;

WAVEFORM TYPE = 1

ATTACK LENGTH = 10

DECAY LENGTH = 5

SUSTAIN LENGTH = 0

RELEASE RATE = 15

VOLUME = 12

HIGH FREQUENCY = 10

LOW FREQUENCY = 106

NOW PLAYING.

WOULD YOU LIKE ANOTHER SOUND?

Suggested Modifications and Projects

1. For future reference, you might want to write down the factors for all of the interesting sounds you come up with.
2. See if you can get some graphics involved with this program, so graphics, corresponding to the sounds, are displayed on the screen.
3. Also, try to make the program ask for changes in the data the user desires. For instance, if a person wanted to keep all of the input the same except for the waveform, the program should be able to recognize this desire and act accordingly. You might want to modify it so that if the user simply presses the **RETURN** key for every factor the user does not want to change, the computer will use the value input used for the previous sound.

PROGRAM: SPRITE CODER

This program (Listing 8-2) allows a person to enter coordinates onto a grid so that a sprite may be formed. The grid is a 24 by 21 display, with coordinates such as (A,01) (C,12) and (X,21). Notice that a single digit such as 1, 5, or 7 has to have a zero in front of it, so it would be input as 01, 05, and 07. As you enter coordinates into

the computer for your sprite, blocks will appear in the positions you have specified. Once you are through creating your sprite, press the 'Z' key, and the computer will display the data required to make that sprite. You might want to use this data with the SPRINPOT program in Chapter 2.

Listing 8-2. Sprite Coder

```

5 DIM S(22,23),B(3,22)
10 REM SPRITE CODER PROGRAM
11 REM BY TIM KNIIGHT
20 PRINT"J"
25 PRINT:PRINT:PRINT:PRINT:PRINT:PRINTSPC(13)"SPRITE CODER":FOR X=1 TO 1000:NEXT
27 PRINT"J"
30 PRINT"          [ABCDEFGHIJKLMNPOQRSTUVWXYZ]"
40 FOR I = 1 TO 20:OOSUB 1000:PRINTI$;SPC(23);I:NEXT
45 PRINT"          21";SPC(23);21
50 PRINT"          [ABCDEFGHIJKLMNPOQRSTUVWXYZ]"
60 PRINTCHR$(19);:PRINT"          ":PRINTCHR$(19);:REM GO HOME!
70 GET X$:IF X$="" THEN 70
75 IF X$="Z" THEN 5000
80 IF ASC(X$)<65 OR ASC(X$)>80 THEN 70
85 PRINTX$;" ";
90 GET Y1$:IF Y1$=""THEN 90
95 IF ASC(Y1$)<48 OR ASC(Y1$)>57 THEN 90
100 GET Y2$:IF Y2$="" THEN 100
105 IF ASC(Y2$)<48 OR ASC(Y2$)>57 THEN 100
110 Y$=Y1$+Y2$:Y=VAL(Y$)
115 IF Y>21 THEN 90
120 PRINTY;
130 L=1009
140 L=L+(Y#40)+ASC(X$)
450 POKE L,209:POKE L+54272,1
500 GOTO 60
999 END
1000 REM CLEAN UP NUMBERING
1010 IF I<10 THEN 1040
1020 I$=STR$(I)
1030 I$="          "+RIGHT$(STR$(I),2):GOTO 1050
1040 I$="          "+STR$(I)
1050 RETURN
5000 REM EXECUTE SPRITE CODING
5010 FOR I = 2 TO 22
5020 FOR J=0 TO 23
5030 IF PEEK(1034+(I#40)+J)=209 THEN S(I,J)=1:GOTO 5035
5031 S(I,J)=0
5035 POKE 1034+(I#40)+J,83
5040 NEXT:NEXT
5050 FOR I = 2 TO 22
5060 FOR J = 0 TO 7
5070 IF S(I,J)=1 THEN B(1,I)=B(1,I)+2*(7-J)
5080 NEXT:NEXT
5090 FOR I = 2 TO 22
5100 FOR J=8 TO 15
5110 IF S(I,J)=1 THEN B(2,I)=B(2,I)+2*(15-J)
5115 FOR I=2 TO 22
5117 FOR J = 16 TO 23
5120 IF S(I,J)=1 THEN B(3,I)=B(3,I)+2*(23-J)
5122 PRINT"J":PRINT"HERE ARE THE DATA NUMBERS NECESSARY"
5123 PRINT"TO CREATE YOUR SPRITE:"PRINT
5130 FOR I = 2 TO 22:FOR J=1 TO 3:PRINTB(J,I),:NEXT:NEXT
5140 END

```


Suggested Modifications and Projects

1. See if you can use a joystick to move the cursor around the screen to create a sprite, rather than having to input every coordinate.
2. Try and have the sprite being created simultaneously displayed, in its true high-resolution size, off to one side of the screen while the sprite is being formed.
3. Make a routine that can erase sprite-blocks so that if a mistake is made, it can be corrected.

PROGRAM: MUSIC MACHINE

Trying to code sheet music by referring to a high frequency/low frequency chart constantly can be rather frustrating and very boring. The Music Machine allows you to enter notes of music directly, without having to worry about programming the music in the old and tedious way. To play your song, just enter 'XXX' when the computer asks you for another note. See Listing 8-3.

Listing 8-3. Music Machine

```

5 PRINT"J": REM CLEAR SCREEN
10 REM MUSIC MACHINE
20 REM BY TIM KNIGHT
30 REM DIMENSION ARRAYS
40 DIM H(36),L(36),N$(100),NT$(36),HH(100),LL(100),LN(100)
50 REM READ DATA
60 FOR I = 1 TO 36:READ H(I),L(I),NT$(I):NEXT
70 REM SET UP SOUND
80 S=54272
90 FOR L = S TO S+24:POKE L,0:NEXT
95 POKE S+3,0:POKE S+2,0
100 POKE S+5,9:POKE S+6,0
110 POKE S+24,15
120 REM INSTRUCTIONS
130 PRINTTAB(12)"MUSIC MACHINE"
140 PRINT:PRINT" TO ENTER YOUR NOTES OF MUSIC, ALONG"
150 PRINT"WITH THE OCTAVE THEY ARE IN AS WELL AS"
160 PRINT"THEIR LENGTH, USE THE FOLLOWING FORMAT:"
170 PRINT:PRINT"[NOTE][SPACE OR #][OCTAVE][LENGTH]":PRINT
180 PRINT" THE NOTE CAN BE C,D,E,F,G,A OR B; THE"
190 PRINT"SPACE AFTER THE NOTE IS NECESSARY IF"
200 PRINT"YOU ARE PLAYING A 'REGULAR' NOTE, WHILE"
210 PRINT"THE # SIGN SHOULD BE USED IN ITS PLACE"
220 PRINT"IF YOU ARE PLAYING A SHARP NOTE. THE"
230 PRINT"OCTAVE MAY BE EITHER 3, 4, OR 5, AND"
240 PRINT"THE LENGTH MAY BE 'S' FOR A SIXTEENTH"
250 PRINT"NOTE, 'E' FOR AN EIGHTH NOTE, 'Q' FOR"

```

cont. on next page

Listing 8-3—cont. Music Machine

```

260 PRINT"A QUARTER NOTE, 'H' FOR A HALF NOTE, "
270 PRINT"OR 'W' FOR A WHOLE NOTE. FOR EXAMPLE,"
280 PRINT"THE HALF NOTE 'E' IN THE MIDDLE OCTAVE:";PRINT
290 PRINTSPC(15)"E 4H"
300 PRINT:PRINT"PRESS 'C' TO BEGIN COMPOSING YOUR MUSIC"
310 GET A$:IF A$<>"C" THEN 310
320 PRINT"Q": REM BEGIN COMPOSITION
330 N=N+1 : REM INCREMENT NOTE NUMBER
340 NT=0:PRINT"ENTER NOTE NUMBER";N;:INPUT N$(N)
345 IF N$(N)="XXX" THEN GOTO 750
350 IF LEN(N$(N))>4 THEN GOSUB 20000
360 IF LEN(N$(N))<>4 GOTO 340
370 FOR I = 1 TO 36:IF NT$(I)=LEFT$(N$(N),3)THEN NT=I:I=37
380 NEXT
390 IF NT=0 THEN PRINT:PRINT"THAT NOTE DOES NOT EXIST...TRY AGAIN":
0 PRINT:GOTO 34
400 HH(N)=H(NT):LL(N)=L(NT)
410 LE$=RIGHT$(N$(N),1)
420 IF LE$="S" THEN LN(N)=125:GOTO 480
430 IF LE$="E" THEN LN(N)=250:GOTO 480
440 IF LE$="Q" THEN LN(N)=500:GOTO 480
450 IF LE$="H" THEN LN(N)=1000:GOTO 480
460 IF LE$="W" THEN LN(N)=2000:GOTO 480
470 PRINT:PRINT"INVALID NOTE LENGTH! TRY AGAIN.":PRINT:GOTO 340
480 POKE 8+1,HH(N):POKE 8,LL(N)
485 POKE S+4,33
490 FOR DR=1 TO LN(N):NEXT
500 POKE S+4,32
505 GOSUB 1000
510 GOTO 330
750 REM PLAY SONG
755 PRINT"Q":PRINT"NOW PLAYING SONG...":PRINT
760 FOR C = 1 TO N-1
765 POKE S+1,HH(C):POKE 8,LL(C)
770 POKE S+4,33
771 FOR DR=1 TO LN(C):NEXT
775 POKE S+4,32
777 FOR IT=1 TO 50:NEXT
780 NEXT
790 INPUT"DO YOU WANT TO PLAY IT AGAIN?";ANS
800 IF LEFT$(ANS,1)="Y" THEN GOTO 750
810 GOSUB 1000
820 RUN
1000 REM CLEAR SOUND CHIP
1010 FOR L = S TO S+24:POKE L,0:NEXT
1011 POKE S+3,8:POKE S+2,0
1012 POKE S+5,9:POKE S+6,0
1013 POKE S+24,15
1020 RETURN
10000 REM DATA FOR NOTES
10010 DATA 0,97,C 3
10020 DATA 8,225,C#3
10030 DATA 9,104,D 3
10040 DATA 9,247,D#3
10050 DATA 10,143,E 3
10060 DATA 11,48,F 3
10070 DATA 11,218,F#3
10080 DATA 12,143,G 3
10090 DATA 13,78,G#3
10100 DATA 14,24,A 3
10110 DATA 14,239,A#3
10120 DATA 15,210,B 3
10130 DATA 16,195,C 4
10140 DATA 17,195,C#4
10150 DATA 18,209,D 4

```

```

10160 DATA 19,239,D#4
10170 DATA 21,31,E 4
10180 DATA 22,96,F 4
10190 DATA 23,101,F#4
10200 DATA 23,30,G 4
10210 DATA 26,156,G#4
10220 DATA 28,49,A 4
10230 DATA 29,223,A#4
10240 DATA 31,165,B 4
10250 DATA 33,135,C 5
10260 DATA 35,134,C#5
10270 DATA 37,162,D 5
10280 DATA 39,223,D#5
10290 DATA 42,62,E 5
10300 DATA 44,193,F 5
10310 DATA 47,107,F#5
10320 DATA 50,60,G 5
10330 DATA 53,57,G#5
10340 DATA 56,99,A 5
10350 DATA 59,190,A#5
10360 DATA 63,75,B 5
20000 REM INVALID FORMAT
20010 PRINT:PRINT"INVALID FORMAT! TRY AGAIN."!PRINT
20020 RETURN

```

Explanation of Lines Within Program

- 5-40 Clear screen and dimension arrays.
- 50-70 Read data for all notes.
- 80-110 Clear sound chip and set up characteristics of sound.
- 120-310 Give instructions.
- 320-510 Allow input of notes and play notes as they are entered.
- 750-780 Play all notes.
- 790-820 See if song should be played again and act accordingly.
- 1000-1020 Subroutine for clearing sound chip.
- 10000-10360 Data for all notes in three octaves.
- 20000-20020 Subroutine for invalid note format.

Important Variables in Program

- S = Sound chip address
- H(36) = Array for high frequencies
- L(36) = Array for low frequencies
- NS(100) = Array for notes in song
- NT\$(36) = Array for notes in three octaves
- HH(100) = High frequencies for notes in song
- LL(100) = Low frequencies for notes in song
- LN(100) = Duration for notes in song
- LE\$ = String for length of note

A Sample Run of a Portion of the Program

```
ENTER NOTE NUMBER 1? C 3S  
ENTER NOTE NUMBER 2? E 4H  
ENTER NOTE NUMBER 3? D3S  
INVALID FORMAT! TRY AGAIN!  
ENTER NOTE NUMBER 3? D#3S
```

Suggested Modifications and Projects

1. Try to make the program display notes on a sheet of music as the music is being played. This could be an interesting test of your skill with graphics.
2. See if you can make this program into an "editor" so that certain notes within a song could be changed at will, rather than the whole song having to be re-typed into the computer.

Chapter 9

The Creation of a Program

Now that you have seen a large number of programs for your computer, perhaps you would like to know just how a program is created. That is, how does a program evolve from a simple idea to a finished, bug-free product? In this last chapter, we would like to answer this question for you.

Writing a program involves basically three steps. First, there is the idea and planning stage for a program. Secondly, a program has to be actually typed into the computer. Lastly, any problems must be removed from the program, and any modifications desired must be made. Here are some more detailed descriptions of these three important steps.

IDEA/PLANNING

Some people believe that creating an idea for a program is just about the most difficult part of programming. In some cases, this seems true. For instance, if you had to come up with a completely original game idea within half an hour, you probably would have some trouble making a game concept which had not been tried before. Still, if you have a creative mind, you can always find something new and exciting that has not yet been tried.

Naturally, ideas for programs which are not games tend to come a little easier. Simply thinking about what a computer can do for people usually leads to an idea. If you can successfully merge the power of a computer with the needs of people, you are certain to get a fine idea for a program.

Once you do have an idea, there should be some planning involved. Flowcharts and a listing of the variables you are going to use are often helpful, but you should also plan for the other aspects of your program. For instance, if you are writing a game, be sure to plan the number of points that will be made for destroying certain aliens. If, on the other hand, you are writing a program to create music, determine just how many notes the program will be able to hold, given the memory your computer has. This planning will save you a lot of trouble as you begin writing your program.

WRITING A PROGRAM

As strange as it may seem, the actual writing of a program is not the most difficult part of a program's creation. For some, it is the easiest task of all. However, when writing a program, a computer user should consider carefully the structure of a program, and be sure to REMark the program thoroughly, so the user will remember what the important lines of the program accomplish.

Sometimes it is easier to write a program in sections. That is, instead of sitting down and trying to type in the whole program from the very beginning, many people take certain routines from their program, complete them, and later combine them to make the finished product. If you were writing a program that created graphics characters, for example, you might want to write a routine to accept input from the keyboard first, then a routine to draw the graphics on the screen, then a routine to save the graphics. By breaking down a big program into several small programs, the job of creating software gets much easier.

DEBUGGING AND FINISHING

Nearly every programmer would agree that debugging a program is the most exasperating, though sometimes exhilarating, job that a person could ever hope to accomplish. The elusive bugs which find their way into nearly every program ever created are often pesky

and destructive, and will certainly eat away at your program and your sanity until you find them. Sometimes, debugging a program actually takes longer than writing it!

Still, once you have finished the job of removing those bugs, a little polishing and finishing is in order. Getting your program just the way you want it is rarely accomplished, since there always seem to be more features that you could add but you really have no desire to. Nevertheless, a programmer's finished product is always something to show off with pride.

Because you have finished this book, you probably have a pretty good idea how to program in the BASIC language. In fact, if you understand most of the things in this book, you have most likely mastered BASIC and are ready to move onto something a little more advanced. Still, before moving on, be sure that you have gained knowledge about the following three things:

1. The PEEKs and POKEs of the Commodore — These are extremely important, for just about any applications, so be sure you understand and know these addresses so you may use them with ease later.
2. Structure — If you see that your programming is a little like a patchwork quilt, try to work on your structure. Before you try anything more advanced than BASIC, assure yourself that your programs are well-structured and organized.
3. Know your computer — Understand how it may be expanded and modified, and know the basics of your computer's operation. Many programmers find themselves stalled temporarily because of minor problems such as their printer running out of ink or a disk drive not functioning properly. By knowing simple things like how to replace an ink ribbon for a printer and knowing how to clean your disk drive periodically, a lot of programming time can be saved in the future.

Once you feel you are ready to move on, there are several different ways you can graduate from the BASIC language.

One way is to purchase a compiler. A compiler is a program that will take a program you write in the BASIC language and compile it into another language, such as machine language. This means that your programs may run in BASIC at very fast speeds without your having to program them in assembly language. The reason this is considered a "graduation" from BASIC is that compilers usually

require special syntax and words so that they tend to be partly BASIC and partly assembler-like code. In this way, you are gently taught about machine language, without having to undergo the ordeal of learning the real language of your computer, the 6502 instruction set.

Another good step up from BASIC is another high-level language. A high-level language is a language that is sophisticated like BASIC, FORTRAN, or PASCAL, unlike machine language which consists of simple mnemonics (codes) very close to the computer's own binary language. If you try a high-level language other than BASIC, you will increase your exposure to the world of computers.

Finally, you might just want to dive into machine language. There are many good books on the 6502 microprocessor, and some of them are written strictly for beginners. You may get a little frustrated at first, but if you do eventually master 6502, your programs can be incredibly fast and efficient, and you can probably start marketing your creations with a large software manufacturer, if your programs are interesting and sophisticated enough.

YOU AND YOUR COMPUTER

Since you are now part of the computer revolution, take advantage of it. You certainly have the knowledge now to do with your computer what you wish, so try whatever you think has possibilities. If you think your computer can help you run your house better, give it a try. If you have a great game idea you believe can make you a substantial amount of money, go ahead and write it. Whatever you decide to do with your computer, remember that computers are made to help you, and you should make every effort to use your computer to its full potential. The Commodore 64 is a great computer at a great price, and it can make your life easier, more enjoyable, and, perhaps, even more profitable. Let your creativity be released through your machine, and keep on computing.

CASSETTE TAPE

The programs in this book are available on cassette tape. If you purchased the tape-book combination or the separate tape, use the following instructions to load the tape. Place tape in Datasette and type LOAD"" or LOAD"MENU". Press RETURN and follow the instructions on the monitor. This will load the first program which is the MENU program. Run the program and you will see numbers in parentheses following the program name. This number is the approximate location on the Datasette counter where the program starts. To load any program listed, fast forward the Datasette to the counter number shown for that program. Type LOAD "Program Name" (exactly as it appears in the MENU), press RETURN, and follow the instructions on the monitor.

HATE TO KEY-IN PROGRAM LISTINGS?

Spend \$7.95 for a ready-to-run tape of the programs in *Commodore 64 BASIC Programs* and you'll avoid spending hours tracking down a bug accidentally typed into a stalled or crashed program (it can happen to *anybody*!)

Ask for No. 22289 \$7.95

Special combination package — book and tape together!

Ask for No. 26171 \$16.95

Use the handy order form and send to the address below or call the Sams Order Desk at 800-428-SAMS or 317-298-5566 and charge it to your MasterCard or Visa account.

Ask about other Sams book/tape combinations for the Commodore 64, VIC 20 and other microcomputers too, or contact your local Sams dealer.



Howard W. Sams & Co., Inc.
4300 W. 62nd Street — P.O. Box 7092
Indianapolis, IN 46206

Catalog No.	Qty.	Price	Total	Catalog No.	Qty.	Price	Total	
Subtotal								
<input type="checkbox"/> Check <input type="checkbox"/> Money Order <input type="checkbox"/> Visa			Add local tax where applicable					
<input type="checkbox"/> MasterCard			Add Handling Charge				2.00	
Total Amount Enclosed								
Account Number _____		Expires _____						
Name (print) _____								
Signature _____								
Address _____								
City _____				State _____		Zip _____		
<small>Prices subject to change without notice • Offer good in USA only. In Canada, contact Lenbrook Electronics, Markham, Ontario L3R 1H2.</small>								
							X0462	

22171

TO THE READER

Sams Computer books cover Fundamentals — Programming — Interfacing — Technology written to meet the needs of computer engineers, professionals, scientists, technicians, students, educators, business owners, personal computerists and home hobbyists.

*Our Tradition is to meet your needs
and in so doing we invite you to tell us what
your needs and interests are by completing
the following:*

1. I need books on the following topics:

2. I have the following Sams titles:

3. My occupation is:

<input type="checkbox"/> Scientist, Engineer	<input type="checkbox"/> D P Professional
<input type="checkbox"/> Personal computerist	<input type="checkbox"/> Business owner
<input type="checkbox"/> Technician, Serviceman	<input type="checkbox"/> Computer store owner
<input type="checkbox"/> Educator	<input type="checkbox"/> Home hobbyist
<input type="checkbox"/> Student	Other _____

Name (print) _____

Address _____

City _____ State _____ Zip _____

Mail to: **Howard W. Sams & Co., Inc.**

Marketing Dept. #CBS1/80
4300 W. 62nd St., P.O. Box 7092
Indianapolis, Indiana 46206

22171

Commodore 64 BASIC Programs

- Gives the Commodore 64 user 30 ready-to-run programs.
- Shows a sample run of the program.
- Explains each program by line numbers.
- Lists important variables in programs.
- Provides suggested modifications of programs.

The wide variety of programs for experimentation and entertainment are divided into the following areas of interest:

- Educational
- Games
- Graphics
- Personal
- Sound
- Utilities

Howard W. Sams & Co., Inc.

4300 West 62nd Street, Indianapolis, Indiana 46268 U.S.A.

\$9.95/22171

ISBN: 0-672-22171-3