

TI-99/4A™ BASIC Programs

Timothy Orr Knight and Darren LaBatt



TI-99/4A™ 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 active 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 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*, *Megabucks From Your Microcomputer*, and *Graphics and Sounds on the IBM PC*. He also wrote a SAMS Combo Pack—*Commodore 64 Graphics and Sounds*. He is co-author, along with Darren LaBatt, of another Combo Pack, *Commodore 64 BASIC Programs*.



Darren LaBatt—Having become an enthusiastic computerist in less than a year, Darren's first and only computer has been the Commodore 64. He recently mastered machine language. Darren is co-author, along with Timothy Orr Knight, of the SAMS Combo Pack, *Commodore 64 BASIC Programs*.

TI-99/4A™ BASIC Programs

by
Tim Knight and Darren LaBatt

Adapted for the TI-99/4A
by
M. Troy Rondot

Howard W. Sams & Co., Inc.

4300 West 62nd St.
Indianapolis, Indiana 46268 USA

Copyright ©1984 by Timothy Orr Knight and Darren LaBatt

FIRST EDITION
SECOND PRINTING—1984

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: 0-672-22385-6
Library of Congress Catalog Card Number: 84-50803

Edited by *Douglas P. DeBrabant*

Printed in the United States of America.

TI is a trademark of Texas Instruments Incorporated.

Preface

This is a book created for the TI-99/4A user interested in learning more about how to use and program his computer. Although the BASIC programs themselves are useful and entertaining, this book was written for more than simply typing programs into a TI-99/4A and running them.

Because the greatest knowledge comes from hands-on experience, we have made the information in this book versatile enough for you to use easily as you experiment with these programs and learn more about your personal computer.

Each program has an introduction that explains what the program does and how it is used. In addition, the variables and lines of each BASIC program work and also 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 TI-99/4A 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 to your benefit.

The programs in this book were originally written for the Commodore 64. Our thanks to M. Troy Rondot for adapting them for the TI-99/4A computer.

TIMOTHY ORR KNIGHT
DARREN LABATT

A NOTE TO THE READER

The programs in this book were not written as applications software but as educational examples of what your personal computer can do. All of the programs have been tested and work on the machine configuration for which they were designed. The programs, or subroutines, are unprotected. This means that you can modify them to better understand how they work or to fit a different machine configuration.

What is a Combo Pack?

A Combo Pack, like this package, is a step beyond your average technical book. While most books give you programming examples through printed listings (which we do here), Combo Packs provide the book and the listings recorded on magnetic media, either diskette, cassette tape, or both.

Every effort has been made to be clear, concise, and informative about how these programs and routines work. If you experience any difficulty with the software operations, the solution can be found in the book or in your computer manuals.

We are rather proud of the time and effort that went into preparing the Combo Pack. If you have purchased and enjoyed it, let us know your thoughts. Your comments will be valuable in preparing future Combo Packs.

LOADING INSTRUCTIONS

The cassette accompanying this Combo Pack contains the subroutine listings and/or program listings printed in the book. To load a cassette file from this tape, perform the following steps:

1. Put the cassette into the cassette recorder.
2. Position the tape at the beginning of the subroutine or program you wish to load.
3. Type **OLD CS1**
Press <Enter>
4. Follow the directions as they appear on your video screen.

This will cause the next program on the tape to load into the computer's memory. When the program is loaded, it is ready to be used as described in the book.

The following list shows the listing names and tape counter positions for the contents of the cassette tape. These numbers are approximate and may vary from recorder to recorder. They should, however, assist you in locating the programs you are searching for.

Tape Directory

Program	Approximate Cassette Locations
Computer Introduction	0
The Art of Boxes	22
Patterns	30
Micro Sketch	39
Scrambler	46
Character Generator	52
Character Coder	61
Our First Song	72
Computer Lullaby	78
Bach Minuet	86
Sounds of Dixie	95
Piano Machine	104
Music Machine	115
Sound Developer	127
Jelly Beans	144
Rocket Race	152
Dice Games	162
No-Frills Dice	177
Simon Says	190
Blackjack	202
History	217
Places and Capitals	236
Metric Converter	254
French Food	264
Geometry	282
Computer Tutor #1	296
Computer Tutor #2	328
Checkbook	355
Recipe Converter	373
Mortgage Machine	387

Contents

CHAPTER 1

AN INTRODUCTION TO THE PROGRAMS	11
The Format — A Review of the TI-99/4A — Computer Introduction	

CHAPTER 2

GRAPHICS OF THE TI-99/4A	19
The Art of Boxes — Patterns — Micro Sketch — Scrambler — Character Generator — Character Coder	

CHAPTER 3

SOUNDS OF THE TI-99/4A	37
Our First Song — Computer Lullaby — Bach Minuet — Sounds of Dixie — Piano Machine — Music Machine — Sound Developer	

CHAPTER 4

GAMES	57
Jelly Beans — Rocket Race — Dice Games — No-Frills Dice — Simon Says — Blackjack	

CHAPTER 5

EDUCATIONAL PROGRAMS	81
History — Places and Capitals — Metric Converter — French Food — Geometry — Computer Tutor #1 — Computer Tutor #2	

CHAPTER 6

PROGRAMS FOR THE HOME	105
Checkbook — Recipe Converter — Mortgage Machine	

CHAPTER 7

THE CREATION OF A PROGRAM	117
Conceptualizing and Planning — Writing a Program — Debugging and Finishing — You and Your Computer	

Chapter 1

An Introduction to the Programs

The TI-99/4A is one of the most powerful, exciting, and economical computers yet introduced into the personal computer market. It has the power to display graphics in vivid colors, generate sounds and music of high quality, and do most of the other extremely useful things a home computer can do.

Because the TI-99/4A has been especially popular, many people are aware of its features. However, a great number who own TI-99/4As are not quite sure what to do with the computer after it is ready to run. In fact, not many people learn to unleash its power without going through a great deal of trial and error. This book of BASIC programs will help solve that problem for TI-99/4A users.

There are several ways this book may be used to benefit anyone using a TI-99/4A, especially if that person is just learning about computers. The book has three main purposes:

1. To provide TI-99/4A users with a total of 30 ready-to run programs in the BASIC language.
2. To permit users to see how the programs work so that they can achieve greater knowledge about the TI-99/4A and programming.

3. To suggest certain modifications to the programs that users can try on their own. These modifications provide opportunities for further learning since modifying an existing program is excellent experience in program development.

This book contains 30 programs, each of which is explained by an analysis section so that, by examining any program closely, you may readily determine how it works. By building, using, and modifying the programs contained in this book, nearly anyone may gain a greater knowledge of the TI-99/4A computer as well as of the BASIC language.

THE FORMAT

For ease and efficiency of use, a specific format has been set up for each of the 30 programs. First, the programs are divided into five “groups” including programs for graphics, sounds, and even programs for learning 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 is no need to follow the order of the chapters.

The format for each of the programs is:

1. An introduction to the program, explaining its purpose, uses, and what special features of the TI- 99/4A are being used in the program.
2. An analysis of how the program works, including important variables and a line-by-line analysis. More advanced programmers should pay close attention to the workings of the program, while beginners should concentrate on the program first and the technical information later.
3. A sample run of each program, so you can see how the program actually “runs” when it is completely typed into the computer. Some programs are represented by photographs so you can see what result the program produces. Other programs, involving strictly text (letters and numbers), are written out in the same way that they would appear on the computer video screen.
4. Suggested modifications: many people after running a program, want to make a few changes of their own. To help you get started, there are several suggestions for modifications to the programs that you might want to try. Changing or improving programs is not only fun, it’s educational.

A REVIEW OF THE TI-99/4A

The following is a short review designed to acquaint you with the computer itself as well as some of the terms used throughout this book.

In this book, words in braces (e.g., {RETURN}) designate keystrokes. That is, when this designation is used, it means the word in braces is a single keystroke on the keyboard (or several keys that must be pressed simultaneously to produce a desired result) instead of individual letters to be typed into the computer.

One of the most important parts of the computer is the keyboard. The keyboard is similar to that of an electric typewriter, but there are some important differences. First, the {RETURN} key is equivalent to the carriage return on the typewriter. Whenever you finish typing in a program line, you must press the {RETURN} key for the computer to put the line into its memory. The {FCTN} key stands for function and may be used for a variety of purposes, such as moving the cursor around and printing certain characters. The {FCTN} key is pressed simultaneously with another key on the keyboard to get various effects, e.g., {FCTN/?} types a question mark. The {FCTN/arrow keys} allow you to move the "cursor", or flashing box, on the computer's video monitor. The {FCTN/INS} allows you to insert letters into your current line. {FCTN/DEL} is used to delete the character at the current cursor position. {FCTN/CLEAR} is used to stop a program in action. {FCTN/QUIT} stops the program, clears the memory, and resets the computer. You must be very careful: *only press these last two keys together after you have saved your program, or it will be lost when the memory is cleared.*

The video monitor, commonly called the "screen", for the TI 99/4A is where the letters, numbers, and graphics are displayed. Since the TI-99/4A is capable of producing a wide variety of colors, a color monitor or television is almost a necessity for a TI-99/4A user.

The input/output device for the TI-99/4A is a disk drive or a cassette recorder. Both perform the same function of storing data onto magnetic media. The important difference between the two is that a disk drive saves information much more quickly and efficiently than a cassette recorder. Regardless of the media, the input/output device saves your programs for future use. It is highly recommended that you save each of your programs after you type them in or edit them, since retyping a program every time you want to use it is very time consuming.

You do not really need any other information to use this book fully. Definitions of terms like RAM, KILOBYTE, and NIBBLE have more to do with a computer salesperson's spiel than this book.

Now that we have cleared the way, let's start working to see what kind of power and excitement we can find by using this marvelous computer.

COMPUTER INTRODUCTION

The purpose of this program is to introduce you to the power of a computer. More importantly, it demonstrates the variety and power of the TI-99/4A's features. This program is basically a demonstration and offers six different options for the user to choose from. The options demonstrate colors, sounds, animation, and the BASIC language.

Listing 1-1

```
100 REM  COMPUTER INTRODUCTION
110 CALL CLEAR
120 RANDOMIZE
130 PRINT "  INTRO TO THE TI-99/4A":
140 PRINT "THE TI-99/4A IS ONE OF THE":"MOST POPULAR HO
ME COMPUTERS":
150 PRINT "JUST A FEW OF THE TI-99/4A'S":"CAPABILITIES
AS A MACHINE":"FOR LEARNING, PLAYING GAMES,"
160 PRINT "AND EXPLORING COMPUTERS":
170 CALL KEY(3,K,S)
180 IF S=0 THEN 170
190 CALL CLEAR
200 PRINT "  INTRO TO THE TI-99/4A":
210 PRINT "  1. COLORS ON THE TI-99/4A":"  2. TI-99/4A C
HARACTERS":"  3. CHARACTER ANIMATION":
220 PRINT "  4. MUSIC MAKER":"  5. SOUNDS OF THE TI-99/4
A":"  6. A SIMPLE GAME":"  7. EXIT THE PROGRAM":
230 PRINT TAB(4);"YOUR CHOICE? (1-7)":
240 CALL KEY(3,K,S)
250 IF (S=0)+(K<49)+(K>55) THEN 240
260 CALL CLEAR
270 ON K-48 GOTO 280,510,780,1040,1160,1330,1740
280 PRINT :::::"ONE OF THE TI-99/4A'S":"MOST POWERFU
L FEATURES IS":"ITS WIDE VARIETY OF COLORS":
290 PRINT "YOU MAY CHANGE THE COLOR OF":"THE BACKROUND,
THE COLOR":"OF THE CHARACTERS, OR THE"
300 PRINT "COLOR AROUND THE CHARACTERS":
310 FOR D=1 TO 1500
320 NEXT D
330 FOR L=1 TO 30
340 A=INT(15*RND)+2
350 B=INT(16*RND)+1
360 IF B=A THEN 350
370 C=INT(15*RND)+2
380 IF (C=B)+(C=A) THEN 370
390 CALL SCREEN(B)
400 FOR L2=1 TO 8
```

```

410 CALL COLOR(L2,A,C)
420 NEXT L2
430 FOR D=1 TO 20
440 NEXT D
450 NEXT L
460 CALL SCREEN(4)
470 FOR L=1 TO 8
480 CALL COLOR(L,2,1)
490 NEXT L
500 GOTO 1700
510 PRINT "THE TI-99/4A IS CAPABLE OF":"PRODUCING A VAR
    IETY OF":"CHARACTERS ON THE SCREEN":
520 PRINT TAB(5);"NORMAL CHARACTERS"::::TAB(5);"UPPER C
    ASE"::::TAB(5);"LOWER CASE"::::
530 PRINT TAB(5);"SPECIAL CHARACTERS"::::
540 L=38
550 FOR R=10 TO 18 STEP 4
560 FOR C=5 TO 27 STEP 2
570 L=L+1
580 IF L<>51 THEN 600
590 L=65
600 IF L<>77 THEN 620
610 L=97
620 CALL HCHAR(R,C,L)
630 NEXT C
640 NEXT R
650 CALL CHAR(128,"00183C7E7E3C1800")
660 CALL HCHAR(22,10,128)
670 CALL CHAR(129,"7E81A581A5BD817E")
680 CALL HCHAR(22,12,129)
690 CALL CHAR(130,"8142241818244281")
700 CALL HCHAR(22,14,130)
710 CALL CHAR(131,"183878581818FFFF")
720 CALL HCHAR(22,16,131)
730 CALL CHAR(132,"1818FF3C3C3C2466")
740 CALL HCHAR(22,18,132)
750 CALL CHAR(133,"3C7EE7C3FFFFFC3C")
760 CALL HCHAR(22,20,133)
770 GOTO 1700
780 PRINT "ANIMATION IS POSSIBLE IN":"TI-99/4A BASIC BY
    USING":"SPECIAL CHARACTERS"::::
790 CALL CHAR(128,"18183C7F3C3C2464")
800 CALL CHAR(129,"0018183CFE3C2426")
810 CALL CHAR(130,"0000000000387F24")
820 FOR L=1 TO 20
830 CALL HCHAR(12,15,128)
840 CALL HCHAR(12,15,129)
850 FOR D=1 TO 30
860 NEXT D
870 NEXT L
880 CALL HCHAR(12,15,32)
890 FOR L=3 TO 25
900 CALL HCHAR(12,L-1,32)
910 CALL HCHAR(12,L,130)
920 FOR D=1 TO 30
930 NEXT D
940 NEXT L
950 CALL HCHAR(12,25,32)

```

cont. on next page

Listing 1-1—cont.

```
960 CALL CHAR(131,"7E81A981A5BD817E")
970 CALL CHAR(132,"7E819581A5BD817E")
980 PRINT " PRESS ANY KEY TO CONTINUE":
990 CALL HCHAR(15,15,131)
1000 CALL KEY(3,K,S)
1010 CALL HCHAR(15,15,132)
1020 IF S=0 THEN 990
1030 GOTO 190
1040 PRINT "MUSIC ON THE TI-99/4A IS":"EASY TO PLAY AND
      CAN BE":"PRODUCED IN AN AMAZING"
1050 PRINT "NUMBER OF VARIATIONS"::"HERE IS A SHORT SON
      G USING":"ONE VOICE":::::"SONG NOW PLAYING....":::
      ::
1060 RESTORE 1130
1070 READ T,D
1080 IF T=9999 THEN 1700
1090 CALL SOUND(D,T,0)
1100 FOR L=1 TO 10
1110 NEXT L
1120 GOTO 1070
1130 DATA 196,250,262,500,196,250,165,250,147,250,131,2
      50,123,500,220,250,196,500,196,250,294,500,247,250
1140 DATA 196,250,175,250,147,250,131,500,220,250,196,5
      00,196,250,330,500,262,250,196,250,165,250,131,250
1150 DATA 175,500,220,250,262,500,247,225,220,225,196,4
      50,247,350,294,350,262,250,247,250,262,250,9999,99
      99
1160 PRINT "BY UTILIZING THE SOUND":"CAPABILITIES OF TH
      E TI-99/4A":"YOU CAN GENERATE A VARIETY"
1170 PRINT "OF DIFFERENT SOUNDS"::"HERE IS A SAMPLE OF
      THE":"SOUNDS POSSIBLE .....":::::::::::
1180 FOR L=1 TO 20
1190 CALL SOUND(30,-5,0)
1200 NEXT L
1210 FOR D=1 TO 100
1220 NEXT D
1230 FOR L=3000 TO 6000 STEP 100
1240 CALL SOUND(40,L,0)
1250 NEXT L
1260 FOR D=1 TO 100
1270 NEXT D
1280 FOR L=150 TO 250 STEP 2
1290 CALL SOUND(50,L,0)
1300 CALL SOUND(50,L*4,0)
1310 NEXT L
1320 GOTO 1700
1330 PRINT "HERE IS A GAME WRITTEN IN":"THE BASIC LANGU
      AGE TO SHOW":"YOU HOW THE COMPUTER IS"
1340 PRINT "GIVEN INSTRUCTIONS"::"THIS GAME IS CALLED H
      IGH-LOW"::"ALL YOU HAVE TO DO IS GUESS"
1350 PRINT "A NUMBER BETWEEN 1 AND 100"::"THE COMPUTER
      WILL HELP YOU":"FIGURE OUT THE NUMBER":
1360 PRINT "AFTER THE GAME YOU WILL BE":"SHOWN THE ACTU
      AL PROGRAM":"USED TO RUN THE GAME":
1370 PRINT " PRESS ANY KEY TO START"::
```

```

1380 CALL KEY(3,K,S)
1390 IF S=0 THEN 1380
1400 CALL CLEAR
1410 C=0
1420 N=INT(100*RND)+1
1430 INPUT "ENTER YOUR GUESS? ":G
1440 C=C+1
1450 IF G=N THEN 1510
1460 IF G<N THEN 1490
1470 PRINT "HIGH !!!!!":
1480 GOTO 1430
1490 PRINT "LOW !!!!!":
1500 GOTO 1430
1510 PRINT "::CONGRATULATIONS":
1520 PRINT "YOU GOT IT IN";C;"GUESSES":
1530 PRINT TAB(7);"PRESS ANY KEY":TAB(5);"TO SEE THE PR
    OGRAM":
1540 CALL KEY(3,K,S)
1550 IF S=0 THEN 1540
1560 CALL CLEAR
1570 PRINT " 10 CALL CLEAR"
1580 PRINT " 20 C=0"
1590 PRINT " 30 N=INT(100*RND)+1"
1600 PRINT " 40 INPUT ";CHR$(34);"ENTER YOUR":      GUES
    S";CHR$(34);":G"
1610 PRINT " 50 C=C+1"
1620 PRINT " 60 IF G=N THEN 120"
1630 PRINT " 70 IF G<N THEN 100"
1640 PRINT " 80 PRINT ";CHR$(34);"HIGH !!!!!";CHR$(34);
    "":
1650 PRINT " 90 GOTO 30"
1660 PRINT "100 PRINT ";CHR$(34);"LOW !!!!!";CHR$(34);
    "":
1670 PRINT "110 GOTO 30"
1680 PRINT "120 PRINT ::";CHR$(34);"CONGRATU":      LAT
    IONS";CHR$(34);":
1690 PRINT "130 PRINT ";CHR$(34);"YOU GOT IT":      IN";
    CHR$(34);":C";CHR$(34);"GUESSES";CHR$(34);:::::
1700 PRINT " PRESS ANY KEY TO CONTINUE":
1710 CALL KEY(3,K,S)
1720 IF S=0 THEN 1710
1730 GOTO 190
1740 CALL CLEAR
1750 END

```

Analysis and Suggestions

Since this program is an introduction and demonstration program, the analysis and the suggestion sections are not detailed. Just enjoy the demonstration and we will concern ourselves with all the details in the following chapters.

Chapter 2

Graphics of the TI-99/4A

Certainly one of the best features of the TI-99/4A is its graphics capability. The graphics capability of this computer rivals the best in the home computer field, and the programs within this chapter will show you how to exploit this capability even more fully than you already have.

THE ART OF BOXES

This program is a graphics demonstration to show off the colors and graphics of your TI-99/4A. Just type it in and let it run.

Listing 2-1

```
100 REM   THE ART OF BOXES
110 RANDOMIZE
120 CALL CLEAR
130 PRINT TAB(5);"WELCOME TO BOX ART":::::::::
140 PRINT "  PLEASE WAIT WHILE SPECIAL":"  CHARACTERS A
    RE BEING":"  CREATED":::::::::
150 DIM D$(16)
160 FOR L=1 TO 6
170 X$=""
180 FOR L2=1 TO 16
190 READ D$(L2)
200 X$=X$&D$(L2)
```

cont. on next page

Listing 2-1—cont.

```

210 CALL CHAR(127+L,X$)
220 CALL HCHAR(20,9+(L*2),127+L)
230 NEXT L2
240 NEXT L
250 PRINT "  PRESS ANY KEY TO START":
260 CALL KEY(3,K,S)
270 IF S=0 THEN 260
280 CALL CLEAR
290 PRINT "      PRESS ANY KEY TO EXIT"
300 C=INT(14*RND)+3
310 B=INT(14*RND)+3
320 IF C=B THEN 310
330 CALL COLOR(13,C,1)
340 CALL SCREEN(B)
350 RS=INT(20*RND)+2
360 RF=INT(20*RND)+2
370 IF RF<=RS THEN 350
380 CS=INT(28*RND)+3
390 CF=INT(28*RND)+3
400 IF CF<=CS THEN 380
410 CALL HCHAR(RS,CS,128)
420 CALL HCHAR(RS,CS+1,132,CF-CS-1)
430 CALL HCHAR(RF,CS,130)
440 CALL HCHAR(RS,CF,129)
450 CALL HCHAR(RF,CS+1,132,CF-CS-1)
460 CALL HCHAR(RF,CF,131)
470 CALL VCHAR(RS+1,CS,133,RF-RS-1)
480 CALL VCHAR(RS+1,CF,133,RF-RS-1)
490 CALL KEY(3,K,S)
500 IF S<>0 THEN 580
510 GOTO 300
520 DATA 0,0,0,0,3,F,3,F,3,F,3,F,3,C,3,C
530 DATA 0,0,0,0,F,C,F,C,F,C,F,C,3,C,3,C
540 DATA 3,C,3,C,3,F,3,F,3,F,3,F,0,0,0
550 DATA 3,C,3,C,F,C,F,C,F,C,F,C,0,0,0
560 DATA 0,0,0,0,F,F,F,F,F,F,F,0,0,0
570 DATA 3,C,3,C,3,C,3,C,3,C,3,C,3,C
580 CALL CLEAR
590 END

```

Explanation of Program

120-140	Clear screen, print title
150-240	Create special graphics characters to draw boxes
250-280	Wait for user input to start drawing boxes
290	Print exit message
300-320	Set random color variables
330-340	Call color subprograms to color screen and boxes
350-400	Generate random box corners and make sure the starting location is smaller than the finishing location

410-490	Use HCHAR subprogram to draw box
500-510	Check for input to exit program, if not go back and draw another box
520-570	Special character data statements
580-590	Program exit

Important Variables in Program

D\$(16),X\$	= Temporary variables for holding special character data
L,L2	= Loop variables to read and create special characters
K,S	= KEY subprogram variables
C	= Random box color
B	= Random background color
RS,RF	= Screen row starting and finishing coordinates
CS,CF	= Screen column starting and finishing coordinates

Sample Run of Program

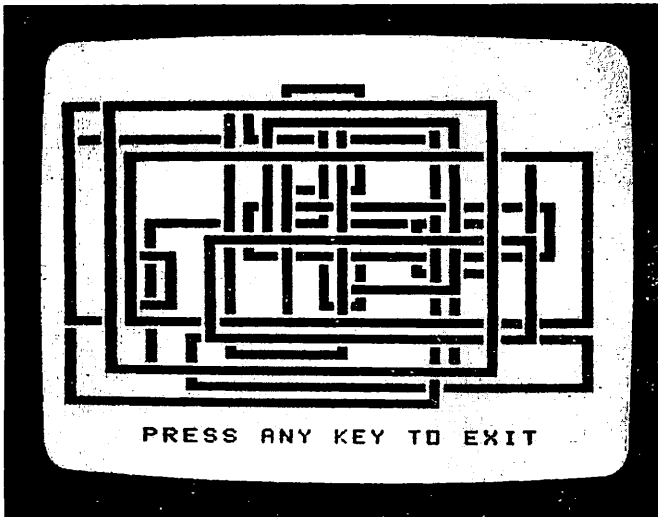


Fig. 2-1.

Suggested Modifications and Projects

1. Try to create more unusual shapes, like triangles or perfect squares.
2. See if you can create the boxes as filled-in blocks instead of just boxes.

PATTERNS

This is a simple, eye-catching demonstration to increase your knowledge of the graphics capabilities of the TI-99/4A.

Listing 2-2

```
100 REM PATTERNS
110 RANDOMIZE
120 CALL CLEAR
130 PRINT TAB(5);"WELCOME TO PATTERNS":;:;:;:;:;:;
140 PRINT " PLEASE WAIT WHILE SPECIAL":" CHARACTERS A
    RE BEING":" CREATED":;:;:;:;:;:;
150 DIM D$(16)
160 FOR L=1 TO 8
170 X$=""
180 FOR L2=1 TO 16
190 READ D$(L2)
200 X$=X$&D$(L2)
210 CALL CHAR(127+L,X$)
220 CALL HCHAR(20,6+(L*2),127+L)
230 NEXT L2
240 NEXT L
250 PRINT " PRESS ANY KEY TO START":;:;
260 CALL KEY(3,K,S)
270 IF S=0 THEN 260
280 CALL CLEAR
290 PRINT " PRESS ANY KEY TO EXIT"
300 R=12
310 C=16
320 B=INT(RND*14)+3
330 A=INT(RND*14)+3
340 IF A=B THEN 330
350 CALL SCREEN(B)
360 CALL COLOR(13,A,1)
370 ON INT(RND*8)+1 GOSUB 410,450,490,530,570,630,690,7
    50
380 CALL KEY(3,K,S)
390 IF S<>0 THEN 910
400 GOTO 320
410 IF R>20 THEN 440
420 R=R+1
430 GOSUB 810
440 RETURN
450 IF R<3 THEN 480
460 R=R-1
470 GOSUB 810
480 RETURN
490 IF C>29 THEN 520
500 C=C+1
510 GOSUB 810
520 RETURN
530 IF C<4 THEN 560
540 C=C-1
550 GOSUB 810
560 RETURN
```

```

570 IF R>20 THEN 620
580 IF C>29 THEN 620
590 R=R+1
600 C=C+1
610 GOSUB 810
620 RETURN
630 IF R>20 THEN 680
640 IF C<4 THEN 680
650 R=R+1
660 C=C-1
670 GOSUB 810
680 RETURN
690 IF R<3 THEN 740
700 IF C>29 THEN 740
710 R=R-1
720 C=C+1
730 GOSUB 810
740 RETURN
750 IF R<3 THEN 800
760 IF C<4 THEN 800
770 R=R-1
780 C=C-1
790 GOSUB 810
800 RETURN
810 CALL HCHAR(R,C,128+INT(RND*8))
820 RETURN
830 DATA 8,0,C,0,E,0,F,0,F,8,F,C,F,E,F,F
840 DATA 0,1,0,3,0,7,0,F,1,F,3,F,7,F,F,F
850 DATA F,F,F,E,F,C,F,8,F,0,E,0,C,0,8,0
860 DATA F,F,7,F,3,F,1,F,0,F,0,7,0,3,0,1
870 DATA F,0,F,0,F,0,F,0,0,F,0,F,0,F,0,F
880 DATA 0,F,0,F,0,F,0,F,F,0,F,0,F,0,F,0
890 DATA 0,0,1,8,3,C,7,E,7,E,3,C,1,8,0,0
900 DATA F,F,F,F,F,F,F,F,F,F,F,F,F,F,F
910 CALL CLEAR
920 END

```

Explanation of Program

- 120-140 Clear screen, print title
- 150-240 Create special graphics characters to draw boxes
- 250-280 Wait for user input to start drawing boxes
- 290 Print exit message
- 300-310 Set initial row and column coordinates
- 320-360 Generate random color variables and call color subprograms
to color screen and boxes
- 370 Generate random number for direction to print the next
character
- 380-390 Get key input and check for input to exit program
- 400-800 Eight subroutines, one for each random direction, to incre-
ment or decrement the row and/or column position for the
next character to be printed

810-820	Subroutine to randomly choose one of the eight special characters and print it on the screen using the HCHAR subprogram
830-900	Special character data statements
910-920	Program exit

Important Variables in Program

D\$(16),X\$	= Temporary variables for holding special character data
L,L2	= Loop variables to read and create special characters
K,S	= KEY subprogram variables
R,C	= Row and column print position for special characters
A,B	= Random color variables

Sample Run of Program

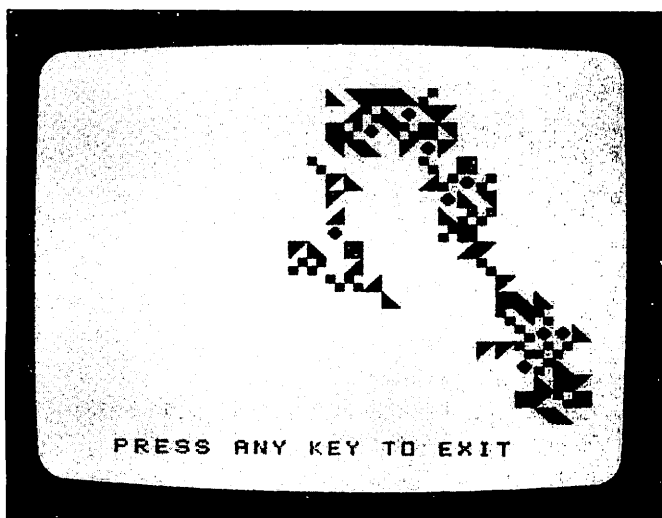


Fig. 2-2.

Suggested Modifications and Projects

1. Change the color variables and special character set so that certain characters have certain colors.
2. Try your own special characters and design some that look best when a pattern is traveling in a certain direction.

MICRO SKETCH

This is a simple graphics program that allows you to move a block around the screen to draw a picture. The E, D, S, and X keys allow moving up, right, left, and down respectively. The T key will allow you to toggle between a block and a blinking block which can be used as an eraser. You will be able to draw any shape you desire on the screen.

Listing 2-3

```

100 REM MICRO SKETCH
110 CALL CLEAR
120 PRINT " WELCOME TO MICRO SKETCH"::::
130 PRINT "THE FOLLOWING KEYS CONTROL":TAB(8);"THE BLOC
    K"::::
140 PRINT " E MOVES THE BLOCK UP"
150 PRINT " S MOVES THE BLOCK LEFT"
160 PRINT " D MOVES THE BLOCK RIGHT"
170 PRINT " X MOVES THE BLOCK DOWN"::::
180 PRINT " T TOGGLES BETWEEN AN":" BLOCK AND A BLINK
    ING": " BLOCK WHICH CAN BE":" USED AS AN ERASER"
    :::
190 PRINT " PRESS ANY KEY TO START"::
200 CALL KEY(O,K,S)
210 IF S=0 THEN 200
220 CALL CLEAR
230 CALL CHAR(128,"FFFFFFFFFFFFFFFF")
240 PRINT TAB(8);"PRESS Q TO QUIT"
250 R=1
260 C=3
270 T=128
280 CALL KEY(3,K,S)
290 CALL HCHAR(R,C,128)
300 CALL HCHAR(R,C,T)
310 IF S=0 THEN 280
320 IF K=81 THEN 520
330 IF (K=69)*(R>1) THEN 390
340 IF (K=83)*(C>3) THEN 410
350 IF (K=68)*(C<31) THEN 430
360 IF (K=88)*(R<22) THEN 450
370 IF (K=84) THEN 470
380 GOTO 280
390 R=R-1
400 GOTO 280
410 C=C-1
420 GOTO 280
430 C=C+1
440 GOTO 280
450 R=R+1
460 GOTO 280
470 IF T=128 THEN 500
480 T=128
490 GOTO 280

```

cont. on next page

Listing 2-3—cont.

```
500 T=32
510 GOTO 280
520 CALL CLEAR
530 END
```

Explanation of Program

110-220 Clear screen, print title and instructions, and wait for user input to start sketcher

230 Create block special character

240 Print exit message

250-260 Set initial row and column coordinates

270 Set T to 128 so solid block printed

280 Call KEY subprogram for key input

290 Print block at row and column coordinates

300 Print block (ASCII 128) or print blank (ASCII 32) by which-ever T is set to, for simulation of solid block or blinking block

310 If no key input, go back to line 280 to check for another key and to reprint block or blinking block

320 If key is "Q" (ASCII 81) then go to program exit

330-360 If a direction key is pressed, check to make sure it will not make sketch go off the screen, and if not go to routines to move block

370 If input key is "T" (ASCII 84) then go to toggle routine

380 Input must have been invalid, go back and get another input

390-460 Four routines to increment or decrement the row or column variables according to input on lines 330-360

470-510 Routine to toggle variable "T" to its opposite value (128 for solid block, 32 for blinking block)

520-530 Program exit

Important Variables in Program

R,C = Row and column position of block
T = Variable to hold 128 or 32 for printing block
K,S = KEY subprogram variables

Sample Run of Program

A sample run of the *Micro Sketch* program is shown in Fig. 2-3.

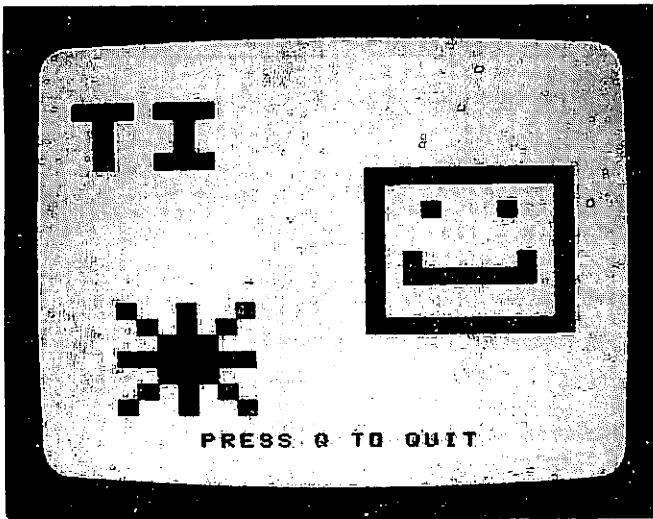


Fig. 2-3.

Suggested Modifications and Projects

1. Try to use more multiple special characters by employing different keys to toggle between them.
2. See if you can make other user definable features like definable block and background colors.

SCRAMBLER

This program, as the name suggests, will make whatever is on the screen unreadable by anyone else. It demonstrates the GCHAR and HCHAR subprograms.

Listing 2-4

```

100 REM  SCRAMBLER
110 DIM D(768)
120 CALL CLEAR
130 PRINT "  WELCOME TO THE SCRAMBLER"::::
140 PRINT "THE TEXT ON THE SCREEN":"WILL BE SAVED IN AN
    ARRAY":"CALLED D(768)"::::
150 PRINT "THEN THE CHARACTERS WILL":"BE SCRAMBLED SO T
    HAT YOU":"CAN NOT READ THEM"::::
160 PRINT "FINALLY, THE CHARACTERS":"WILL BE RESTORED":
    :::

```

cont. on next page

Listing 2-4—cont.

```
170 PRINT "SCRAMBLER NOW RUNNING....":  
180 FOR R=1 TO 24  
190 FOR C=3 TO 30  
200 CALL GCHAR(R,C,X)  
210 D(30*(R-1)+C)=X  
220 IF X=32 THEN 240  
230 CALL HCHAR(R,C,130)  
240 NEXT C  
250 NEXT R  
260 M$="RUN AGAIN? (Y/N)? " "  
270 FOR L=603 TO 627  
280 D(L)=ASC(SEG$(M$,L-602,1))  
290 NEXT L  
300 FOR R=1 TO 24  
310 FOR C=3 TO 30  
320 CALL HCHAR(R,C,D(30*(R-1)+C))  
330 NEXT C  
340 NEXT R  
350 CALL KEY(3,K,S)  
360 IF K=89 THEN 120  
370 IF K=78 THEN 390  
380 GOTO 350  
390 CALL CLEAR  
400 END
```

Explanation of Program

- 110 Dimension array to hold screen characters
- 120-170 Clear screen and print instructions
- 180-250 Routine to get characters from screen, save their value in
 array D(768), and place a strange character on the screen in
 its place
- 260-290 Loop to change one line of the text on the screen to an option
 message
- 300-340 Routine to restore the characters to the screen
- 350-380 User option to re-run or exit the program
- 390-400 Program exit

Important Variables in Program

- D(768) = Array to hold characters from screen
- R,C = Row and column loop variables to get characters from
 screen and replace them
- X = GCHAR subprogram variable
- M\$ = Message string to be placed on screen when it is to be
 restored
- L = Loop to add message string to the characters on screen
- K,S = KEY subprogram variables

Sample Run of Program

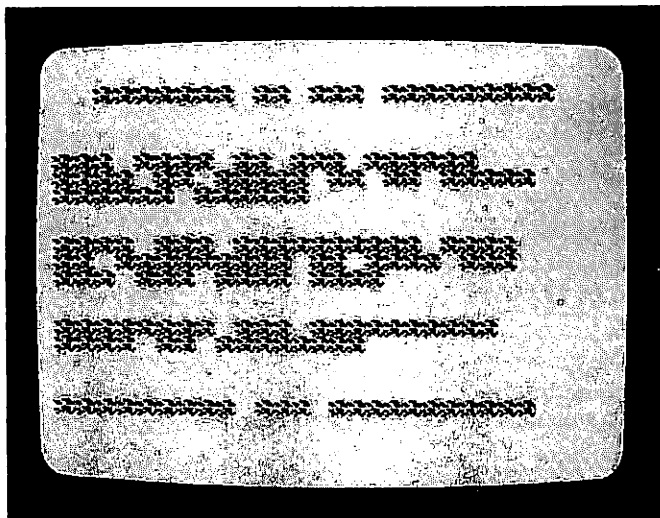


Fig. 2-4.

Suggested Modifications and Projects

1. Use other loops to change the text on the screen before it is restored.
2. Use this routine in other programs to hide information from the user.

CHARACTER GENERATOR

This program will allow you to enter data for your own special characters. It will be very helpful because it allows you to make as many changes to the data as necessary to get exactly the character you want.

Listing 2-5

```

100 REM   CHARACTER GENERATOR
110 CALL CLEAR
120 PRINT TAB(8);"WELCOME TO THE";TAB(6);"CHARACTER GEN
    ERATOR"::::
130 PRINT "THIS PROGRAM WILL ALLOW YOU":"TO ENTER DATA
    TO CREATE":"YOUR OWN SPEC
    IAL CHARACTERS"::::
140 PRINT "THIS PROGRAM WILL BE MOST":"HELPFUL BY USING
    IT IN":"CONJUNCTION WITH THE"
150 PRINT "CHARACTER CODER PROGRAM"::::
160 PRINT TAB(5);"PRESS ANY KEY TO START"::::

```

cont. on next page

Listing 2-5—cont.

```

170 CALL KEY(O,K,S)
180 IF S=0 THEN 170
190 DIM D$(16)
200 FOR L2=1 TO 16
210 D$(L2)="F"
220 NEXT L2
230 CALL CLEAR
240 PRINT "CURRENT DATA FOR BLOCK":TAB(10);"IS"::::::
    ::::TAB(5);"0" 1 "*"
250 PRINT TAB(5);"2" * 3 ***:TAB(5);"4" * 5
    * *:TAB(5);"6" ** 7 ***:TAB(5);"8" * 9 *
    F"
260 PRINT TAB(5);"A" * * B * ***:TAB(5);"C" ** D *
    * *:TAB(5);"E" *** F *****:
270 FOR L=1 TO 16
280 GOSUB 600
290 GOSUB 670
300 M$="CHANGE BLOCK "&STR$(L)&"? (Y/N)? "
310 GOSUB 740
320 CALL KEY(3,K,S)
330 IF K=89 THEN 360
340 IF K=78 THEN 410
350 GOTO 320
360 M$="ENTER NEW VALUE (0-F)? "
370 GOSUB 740
380 CALL KEY(3,K,S)
390 IF ((K>47)*(K<58))+((K>64)*(K<71))THEN 400 ELSE 380
400 D$(L)=CHR$(K)
410 GOSUB 600
420 GOSUB 670
430 M$="BLOCK "&STR$(L)&" CORRECT? (Y/N)?"
440 GOSUB 740
450 CALL KEY(3,K,S)
460 IF K=89 THEN 490
470 IF K=78 THEN 360
480 GOTO 450
490 NEXT L
500 CALL CLEAR
510 PRINT "THE DATA FOR YOUR":"CHARACTER IS"::::
520 FOR L5=1 TO 16
530 PRINT D$(L5);
540 NEXT L5
550 PRINT :::"MAKE MORE CHANGES? (Y/N)?":::
560 CALL KEY(3,K,S)
570 IF K=89 THEN 230
580 IF K=78 THEN 790
590 GOTO 560
600 X$=""
610 FOR L3=1 TO 16
620 X$=X$&D$(L3)
630 NEXT L3
640 CALL CHAR(128,X$)
650 CALL HCHAR(8,15,128)
660 RETURN
670 B$=STR$(L)

```

```

680 IF LEN(B$)=2 THEN 700
690 B$=" "&B$
700 CALL HCHAR(2,26,ASC(SEG$(B$,1,1)))
710 CALL HCHAR(2,27,ASC(SEG$(B$,2,1)))
720 CALL HCHAR(4,16,ASC(D$(L)))
730 RETURN
740 FOR L4=1 TO LEN(M$)
750 PC=ASC(SEG$(M$,L4,1))
760 CALL HCHAR(11,L4+2,PC)
770 NEXT L4
780 RETURN
790 CALL CLEAR
800 END

```

Explanation of Program

110-180	Clear screen, print instructions, and wait for input to start
190-220	Create block special character to start with
230-260	Print character bit codes for easy reference
270	Start main loop to create the character
280	Use subroutine at line 600 to print current character to the screen
290	Use subroutine at line 670 to print current data for each sub-block
300-350	User option to change each sub-block
360-420	Routine to get new value for sub-block and reprint the new block to the screen
430-480	User option to re-do his last change
490	Completion of the main loop
500-540	Print out character data after changes completed
550-590	User option to make even more changes to the character
600-660	Subroutine to print special character to screen
670-730	Subroutine to print current data values to screen
740-780	Subroutine to print message to screen
790-800	Program exit

Important Variables in Program

D\$(16)	= Array to hold character data
L2	= Loop variable to fill character solid
L	= Main loop to enter and change character data
M\$	= Message string for print subroutine
L3	= Loop variable to take the 16 data items and create a single data string for the character
X\$	= Data created by L3
B\$	= Temporary string variable to print data values

- L4 = Loop variable to print message string to screen using HCHAR subprogram
 PC = Loop variable used in print subroutine to print message string
 K,S = KEY subprogram variables

Sample Run of Program

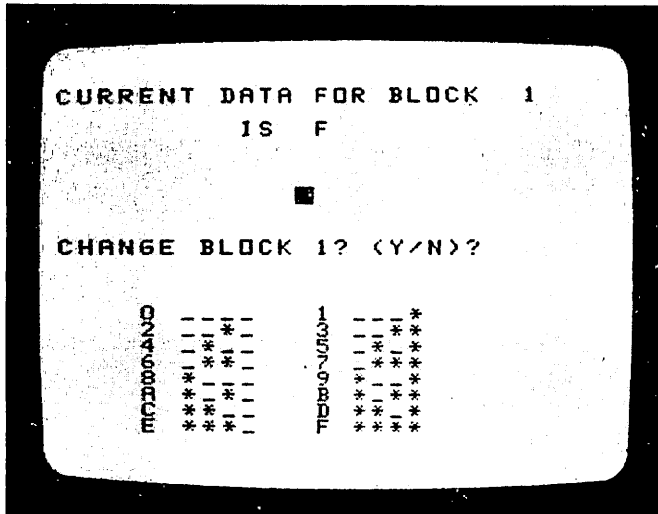


Fig. 2-5.

Suggested Modifications and Projects

1. Have the program allow the creation of multiple special characters that can be added together for graphics animation characters.
2. See if you can make the current character move across the top of the screen while you make your changes.

CHARACTER CODER

This program will also help you create your own special characters. It works differently than the last program though because it allows you to choose the bit pattern for the character you are designing. The program then takes the bit pattern and gives the data statement needed for the CHAR subprogram. It will also allow you to make as many changes to the data as necessary to get exactly the character you want.

Listing 2-6

```

100 REM CHARACTER CODER
110 CALL CLEAR
120 PRINT TAB(8); "WELCOME TO":TAB(4); "THE CHARACTER COD
ER":::
130 PRINT "THIS PROGRAM TAKES THE": "PICTURE THAT YOU PU
T ON THE": "SCREEN AND FIGURES THE DATA"
140 PRINT "STATEMENT NEEDED TO CREATE": "A SPECIAL CHARA
CTER": "TO MAKE THE PICTURE YOU MUST"
150 PRINT "ENTER A LETTER FOR A ROW,": "THEN A LETTER FO
R THE COLUMN": "YOU CAN TURN OFF A MISTAKE"
160 PRINT "BY RE-ENTERING THE SAME": "LOCATION":::
170 PRINT " PRESS ANY KEY TO START":::
180 CALL KEY(3,K,S)
190 IF S=0 THEN 180
200 CALL CHAR(128,"FFFFFFFFFFFFFFFF")
210 CALL CLEAR
220 PRINT " PRESS Z TO FIGURE DATA";
230 FOR L=1 TO 8
240 CALL HCHAR(5,L+12,L+64)
250 CALL HCHAR(5+L,12,L+64)
260 CALL HCHAR(14,L+12,L+64)
270 CALL HCHAR(5+L,21,L+64)
280 NEXT L
290 P$=" ROW? COLUMN? "
300 GOSUB 900
310 CALL KEY(3,K,S)
320 IF S=0 THEN 310
330 IF K=90 THEN 500
340 IF (K<64)+(K>72) THEN 310
350 IF F=1 THEN 420
360 F=1
370 CALL HCHAR(18,14,K)
380 S1=K
390 FOR D=1 TO 100
400 NEXT D
410 GOTO 310
420 F=0
430 CALL HCHAR(18,25,K)
440 CALL GCHAR(S1-59,K-52,X)
450 IF X=128 THEN 480
460 CALL HCHAR(S1-59,K-52,128)
470 GOTO 290
480 CALL HCHAR(S1-59,K-52,32)
490 GOTO 290
500 P$=" FIGURING DATA.... "
510 GOSUB 900
520 FOR L=1 TO 8
530 CALL GCHAR(L+5,13,X1)
540 CALL GCHAR(L+5,14,X2)
550 CALL GCHAR(L+5,15,X3)
560 CALL GCHAR(L+5,16,X4)
570 T(L,1)=(8*ABS(X1=128))+(4*ABS(X2=128))+(2*ABS(X3=12
8))+ABS(X4=128)
580 CALL GCHAR(L+5,17,X1)

```

cont. on next page

Listing 2-6—cont.

```
590 CALL GCHAR(L+5,18,X2)
600 CALL GCHAR(L+5,19,X3)
610 CALL GCHAR(L+5,20,X4)
620 T(L,2)=(8*ABS(X1=128))+(4*ABS(X2=128))+(2*ABS(X3=12
8))+ABS(X4=128)
630 NEXT L
640 FOR L=1 TO 8
650 FOR L2=1 TO 2
660 IF T(L,L2)>9 THEN 690
670 CALL HCHAR(20,L+L+6+L2,T(L,L2)+48)
680 GOTO 700
690 CALL HCHAR(20,L+L+6+L2,T(L,L2)+55)
700 NEXT L2
710 NEXT L
720 P$="      MORE CHANGES? (Y/N)"
730 GOSUB 900
740 CALL KEY(3,K,S)
750 IF K=78 THEN 780
760 IF K=89 THEN 290
770 GOTO 740
780 P$="      RUN AGAIN? (Y/N)  "
790 GOSUB 900
800 CALL KEY(3,K,S)
810 IF K=78 THEN 940
820 IF K=89 THEN 840
830 GOTO 800
840 CALL CLEAR
850 FOR L=1 TO 8
860 T(L,1)=0
870 T(L,2)=0
880 NEXT L
890 GOTO 113
900 FOR L=1 TO LEN(P$)
910 CALL HCHAR(18,L+2,ASC(SEG$(P$,L,1)))
920 NEXT L
930 RETURN
940 CALL CLEAR
950 END
```

Explanation of Program

- | | |
|---------|--|
| 110-190 | Clear screen, print instructions, and wait for input to start |
| 200 | Create block special character to start with |
| 210-220 | Print instruction message |
| 230-280 | Print letters around area to enter data for row and column reference |
| 290-340 | User options to input row and column to change a bit to on or off, or to figure the data |
| 350-410 | Takes input and verifies that it is a valid row and column |
| 420-490 | Routine to change bit to opposite value and print change to screen |

500-510	Print message to acknowledge user input to figure data
520-630	Loop to get bits from block, figure total value, and store it in array D(8,2)
640-710	Loop to take values and print character data statement to screen
720-770	User option to make more changes
780-830	User option to start program over for a new character
840-890	If start over, reset values in data array to zero and go to the beginning of the program
900-930	Subroutine to print messages to screen
940-950	Program exit

Important Variables in Program

D\$(16)	= Array to hold character data
L,L2	= Loop variables to print characters, get characters, and figure data
P\$	= Message string for print subroutine
F	= Flag to identify if input is row or column
S1	= Selection one or row input value
D	= Delay loop variable
T(8,2)	= Array to hold and add up data values for character
K,S	= KEY subprogram variables

Sample Run of Program

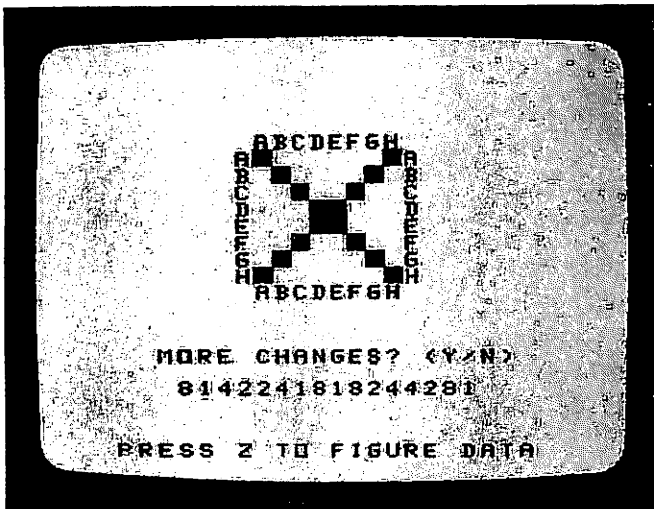


Fig. 2-6.

Suggested Modifications and Projects

1. Try the modifications to the last program.
2. Simultaneously display character in its true size on the screen while you are creating it.

Chapter 3

Sounds of the TI-99/4A

The sounds that can be produced by using the TI-99/4A are as amazing as the graphics, especially considering the economical price of the computer. Here are a total of seven programs, showing off the clear, crisp, versatile music of the TI-99/4A. Included are four songs and three programs to help you create your own sounds and music. Sample runs of the songs are not included because all you have to do is listen to enjoy these programs.

OUR FIRST SONG

Here is a little number for your computer that uses a single voice in the SOUND subprogram of the TI-99/4A.

Listing 3-1

```
100 REM  OUR FIRST SONG
110 CALL CLEAR
120 PRINT " WELCOME TO OUR FIRST SONG":::::::::::::
130 PRINT "  PRESS ANY KEY TO START":TAB(10);"THE MUSI
    C":::
140 CALL KEY(3,K,S)
150 IF S=0 THEN 140
160 CALL CLEAR
```

cont. on next page

Listing 3-1—cont.

```
170 PRINT TAB(6);"PLAYING SONG.....":::::::::::::
180 READ T,D
190 IF D=9999 THEN 280
200 CALL SOUND(D,T,0)
210 FOR L=1 TO 20
220 NEXT L
230 GOTO 180
240 DATA 196,250,262,500,196,250,165,250,147,250,131,25
    0,123,500,220,250,196,500,196,250
250 DATA 294,500,247,250,196,250,175,250,147,250,131,50
    0,220,250,196,500,196,250,330,500
260 DATA 262,250,196,250,165,250,131,250,175,500,220,25
    0,263,500,247,225,220,225,196,450
270 DATA 247,350,294,350,262,250,247,250,262,250,9999,9
    999
280 CALL CLEAR
290 END
```

Explanation of Program

120-150	Clear screen, print title, wait for input to start
160-170	Print song playing message
180-190	Read in sound data until an end value reached (9999)
200	Call SOUND subprogram
210-220	Short delay so notes not all run together
230	Go back and read another note
240-270	Sound data
280-290	Program exit

Important Variables in Program

T	= Tone or frequency of note
D	= Duration of note
L	= Delay loop variable
K,S	= KEY subprogram variables

Suggested Modifications and Projects

1. See if you can make this into a song 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.

COMPUTER LULLABY

Here is another little number for your TI-99/4A.

Listing 3-2

```

100 REM  COMPUTER LULLABY
110 CALL CLEAR
120 PRINT TAB(9);"WELCOME TO":TAB(6);"COMPUTER LULLABY"
   ::::::::::::
130 PRINT "    PRESS ANY KEY TO START":TAB(10);"THE MUSI
   C":::
140 CALL KEY(3,K,S)
150 IF S=0 THEN 140
160 CALL CLEAR
170 PRINT TAB(6);"PLAYING SONG....."::::::::::::
180 READ T,D
190 IF D=9999 THEN 300
200 CALL SOUND(D,T,0)
210 FOR L=1 TO 30
220 NEXT L
230 GOTO 180
240 DATA 262,125,294,125,294,125,262,125,294,750,262,12
   5,294,125,330,125,294,250,262,250,247,125,262,125
250 DATA 262,125,247,125,262,750,247,125,262,125,294,12
   5,262,250,247,250,220,125,247,125,247,125,220,125
260 DATA 247,700,220,125,247,125,262,125,247,250,220,25
   0,349,1000,294,125,330,125,330,125,294,125,330,750
270 DATA 294,125,330,125,349,125,330,250,294,250,262,12
   5,294,125,294,125,262,125,294,750,262,125,294,125
280 DATA 330,125,294,250,262,250,247,125,262,125,262,12
   5,247,125,262,750,247,125,262,125,294,125,262,250
290 DATA 247,250,392,1000,247,125,262,125,292,125,262,2
   50,247,250,262,400,9999,9999
300 CALL CLEAR
310 END

```

Explanation of Program

120-150	Clear screen, print title, wait for input to start
160-170	Print song playing message
180-190	Read in sound data until an end value reached (9999)
200	Call SOUND subprogram
210-220	Short delay so notes not all run together
230	Go back and read another note
240-290	Sound data
300-310	Program exit

Important Variables in Program

T	= Tone or frequency of note
D	= Duration of note
L	= Delay loop variable
K,S	= KEY subprogram variables

Suggested Modifications and Projects

1. Try the modifications for the last program.
2. See if you can add some graphics to the screen while the song is playing, perhaps having the screen appear in multiple colors to the beat of the music.

BACH MINUET

For you classical fans, here is a soothing piece from the musical genius Bach.

Listing 3-3

```

100 REM  BACH MINUET
110 CALL CLEAR
120 PRINT TAB(9);"WELCOME TO":TAB(6);"THE BACH MINUET":
    :::::::::::::::
130 PRINT "    PRESS ANY KEY TO START":TAB(10);"THE MUSI
    C":::
140 CALL KEY(3,K,S)
150 IF S=0 THEN 140
160 CALL CLEAR
170 PRINT TAB(6);"PLAYING SONG....."::::::::::::::
180 READ T,D
190 IF D=9999 THEN 340
200 CALL SOUND(D*9,T,0)
210 FOR L=1 TO 40
220 NEXT L
230 GOTO 180
240 DATA 294,30,196,20,220,20,247,20,262,20,294,30,196,
    30,196,30,330,30,262,20,294,20,330,20,370,20,392,30
250 DATA 196,30,196,30,262,30,294,20,262,20,247,20,220,
    20,247,30,262,20,247,20,220,20,196,20,185,30,196,20
260 DATA 220,20,247,20,196,20,220,40,294,30,196,20,220,
    20,247,20,262,20,294,30,196,30,196,30,330,30,262,20
270 DATA 294,20,330,20,370,20,392,30,196,30,196,30,262,
    30,294,20,262,20,247,20,220,20,247,30,262,20,247,20
280 DATA 220,20,196,20,220,30,247,20,220,20,196,20,185,
    20,196,40,294,30,196,20,220,20,247,20,262,20,294,30
290 DATA 196,20,196,30,330,30,262,20,294,20,330,20,370,
    20,392,30,196,30,196,30,262,30,294,20,262,20,247,20
300 DATA 220,20,247,30,262,20,247,20,220,20,196,20,185,
    30,196,20,220,20,247,20,196,20,220,40,294,30,196,20
310 DATA 220,20,247,20,262,20,294,30,196,30,196,30,330,
    30,262,20,294,20,330,20,370,20,392,30,196,30,196,30
320 DATA 262,30,294,20,262,20,247,20,220,20,247,30,262,
    20,247,20,220,20,196,20,220,30,247,20,220,20,196,20
330 DATA 185,20,196,30,9999,9999
340 CALL CLEAR
350 END

```

Explanation of Program

120-150	Clear screen, print title, wait for input to start
160-170	Print song playing message
180-190	Read in sound data until an end value reached (9999)
200	Call SOUND subprogram
210-220	Short delay so notes not all run together
230	Go back and read another note
240-330	Sound data
340-350	Program exit

Important Variables in Program

T	= Tone or frequency of note
D	= Duration of note
L	= Delay loop variable
K,S	= KEY subprogram variables

Suggested Modifications and Projects

1. Try the modifications to the last two programs.
2. Change the program to contain more than one song and let the user decide which one he wants to hear.

SOUNDS OF DIXIE

Here is a song that uses two voices at the same time to provide an amazing harmonic effect.

Listing 3-4

```

100 REM  SOUNDS OF DIXIE
110 CALL CLEAR
120 PRINT TAB(9);"WELCOME TO":TAB(5);"THE SOUNDS OF DIX
    IE"::::::::::::::::::
130 PRINT "  PRESS ANY KEY TO START":TAB(10);"THE MUSI
    C"::::
140 CALL KEY(3,K,S)
150 IF S=0 THEN 140
160 CALL CLEAR
170 PRINT TAB(6);"PLAYING SONG....."::::::::::::::::::
180 V2=30
190 T2=7000
200 FOR L=1 TO 56
210 READ T,D
220 IF (L=1)+(L=9)+(L=13)+(L=22)+(L=27)+(L=32)+(L=40)+(
    L=50)+(L=57) THEN 340
230 IF (L=5)+(L=18)+(L=36)+(L=46) THEN 370

```

cont. on next page

Listing 3-4—cont.

```
240 IF (L=12)+(L=25)+(L=28)+(L=53) THEN 400
250 CALL SOUND(D*7,T,0,T2,V2)
260 IF L=13 THEN 310
270 IF L=28 THEN 310
280 IF L=41 THEN 310
290 NEXT L
300 GOTO 470
310 FOR L1=1 TO 100
320 NEXT L1
330 GOTO 290
340 V2=8
350 T2=262
360 GOTO 250
370 V2=8
380 T2=349
390 GOTO 250
400 V2=8
410 T2=196
420 GOTO 250
430 DATA 330,40,294,20,262,20,330,20,294,20,262,20,523,
40,440,20,535,50,392,50,330,40,262,40,294,50,330,40
440 DATA 294,20,262,20,330,20,294,20,262,40,523,40,440,
20,523,40,392,40,330,20,262,20,294,40,294,40,262,40
450 DATA 494,40,523,20,587,40,392,40,392,40,440,20,392,
40,523,40,523,40,440,40,349,40,440,40,394,40,330,40
460 DATA 294,20,262,20,330,20,294,20,262,40,523,40,440,
20,523,40,392,40,330,20,262,20,294,40,294,20,262,40
470 CALL CLEAR
480 END
```

Explanation of Program

- | | |
|---------|--|
| 120-150 | Clear screen, print title, wait for input to start |
| 160-170 | Print song playing message |
| 180-190 | Set initial frequency and volume for voice 2 |
| 200 | Start of loop to read in 56 notes and to keep track of how many notes have been played |
| 210 | Read in sound data |
| 220-240 | At certain notes, go to lines 340-420 to change frequency and volume of voice 2 |
| 250 | Call SOUND subprogram |
| 260-280 | At certain notes go to lines 310-330 to add a longer delay than normal between notes |
| 290 | Complete the loop and go get another note |
| 310-330 | Routine to add delay between two notes |
| 340-420 | Three routines to change frequency and volume of voice 2 |
| 430-460 | Sound data |
| 470-480 | Program exit |

Important Variables in Program

T,T2 = Frequency of voice 1 and voice 2
 D = Duration of notes
 V2 = Volume of voice 2
 L = Loop variable to keep track of number of notes played
 L1 = Delay loop variable
 K,S = KEY subprogram variables

Suggested Modifications and Projects

1. Try the modifications to the other songs.
2. Try adding a third voice to this song.

PIANO MACHINE

When coding music into your programs, you may find it rather monotonous having to look up the frequencies for every note you need. This program will put an end to this monotony by transforming your keyboard into a five octave piano keyboard. The program will play the note you choose so you can hear it and give you the data necessary for the SOUND subprogram.

Listing 3-5

```

100 REM PIANO MACHINE
110 CALL CLEAR
120 PRINT " WELCOME TO PIANO MACHINE":
130 PRINT "THIS PROGRAM WILL HELP YOU":"TO WRITE SONGS
    ON YOUR":"COMPUTER":
140 PRINT "YOU CAN PLAY NOTES BY":"PRESSING D,F,G,H,J,K
    ,":"AND L FOR THE WHITE KEYS"
150 PRINT "AND BY PRESSING R,T,U,I,":"AND O FOR THE BLA
    CK KEYS":
160 PRINT "YOU CAN CHANGE THE DURATION":"OF THE NOTES B
    Y PRESSING":"KEYS 1 THROUGH 9":
170 PRINT "YOU CAN CHANGE THE OCTAVE":"BY PRESSING THE
    C,V,B,N,":"AND M KEYS":
180 PRINT " PRESS ANY KEY TO START":
190 D=240
200 O=4
210 CALL KEY(3,K,S)
220 IF S=0 THEN 210
230 CALL CLEAR
240 PRINT "          R   T   U   I   O":"NOTE":          C#
    D# F# G# A#":
250 PRINT "          D   F   G   H   J   K   L":"NOTE":          C
    D   E   F   G   A   B":
    
```

cont. on next page

Listing 3-5—cont.

```
260 PRINT "          C V B N M": "OCTAVE"; TAB(11); "*" ; : "
      3 4 5 6 7" : : :
270 PRINT "          1 2 3 4 5 6 7 8 9"
280 PRINT "LENGTH"; TAB(15); "*" ; : : :
290 PRINT TAB(9); "DURATION  FREQUENCY": "DATA"; TAB(12); "
      240" : : :
300 PRINT TAB(7); "PRESS Q TO QUIT"
310 CALL KEY(3,K,S)
320 IF S=0 THEN 310
330 T=0
340 R=2
350 IF K<>81 THEN 370
360 GOTO 1110
370 IF K<>82 THEN 400
380 T=139
390 C=10
400 IF K<>84 THEN 430
410 T=156
420 C=14
430 IF K<>85 THEN 460
440 T=185
450 C=18
460 IF K<>73 THEN 490
470 T=208
480 C=22
490 IF K<>79 THEN 520
500 T=233
510 C=26
520 IF T<>0 THEN 980
530 R=7
540 IF K<>68 THEN 570
550 T=131
560 C=10
570 IF K<>70 THEN 600
580 T=147
590 C=13
600 IF K<>71 THEN 630
610 T=165
620 C=16
630 IF K<>72 THEN 660
640 T=175
650 C=19
660 IF K<>74 THEN 690
670 T=196
680 C=22
690 IF K<>75 THEN 720
700 T=220
710 C=25
720 IF K<>76 THEN 750
730 T=247
740 C=28
750 IF (K<49)+(K>57) THEN 840
760 CALL HCHAR(17,INT(D/30)+9,32)
770 D=(K-48)*60
780 CALL HCHAR(17,INT(D/30)+9,42)
```

```

790 P$=STR$(D)&" "
800 R=21
810 C=13
820 GOSUB 1070
830 GOTO 310
840 IF (K<>67)*(K<>86)*(K<>66)*(K<>78)*(K<>77) THEN 970
850 CALL HCHAR(12,2*O+5,32)
860 IF K<>67 THEN 880
870 O=3
880 IF K<>86 THEN 900
890 O=4
900 IF K<>66 THEN 920
910 O=5
920 IF K<>78 THEN 940
930 O=6
940 IF K<>77 THEN 960
950 O=7
960 CALL HCHAR(12,2*O+5,42)
970 IF T=0 THEN 310
980 CALL HCHAR(R,C,42)
990 T=T*2^(O-3)
1000 CALL SOUND(D,T,0)
1010 CALL HCHAR(R,C,32)
1020 P$=STR$(T)&" "
1030 R=21
1040 C=23
1050 GOSUB 1070
1060 GOTO 310
1070 FOR L=1 TO LEN(P$)
1080 CALL HCHAR(R,L+C,ASC(SEG$(P$,L,1)))
1090 NEXT L
1100 RETURN
1110 CALL CLEAR
1120 END

```

Explanation of Program

- | | |
|---------|---|
| 110-180 | Clear screen, print instructions |
| 190-200 | Set initial note duration and octave |
| 210-220 | Wait for input to start |
| 230-300 | Print piano keyboard on screen |
| 310-320 | Get input from computer keyboard |
| 330 | Reset frequency value to zero |
| 340 | Set print row for sharp notes |
| 350-360 | Determine if input was "Q" to quit, if it was, then go to program exit |
| 370-520 | Determine if input was for sharp note, if it was, then set frequency, set print column, and go play note |
| 530 | Set print row for regular notes |
| 540-740 | Determine if input was for regular note, if it was then set frequency, set print column, and go play note |
| 750-830 | Determine if input was for duration change, if it was then |

set duration, print new duration asterisk, print new duration data value to screen, and go get the next input

840-960 Determine if input was for octave change, if it was, then set octave, print new octave to screen, and go get next input

970 If variable T still zero then input must have been invalid, so go get another input

980-1050 Print note to screen and play note

1060 Go get next input

1070-1100 Subroutine to print out messages to screen

1110-1120 Program exit

Important Variables in Program

T = Tone or frequency of note

D = Duration of note

O = Octave of note

R,C = Row and column to print on

P\$ = Message string for print subroutine

L = Loop variable for print subroutine

K,S = KEY subprogram variables

Sample Run of Program

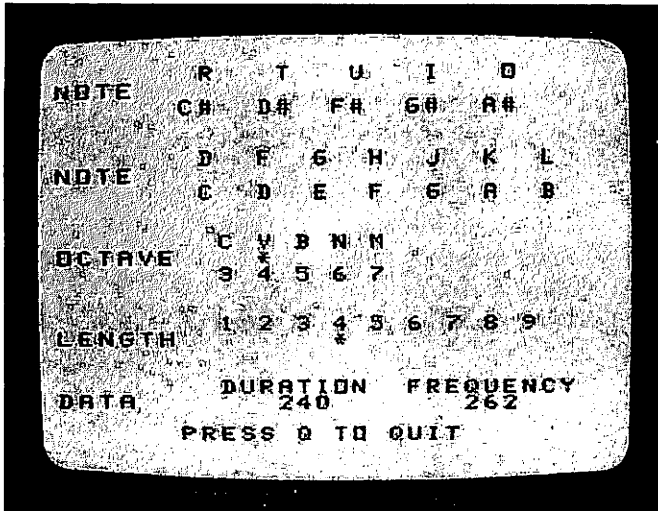


Fig. 3-1.

Suggested Modifications and Projects

1. See if you can convert the program so the duration is dependent on how long you hold the key down.
2. Try to convert the program to print the notes on a music sheet using graphics characters instead of the method to display the keyboard used in this program.

MUSIC MACHINE

This program also makes it easier for you play music on your TI-99/4A. The *Music Machine* allows you to enter notes of music directly without having to worry about programming the frequencies or durations. To play your song, just type "XXX" when the computer asks you for another note.

Listing 3-6

```

100 REM MUSIC MACHINE
110 DIM N$(12),F(12),N(100),D(100),A$(100)
120 CALL CLEAR
130 PRINT TAB(8);"WELCOME TO":TAB(5);"THE MUSIC MACHINE
   "::::
140 PRINT "THIS PROGRAM WILL HELP YOU":;"GET STARTED WR
   ITING YOUR":;"OWN SONGS FOR THE TI-99/4A":;;;
150 PRINT "PRESS ANY KEY TO CONTINUE":;;
160 CALL KEY(3,K,S)
170 IF S=0 THEN 160
180 CALL CLEAR
190 PRINT "FOR EACH NOTE ENTER IN THE":"FOLLOWING ORDER
   "::::"1. NOTE (A-G) OR (A#-G#)":"2. OCTAVE (3-7)"
200 PRINT "3. LENGTH S-SIXTEENTH":TAB(12);"E-EIGHTH":T
   AB(12);"Q-QUARTER":TAB(12);"H-HALF":TAB(12);"W-WHOL
   E"::::
210 PRINT "HERE ARE SOME EXAMPLES":;
220 PRINT "A4Q A#4Q G#5S B4H C4H":;
230 PRINT "THE COMPUTER WILL PLAY YOUR":"SONG AND SHOW
   YOU THE":"NECESSARY DATA STATEMENTS":;
240 PRINT " PRESS ANY KEY TO START":;
250 CALL KEY(3,K,S)
260 IF S=0 THEN 250
270 CALL CLEAR
280 FOR L=1 TO 12
290 READ N$(L),F(L)
300 NEXT L
310 PRINT "TYPE XXX FOR THE NOTE TO":"PLAY YOUR SONG":;
   :
320 C=C+1
330 PRINT ":"TYPE NOTE #";STR$(C);
340 INPUT A$(C)
350 IF A$(C)="XXX" THEN 700

```

cont. on next page

Listing 3-6—cont.

```
360 IF (LEN(A$(C))=3)+((LEN(A$(C))=4)*(SEG$(A$(C),2,1)=
   "#"))THEN 390
370 PRINT "INVALID FORMAT - TRY AGAIN"
380 GOTO 330
390 T$=SEG$(A$(C),1,2)
400 IF SEG$(T$,2,1)="#" THEN 420
410 T$=SEG$(T$,1,1)
420 FOR L=1 TO 12
430 IF T$=N$(L)THEN 460
440 NEXT L
450 GOTO 370
460 N(C)=F(L)
470 T$=SEG$(A$(C),LEN(A$(C))-1,1)
480 IF (ASC(T$)<51)+(ASC(T$)>55)THEN 370
490 O=VAL(T$)
500 IF (O<3)+(O>7)THEN 370
510 N(C)=N(C)*2^(O-3)
520 T$=SEG$(A$(C),LEN(A$(C)),1)
530 IF (T$<>"S")*(T$<>"E")*(T$<>"Q")*(T$<>"H")*(T$<>"W"
   )THEN 370
540 IF T$<>"S" THEN 570
550 D(C)=25
560 GOTO 670
570 IF T$<>"E" THEN 600
580 D(C)=50
590 GOTO 670
600 IF T$<>"Q" THEN 630
610 D(C)=100
620 GOTO 670
630 IF T$<>"H" THEN 660
640 D(C)=200
650 GOTO 670
660 D(C)=400
670 CALL SOUND(D(C),N(C),0)
680 IF C=100 THEN 700
690 GOTO 320
700 CALL CLEAR
710 PRINT TAB(6);"PLAYING SONG.....":::::::::
720 FOR L=1 TO C-1
730 CALL SOUND(D(L),N(L),0)
740 FOR L2=1 TO 10
750 NEXT L2
760 NEXT L
770 PRINT "    PLAY SONG AGAIN? (Y/N)":::
780 CALL KEY(3,K,S)
790 IF K=78 THEN 820
800 IF K=89 THEN 700
810 GOTO 780
820 PRINT "    RE-RUN PROGRAM? (Y/N)":::
830 CALL KEY(3,K,S)
840 IF K=78 THEN 910
850 IF K=89 THEN 870
860 GOTO 830
870 CALL CLEAR
```

```

880 C=0
890 GOTO 310
900 DATA C,131,C#,139,D,147,D#,156,E,165,F,175,F#,185,G
    ,196,G#,208,A,220,A#,233,B,247
910 CALL CLEAR
920 END

```

Explanation of Program

110	Dimension arrays to hold notes, frequencies, and durations
120-270	Clear screen, print instructions, wait for input to start
280-300	Loop to read note names and frequencies
310	Print exit instruction message
320	Increment number of valid notes entered counter
330-340	Input note
350	If input "XXX", then exit program
360	Check length of input for possible invalid input, if not go to line 390
370-380	Print invalid note message and go get next attempt
390-410	Set temporary variable T\$ to first one or two characters of the input to determine note chosen
420-440	Loop to match input to note names
450	If no match, go to invalid note message at line 370
460	Set note frequency according to note chosen
470	Set T\$ to next to last character of input
480	Determine if octave chosen in input is valid, if not go to invalid note message
490-510	Set O equal to the octave, adjust frequency of the note by the octave
520	Set T\$ to last character to set note length
530	If invalid go to invalid note message
500-660	Use T\$ to set the duration of the note
670	Call SOUND subprogram to play note
680	If 100 notes entered then go play song
690	Go get another note
700-760	Loop to play song
770-810	User option to replay song
820-860	User option to enter another song
870-890	Reset note counter and go to beginning of program to enter a new song
900	Data for note names and frequencies
910-920	Program exit

Important Variables in Program

N\$(12),F(12)	= Arrays to hold note names and frequencies
N(100),D(100)	= Arrays to hold song notes and frequency
A\$(100)	= Array to hold input notes
C	= Number of valid notes entered
T\$	= Temporary variable to determine note, octave, and duration of input notes
L,L2	= Loop variables to play song
K,S	= KEY subprogram variables

Sample Run of Program

```
TYPE XXX FOR THE NOTE
TO PLAY YOUR SONG

TYPE NOTE #1? E4H
TYPE NOTE #2? A3Q

TYPE NOTE #3? D2Q
INVALID FORMAT—TRY AGAIN
TYPE NOTE #3? D3Q
```

Fig. 3-2.

Suggested Modifications and Projects

1. Try to make the program display the notes on a sheet of music as the music is played. This would be a very good test of your skills with graphics.
2. Make the program into an editor so that certain notes of the song can be changed at will, rather than the whole song having to be retyped every time.

SOUND DEVELOPER

This program can help you design special sound effects for your own games. There are seven different types of sound that you will be able to choose from. Within each type of sound you will be able to set frequencies, volumes, durations, delays, and number repetitions. By having such a wide variety of options, it is amazing how many different types of sounds you can create.

Listing 3-7

```

100 REM  SOUND DEVELOPER
110 CALL CLEAR
120 PRINT TAB(9);"WELCOME TO":TAB(5);"THE SOUND DEVELOP
    ER":::
130 PRINT "THIS PROGRAM WILL HELP YOU":;"DESIGN DIFFERE
    NT SOUNDS":;"FOR YOUR OWN PROGRAMS":::
140 GOTO 1650
150 CALL CLEAR
160 PRINT TAB(8);"SOUND MENU"::::TAB(5);"1. SINGLE TONE
    S":TAB(5);"2. SINGLE UP&DOWN":TAB(5);"3. HIGH-LOW"
170 PRINT TAB(5);"4. HIGH-LOW UP&DOWN":TAB(5);"5. PERIO
    DIC NOISE":TAB(5);"6. WHITE NOISE"
180 PRINT TAB(5);"7. TWO VOICES":TAB(5);"8. EXIT PROGRA
    M"::::TAB(5);"YOUR CHOICE? (1-8)"::::
190 CALL KEY(3,K,S)
200 IF (S=0)+(K<49)+(K>56)THEN 190
210 CALL CLEAR
220 ON K-48 GOTO 230,580,820,1050,1280,1320,1360,1690
230 PRINT : "ENTER FREQUENCY": "TYPE (110-44733)";
240 INPUT F
250 IF (F<110)+(F>44733)THEN 230
260 PRINT : "ENTER DURATION": "TYPE (-4250 TO 4250)";
270 INPUT D
280 IF (D=0)+(D<-4250)+(D>4250)THEN 260
290 IF FL=4 THEN 1440
300 PRINT : "ENTER VOLUME": "TYPE (30-0)";
310 INPUT V
320 IF (V<0)+(V>30)THEN 300
330 IF FL=1 THEN 400
340 IF FL=3 THEN 1070
350 IF (V<0)+(V>30)THEN 300
360 PRINT : "ENTER NUMBER OF REPETITIONS": "TYPE (1-100)"
    ;
370 INPUT R
380 IF (R<1)+(R>100)THEN 360
390 IF R=1 THEN 430
400 PRINT : "ENTER DELAY": "TYPE (1-200)";
410 INPUT P
420 IF (P<1)+(P>200)THEN 400
430 IF FL=1 THEN 700
440 IF FL=2 THEN 910
450 IF FL=3 THEN 1110
460 IF FL=4 THEN 1530
470 CALL CLEAR
480 PRINT "SOUND( ";STR$(D);", ";STR$(F);", ";STR$(V);")":
    ;
490 PRINT "REPETITIONS=";R;:::
500 PRINT "DELAY=";P;:::::
510 IF FL=4 THEN 1580
520 FOR L=1 TO R
530 CALL SOUND(D,F,V)
540 FOR L2=1 TO P
550 NEXT L2
560 NEXT L

```

cont. on next page

Listing 3-7—cont.

```
570 GOTO 1650
580 PRINT : "ENTER STARTING FREQUENCY": "TYPE (110-7000)"
    ;
590 INPUT SF
600 IF (SF<110)+(SF>7000) THEN 580
610 PRINT : "ENTER END FREQUENCY": "TYPE (110-7000)";
620 INPUT EF
630 IF (EF<110)+(EF>7000) THEN 610
640 PRINT : "STEP UP OR DOWN": "TYPE (-100 TO 100)";
650 INPUT S
660 IF (S=0)+(S<-100)+(S>100) THEN 640
670 IF FL=3 THEN 260
680 FL=1
690 GOTO 260
700 FL=0
710 CALL CLEAR
720 PRINT "SOUND(", STR$(D); ", ", STR$(SF); ", ", STR$(V); ")"
    ::
730 PRINT "SOUND(", STR$(D); ", ", STR$(EF); ", ", STR$(V); ")"
    ::
740 PRINT "STEP="; S; ::
750 PRINT "DELAY="; P; :: :: ::
760 FOR L=SF TO EF STEP S
770 CALL SOUND(D,L,V)
780 FOR L2=1 TO P
790 NEXT L2
800 NEXT L
810 GOTO 1650
820 PRINT : "ENTER LOW FREQUENCY": "TYPE (110-7000)";
830 INPUT LF
840 IF (LF<110)+(LF>7000) THEN 820
850 PRINT : "ENTER HIGH FREQUENCY": "TYPE (110-7000)";
860 INPUT HF
870 IF (HF<110)+(HF>7000) THEN 850
880 IF FL=3 THEN 640
890 FL=2
900 GOTO 260
910 FL=0
920 CALL CLEAR
930 PRINT "SOUND(", STR$(D); ", ", STR$(LF); ", ", STR$(V); ")"
    ::
940 PRINT "SOUND(", STR$(D); ", ", STR$(HF); ", ", STR$(V); ")"
    ::
950 IF FL=3 THEN 1150
960 PRINT "REPETITIONS="; R; ::
970 PRINT "DELAY="; P; :: :: ::
980 FOR L=1 TO R
990 CALL SOUND(D,LF,V)
1000 CALL SOUND(D,HF,V)
1010 FOR L2=1 TO P
1020 NEXT L2
1030 NEXT L
1040 GOTO 1650
1050 FL=3
1060 GOTO 820
```

```

1070 PRINT : "ENTER NUMBER OF STEPS": "TYPE (1 TO 100)";
1080 INPUT R
1090 IF (R<1)+(R>100) THEN 1070
1100 GOTO 400
1110 FL=0
1120 CALL CLEAR
1130 FL=3
1140 GOTO 930
1150 FL=0
1160 PRINT "NUMBER OF STEPS="; R; ::
1170 PRINT "STEP="; S; ::
1180 PRINT "DELAY="; P; :: :: ::
1190 FOR L=1 TO R
1200 IF LF+(S*L)<131 THEN 1260
1210 IF HF+(S*L)<131 THEN 1260
1220 CALL SOUND(D, LF+(S*L), V)
1230 CALL SOUND(D, HF+(S*L), V)
1240 FOR L2=1 TO P
1250 NEXT L2
1260 NEXT L
1270 GOTO 1650
1280 PRINT : "ENTER NUMBER OF NOISE": "TYPE (-1, -2, -3)";
1290 INPUT F
1300 IF (F<-3)+(F>-1) THEN 1280
1310 GOTO 260
1320 PRINT : "ENTER NUMBER OF NOISE": "TYPE (-5, -6, -7)";
1330 INPUT F
1340 IF (F<-7)+(F>-5) THEN 1320
1350 GOTO 260
1360 PRINT : "ENTER VOICE 1 FREQUENCY": "TYPE (110 TO 700
0)";
1370 INPUT F1
1380 IF (F1<110)+(F1>7000) THEN 1360
1390 PRINT : "ENTER VOICE 2 FREQUENCY": "TYPE (110 TO 700
0)";
1400 INPUT F2
1410 IF (F2<110)+(F2>7000) THEN 1390
1420 FL=4
1430 GOTO 260
1440 FL=0
1450 PRINT : "ENTER VOICE 1 VOLUME": "TYPE (30-0)";
1460 INPUT V1
1470 IF (V1<0)+(V1>30) THEN 1450
1480 PRINT : "ENTER VOICE 2 VOLUME": "TYPE (30-0)";
1490 INPUT V2
1500 IF (V2<0)+(V2>30) THEN 1480
1510 FL=4
1520 GOTO 360
1530 FL=0
1540 CALL CLEAR
1550 PRINT "SOUND("; STR$(D); ", "; STR$(F1); ", "; STR$(V1); "
, "; STR$(F2); ", "; STR$(V2); ")"; ::
1560 FL=4
1570 GOTO 490
1580 FL=0
1590 FOR L=1 TO R
1600 CALL SOUND(D, F1, V1, F2, V2)

```

cont. on next page

Listing 3-7—cont.

```
1610 FOR L2=1 TO P
1620 NEXT L2
1630 NEXT L
1640 GOTO 1650
1650 PRINT "PRESS ANY KEY TO CONTINUE":
1660 CALL KEY(3,K,S)
1670 IF S=0 THEN 1660
1680 GOTO 150
1690 CALL CLEAR
1700 END
```

Explanation of Program

110-180	Clear screen and print sound choice menu
190-220	Get user's choice of sound type and go to proper part of the program
230-250	Routine to input a single frequency
260-280	Routine to input duration
290	If flag set, go back to another part of program
300-320	Routine to input volume
330-340	If flag set, go back
360-380	Routine to input number of repetitions
390-420	Routine to input delay between notes
430-460	If flag set, go back
470-570	Print statements and play single tones
580-630	Routine to input starting and finishing frequencies
630-660	Routine to input step between frequencies
670	If flag set, go back
680-700	Set a flag, go to another routine
710-810	Print statements and play single tones up and down
820-870	Routine to input high and low frequencies
880-910	Set a flag, go to another routine
920-1040	Print statements and play up and down sounds
1050-1060	Set a flag, go to another routine
1070-1090	Routine to input number of steps up or down
1100-1150	Go to another routine
1160-1270	Print statements and play high-low frequencies, up or down
1280-1350	Routines to input type of noise and go to another routine
1360-1410	Routines to input frequencies for multiple voices
1420-1440	Set flag, go to another routine
1450-1500	Routines to input volumes for multiple voices

1510-1530	Set flag, go to another routine
1540-1640	Print statements and play multiple voices
1650-1680	Wait for input, then go back to sound menu
1690-1700	Program exit

Important Variables in Program

F,SF,EF,LF,HF	= Frequencies of notes
D	= Duration of notes
V,V1,V2	= Volume of notes
R	= Loop variable for number of repetitions
S	= Step (increment or decrement) of sound frequencies
FL	= Flag to point to option chosen so routines can be used by other parts of program
L,L2	= Loop variables to play song
K,S	= KEY subprogram variables

Sample Run of Program

```

ENTER FREQUENCY
TYPE (110-44733)? 200

ENTER DURATION
TYPE (-4250 TO 4250)? 20

ENTER VOLUME
TYPE (30-0)? 0

ENTER NUMBER OF REPETITIONS
TYPE (1-100)? 20

ENTER DELAY
TYPE (1-200)? 20
  
```

Fig. 3-3.

Suggested Modifications and Projects

1. Allow the program even more options like 3 voices or using noises with normal frequencies.
2. Make the program keep track of the variables so that you can just type something, like "N", for no change. Then you could go through sounds and only change one aspect.

Chapter 4

Games

A computer such as the TI-99/4A, 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 computer games is included in this chapter. You might even get a game idea yourself that you could create on your own computer.

JELLY BEANS

This game uses both the graphics and sounds of the TI-99/4A to play a variation of the game High-Low. This game is self explanatory and a lot of fun for youngsters who are trying to get used to a computer.

Listing 4-1

```
100 REM  JELLY BEANS
110 CALL CLEAR
120 RANDOMIZE
130 M$="JELLYBEANS"
140 A=21
150 FOR L=1 TO 10
160 FOR L2=2 TO A
170 CALL HCHAR(11,L2,ASC(SEG$(M$,11-L,1)))
180 CALL HCHAR(11,L2-1,32)
190 NEXT L2
```

Listing 4-1—cont.

```
200 A=A-1
210 IF L<>5 THEN 230
220 A=A-1
230 NEXT L
240 FOR L=1 TO 200
250 NEXT L
260 CALL CLEAR
270 CALL SCREEN(2)
280 T=0
290 G=0
300 X$="00183C7E7E3C1800"
310 FOR L=1 TO 4
320 CALL CHAR(L*8+88,X$)
330 CALL COLOR(L+8,L*2+4,1)
340 NEXT L
350 N=INT(RND*100)+5
360 FOR L=1 TO N
370 R=INT(20*RND)+2
380 C=INT(28*RND)+3
390 CALL GCHAR(R,C,X)
400 IF (X=96)+(X=104)+(X=112)+(X=120) THEN 370
410 CALL HCHAR(R,C,INT(4*RND)*8+96)
420 CALL SOUND(5,-1,0)
430 NEXT L
440 FOR L=1 TO 300
450 NEXT L
460 CALL CLEAR
470 CALL SCREEN(3)
480 PRINT "HOW MANY JELLY BEANS DID"
490 INPUT "          YOU SEE? ":G
500 T=T+1
510 IF G=N THEN 570
520 IF G>N THEN 550
530 PRINT :::"THERE WERE MORE JELLY BEANS"::::
540 GOTO 440
550 PRINT :::"THERE WERE LESS JELLY BEANS"::::
560 GOTO 440
570 CALL CLEAR
580 T$="GUESSES"
590 IF T<>1 THEN 610
600 T$=SEG$(T$,1,5)
610 PRINT TAB(10);"YOUR RIGHT":;" IT ONLY TOOK YOU";T;
    T$:::::
620 PRINT TAB(6);"PLAY AGAIN? (Y/N)?":::
630 CALL KEY(3,K,S)
640 IF K=78 THEN 670
650 IF K=89 THEN 260
660 GOTO 630
670 CALL CLEAR
680 END
```

Explanation of Program

- 110-250 Loop to print moving title on screen
260-270 Clear screen, change background color to black

280-290	Set guess and number of guesses to zero
300-340	Loop to create jelly beans in four different character sets so they can be different colors
350	Generate random number of jelly beans
360-430	Loop to print jelly beans to screen
440-450	Short delay before screen cleared
460-470	Clear the screen, restore original background color
480-490	Input guess of number of jelly beans
500	Increment number of guesses
510	If guess correct, goto line 570
520-560	Print either too high or too low message and go back for another guess
570-610	Print win and number of guesses message
620-660	User option to play again
670-680	Program exit

Important Variables in Program

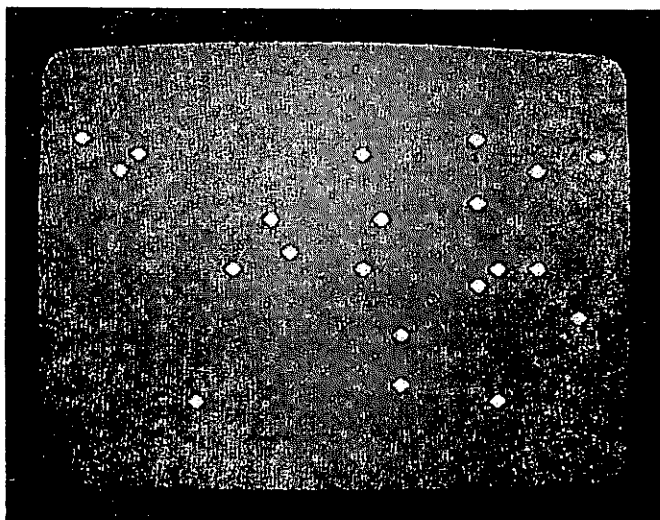
M\$	= String to print title screen
A	= Counter to print title
L,L2	= Loop variables
N	= Random number of jelly beans
T	= Number of turns or guesses
G	= Guess by user
X\$	= String variable to hold jelly bean character data
T\$	= String to print "GUESS" or "GUESSES" in win message
K,S	= KEY subprogram variables

Sample Run of Program

A sample run of the *Jelly Beans* program is shown in Fig. 4-1.

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 special graphics characters, rather than the jelly bean used.
3. Try to program a variable skill level into your program, so that the rate the jelly beans appear on the screen is slower for novice players and faster for experienced players.

**Fig. 4-1.**

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 you think will win, and see how well you fare at gambling on the spaceships.

Listing 4-2

```
100 REM  ROCKET RACE
110 CALL CLEAR
120 RANDOMIZE
130 PRINT TAB(4);"WELCOME TO ROCKET RACE"::::
140 PRINT "YOU WILL GET TO PLACE A":"BET ON THE ROCKET
    THAT YOU":"THINK WILL WIN THE RACE"::::
150 PRINT "IF YOU PICK THE WINNER":"THEN YOU WILL WIN 3
    TIMES":"YOUR BET"::::
160 PRINT "    PRESS ANY KEY TO START"::::
170 CALL KEY(3,K,S)
180 IF S=0 THEN 170
190 M=100
200 CALL CLEAR
210 PRINT "WHICH ROCKET DO YOU BET":"WILL FINISH FIRST?
    (A,B,C)"::::
220 CALL KEY(3,K,S)
230 IF (S=0)+(K<65)+(K>67) THEN 220
240 F=K-64
250 CALL CLEAR
260 PRINT "YOU HAVE";M;"DOLLARS"::::
```

```
270 PRINT "WHAT IS YOUR BET"
280 PRINT : "ENTER ( 1 -";M;")";
290 INPUT "? " : B
300 IF (B<1)+(B>M) THEN 280
310 B=INT(B)
320 CALL CLEAR
330 CALL CHAR(128,"101038383838386C")
340 PRINT TAB(9); "A      B      C"
350 C1=21
360 C2=21
370 C3=21
380 C1=C1-INT(2*RND)
390 C2=C2-INT(2*RND)
400 C3=C3-INT(2*RND)
410 CALL HCHAR(C1+1,11,32)
420 CALL HCHAR(C2+1,16,32)
430 CALL HCHAR(C3+1,21,32)
440 CALL HCHAR(C1,11,128)
450 CALL HCHAR(C2,16,128)
460 CALL HCHAR(C3,21,128)
470 IF (C1=1)+(C2=1)+(C3=1) THEN 490
480 GOTO 380
490 CALL CLEAR
500 ON INT(RND*3)+1 GOTO 510,540,570
510 IF C1<>1 THEN 500
520 W=1
530 GOTO 590
540 IF C2<>1 THEN 500
550 W=2
560 GOTO 590
570 IF C3<>1 THEN 500
580 W=3
590 IF W=F THEN 640
600 PRINT TAB(10);CHR$(W+64); " WINS":TAB(8);"YOU CHOSE
    ";CHR$(F+64);:TAB(9);"YOU LOSE"::::
610 M=M-B
620 IF M<=0 THEN 720
630 GOTO 660
640 PRINT TAB(10);CHR$(W+64); " WINS":TAB(8);"YOU CHOSE
    ";CHR$(F+64);:TAB(9);"YOU WIN"::::
650 M=M+(3*B)
660 PRINT " YOU NOW HAVE";M;"DOLLARS"::
670 PRINT TAB(7);"BET AGAIN? (Y/N)"::
680 CALL KEY(3,K,S)
690 IF K=78 THEN 780
700 IF K=89 THEN 200
710 GOTO 680
720 PRINT TAB(7);"YOU'RE BROKE"::
730 PRINT TAB(5);"ANOTHER GAME? (Y/N)"::
740 CALL KEY(3,K,S)
750 IF K=78 THEN 780
760 IF K=89 THEN 190
770 GOTO 740
780 CALL CLEAR
790 END
```

Explanation of Program

110-180	Clear screen, print instructions, wait for input to start the game
190	Set initial amount of betting money
200-240	Input user's choice of winning rocket
250-310	Input user's bet
320-480	Move the rockets up the screen using the HCHAR subprogram until one rocket reaches the top of the screen
490-580	Randomly check to see which rocket is the winner until a winner is found (randomly in case two rockets get to top at the same time)
590	Check to see if user chose the winning rocket
600-630	If not, print losing message, subtract the bet, and go to line 720 or line 660 depending on how much money they have left
640-650	Print winning message and add the winnings to the money
660	Print amount of money message
670-710	User option to bet on another race
720	Print your broke message
730-770	User option to play another game
780-790	Program exit

Important Variables in Program

M	= Amount of money
F	= User's choice of first place rocket
B	= Amount of bet
C1,C2,C3	= Locations of rockets A, B, and C
W	= Winning rocket number
K,S	= KEY subprogram variables

Sample Run of Program

A sample run of the *Rocket Race* program is shown in Fig.4-2.

Suggested Modifications and Projects

1. Put a fanfare at the beginning and end of the race, just like a real race would have.
2. Try to program the special characters for different shapes, maybe balloons or animated birds.
3. See if you can program the keyboard so that two players can press certain keys rapidly to make the rockets move. In this way the rock-

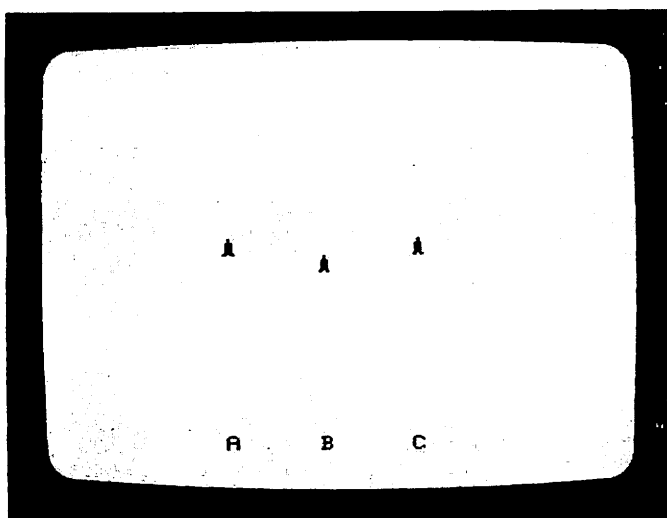


Fig. 4-2.

ets would not move randomly, but would be controlled by competing players.

DICE GAMES

Chuck-a-luck and high-low are the two popular dice games contained in this program. In chuck-a-luck, three die will be "thrown" and you will try to guess what number will appear most often. In high-low, two die will be "thrown" and you will try to guess if the total will be less than seven, equal to seven, or higher than seven.

Listing 4-3

```

100 REM DICE GAMES
110 CALL CLEAR
120 RANDOMIZE
130 CALL CHAR(128,"FEFEFEFEFEFEFE00")
140 CALL CHAR(129,"FEFEDEFEF6FEFE00")
150 CALL CHAR(130,"FEBEFEEFEFAFE00")
160 CALL CHAR(131,"FEBAFEFEFEBAFE00")
170 CALL CHAR(132,"FEBAFEEFEFEBAFE00")
180 CALL CHAR(133,"FEAAFEFEFEAAFE00")
190 CALL CHAR(134,"183C7EFFFF7E3C18")
200 CALL CLEAR
210 M=100
220 PRINT TAB(4);"WELCOME TO DICE GAMES"::::

```

cont. on next page

Listing 4-3—cont.

```
230 PRINT " YOU HAVE";M;"DOLLARS":,:
240 IF M<=0 THEN 1230
250 PRINT TAB(6);"1. CHUCK-A-LUCK":TAB(6);"2. HIGH-LOW
   ":TAB(6);"3. INSTRUCTIONS":TAB(6);"4. EXIT PROGRA
   M":,:
260 PRINT TAB(3);"YOUR CHOICE? (1,2,3,4)":,:
270 CALL KEY(3,K,S)
280 IF K=49 THEN 460
290 IF K=50 THEN 670
300 IF K=51 THEN 330
310 IF K=52 THEN 1290
320 GOTO 270
330 CALL CLEAR
340 PRINT TAB(8);"CHUCK-A-LUCK":,::"IN THIS GAME YOU GE
   T TO"::"CHOOSE A DIE VALUE FROM"::"1 TO 6":,:
350 PRINT "THREE DIE WILL THEN BE"::"ROLLED -- IF YOUR D
   IE"::"MATCHES A ROLLED DIE"
360 PRINT "THEN YOU WILL WIN THE"::"AMOUNT YOU BET":,:
370 GOSUB 910
380 CALL CLEAR
390 PRINT TAB(10);"HIGH-LOW":,::"IN THIS GAME TWO DIE WI
   LL"::"BE ROLLED":,:
400 PRINT "YOU TRY TO GUESS IF THE"::"ROLL WILL BE LESS
   THAN 7,"::"GREATER THAN 7, OR EQUAL"::"TO 7":,:
410 PRINT "IF YOU GUESS HIGH OR LOW"::"CORRECT THEN YOU
   WIN YOUR"::"BET, BUT IF YOU GUESS EQUAL"
420 PRINT "CORRECT THEN YOU WIN"::"2 TIMES YOUR BET":,:
430 GOSUB 910
440 CALL CLEAR
450 GOTO 230
460 CALL CLEAR
470 PRINT "WHICH DIE DO YOU WISH TO"
480 PRINT "BET ON? (1-6)":,:,:,:,:
490 CALL KEY(3,K,S)
500 IF (S=0)+(K<49)+(K>54)THEN 490
510 C=K-48
520 GOSUB 950
530 N=3
540 GOSUB 1020
550 IF D1<>C THEN 570
560 M=M+B
570 IF D2<>C THEN 590
580 M=M+B
590 IF D3<>C THEN 610
600 M=M+B
610 IF (D1=C)+(D2=C)+(D3=C)THEN 630
620 M=M-B
630 PRINT " YOU NOW HAVE";M;"DOLLARS":,:
640 GOSUB 910
650 CALL CLEAR
660 GOTO 230
670 CALL CLEAR
680 PRINT TAB(7);"1. LOWER THAN 7":TAB(7);"2. EQUAL TO
   7":TAB(7);"3. GREATER THAN 7":,:
690 PRINT TAB(6);"YOUR CHOICE? (1,2,3)":,:,::
```

```

700 CALL KEY(3,K,S)
710 IF (S=0)+(K<49)+(K>54) THEN 700
720 C=K-48
730 GOSUB 950
740 N=2
750 GOSUB 1020
760 T=D1+D2
770 IF C<>1 THEN 810
780 IF T>6 THEN 890
790 M=M+B
800 GOTO 900
810 IF C<>2 THEN 850
820 IF T<>7 THEN 890
830 M=M+(2*B)
840 GOTO 900
850 IF C<>3 THEN 890
860 IF T<8 THEN 890
870 M=M+B
880 GOTO 630
890 M=M-B
900 GOTO 630
910 PRINT " PRESS ANY KEY TO CONTINUE":::
920 CALL KEY(3,K,S)
930 IF S=0 THEN 920
940 RETURN
950 CALL CLEAR
960 PRINT "HOW MUCH DO YOU WISH TO BET?"
970 PRINT : "ENTER ( 1 -";M;")";
980 INPUT B
990 IF (B<1)+(B>M) THEN 970
1000 B=INT(B)
1010 RETURN
1020 CALL CLEAR
1030 PRINT TAB(7); "HERE IS THE ROLL":::
1040 D1=INT(6*RND)+1
1050 D2=INT(6*RND)+1
1060 D3=INT(6*RND)+1
1070 FOR L=1 TO N
1080 FOR L2=3 TO 17+N-(L*2)
1090 CALL HCHAR(15,L2-1,32)
1100 CALL HCHAR(15,L2,134)
1110 CALL HCHAR(15,L2,INT(RND*6)+128)
1120 CALL SOUND(2,-5,0)
1130 NEXT L2
1140 IF L<>1 THEN 1160
1150 D=D1
1160 IF L<>2 THEN 1180
1170 D=D2
1180 IF L<>3 THEN 1200
1190 D=D3
1200 CALL HCHAR(15,17+N-(L*2),D+127)
1210 NEXT L
1220 RETURN
1230 PRINT TAB(8); "YOU'RE BROKE":::
1240 PRINT TAB(5); "PLAY AGAIN? (Y/N)":::
1250 CALL KEY(3,K,S)
1260 IF K=89 THEN 200

```

cont. on next page

Listing 4-3—cont.

```
1270 IF K=78 THEN 1290
1280 GOTO 1250
1290 CALL CLEAR
1300 END
```

Explanation of Program

130-200	Creation of dice special characters
210	Set initial betting money
220-260	Print dice games choice menu
270-320	Get user's choice input and go to the proper part of the program
330-450	Print game instructions, call wait subroutine between pages and before returning to choice menu
460-510	Input user's die choice for chuck-a-luck
520	Go to bet input subroutine
530-540	Set number of die to be thrown and go to die rolling subroutine
550-600	Check for matches to the user's choice, and if so, then add winnings to amount of money
610-620	If no match, then subtract bet from money
630-660	Print new amount of money, go to wait subroutine, and return to main choice menu
670-720	Print choice menu for high-low and input user's choice
730	Go to bet input subroutine
740-750	Set number of die to be thrown and go to die rolling subroutine
760	Compute die total
770-880	Check by menu choice number to see if user correct, if so add winnings to bet, go back to line 630 and print new amount of money, go to wait subroutine, and finally return to main choice menu
890-900	Since user wrong, subtract bet from money, and go to line 630
910-940	Wait subroutine—wait for input before continuing the program
950-1010	Bet subroutine—input user's bet
1020-1220	Die rolling subroutine—generates random die numbers, then uses the HCHAR subprogram and the special die characters to simulate die rolling across the screen. Random

die value is printed to screen using variable D, whose value is figured in lines 1140-1190

1230-1280 Print your broke message and user option to re-play the game

1290-1300 Program exit

Important Variables in Program

M = Amount of money
 C = Number bet on in chuck-a-luck or choice in high-low game
 N = Number of die to roll
 T = Total of die in high-low
 B = Amount of bet
 D1,D2,D3 = Random values of die
 L,L2 = Loop variables for die rolling subroutine
 K,S = KEY subprogram variables

Sample Run of Program

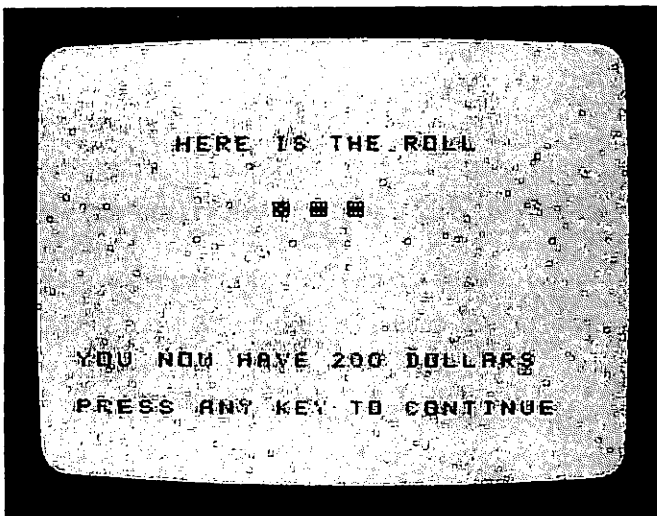


Fig. 4-3.

Suggested Modifications and Projects

1. Add other dice games that utilize the subroutines of this program.
2. Add sounds or songs for winning, losing, and other parts of the program.

NO-FRILLS DICE

No-Frills Dice is a basic version of *Dice Games*. It does not include any of the graphics features so it is slightly shorter and easier to type into the computer. It runs faster because printing normal characters is faster than printing special graphics characters.

Listing 4-4

```
100 REM NO-FRILLS DICE
110 CALL CLEAR
120 RANDOMIZE
130 M=100
140 PRINT TAB(4);"WELCOME TO DICE GAMES"::::
150 PRINT " YOU HAVE";M;"DOLLARS"::::
160 IF M<=0 THEN 1040
170 PRINT TAB(6);"1. CHUCK-A-LUCK":TAB(6);"2. HIGH-LOW
   ":TAB(6);"3. INSTRUCTIONS":TAB(6);"4. EXIT PROGRA
   M"::::
180 PRINT TAB(3);"YOUR CHOICE? (1,2,3,4)"::::
190 CALL KEY(3,K,S)
200 IF K=49 THEN 380
210 IF K=50 THEN 590
220 IF K=51 THEN 250
230 IF K=52 THEN 1100
240 GOTO 190
250 CALL CLEAR
260 PRINT TAB(8);"CHUCK-A-LUCK"::::"IN THIS GAME YOU GE
   T TO":"CHOOSE A DIE VALUE FROM":"1 TO 6"::::
270 PRINT "THREE DIE WILL THEN BE":"ROLLED -- IF YOUR D
   IE":"MATCHES A ROLLED DIE"
280 PRINT "THEN YOU WILL WIN THE":"AMOUNT YOU BET"::::
290 GOSUB 830
300 CALL CLEAR
310 PRINT TAB(10);"HIGH-LOW"::::"IN THIS GAME TWO DIE WI
   LL":"BE ROLLED"::::
320 PRINT "YOU TRY TO GUESS IF THE":"ROLL WILL BE LESS
   THAN 7,":"GREATER THAN 7, OR EQUAL":"TO 7"::::
330 PRINT "IF YOU GUESS HIGH OR LOW":"CORRECT THEN YOU
   WIN YOUR":"BET, BUT IF YOU GUESS EQUAL"
340 PRINT "CORRECT THEN YOU WIN":"2 TIMES YOUR BET"::::
350 GOSUB 830
360 CALL CLEAR
370 GOTO 150
380 CALL CLEAR
390 PRINT "WHICH DIE DO YOU WISH TO"
400 PRINT "BET ON? (1-6)"::::
410 CALL KEY(3,K,S)
420 IF (S=0)+(K<49)+(K>54) THEN 410
430 C=K-48
440 GOSUB 870
450 N=3
460 GOSUB 940
470 IF D1<>C THEN 490
480 M=M+B
```

```
490 IF D2<>C THEN 510
500 M=M+B
510 IF D3<>C THEN 530
520 M=M+B
530 IF (D1=C)+(D2=C)+(D3=C)THEN 550
540 M=M-B
550 PRINT " YOU NOW HAVE";M;"DOLLARS":::
560 GOSUB 830
570 CALL CLEAR
580 GOTO 150
590 CALL CLEAR
600 PRINT TAB(7);"1. LOWER THAN 7":TAB(7);"2. EQUAL TO
    7":TAB(7);"3. GREATER THAN 7"::::
610 PRINT TAB(6);"YOUR CHOICE? (1,2,3)"::::
620 CALL KEY(3,K,S)
630 IF (S=0)+(K<49)+(K>54)THEN 620
640 C=K-48
650 GOSUB 870
660 N=2
670 GOSUB 940
680 T=D1+D2
690 IF C<>1 THEN 730
700 IF T>6 THEN 810
710 M=M+B
720 GOTO 820
730 IF C<>2 THEN 770
740 IF T<>7 THEN 810
750 M=M+(2*B)
760 GOTO 820
770 IF C<>3 THEN 810
780 IF T<8 THEN 810
790 M=M+B
800 GOTO 550
810 M=M-B
820 GOTO 550
830 PRINT " PRESS ANY KEY TO CONTINUE":::
840 CALL KEY(3,K,S)
850 IF S=0 THEN 840
860 RETURN
870 CALL CLEAR
880 PRINT "HOW MUCH DO YOU WISH TO BET?"
890 PRINT ":"ENTER ( 1 -";M;")";
900 INPUT B
910 IF (B<1)+(B>M)THEN 890
920 B=INT(B)
930 RETURN
940 CALL CLEAR
950 PRINT TAB(7);"HERE IS THE ROLL"::::
960 D1=INT(6*RND)+1
970 D2=INT(6*RND)+1
980 D3=INT(6*RND)+1
990 IF N=2 THEN 1020
1000 PRINT TAB(10);D1;D2;D3:~::~:
1010 GOTO 1030
1020 PRINT TAB(12);D1;D2:~::~:
1030 RETURN
1040 PRINT TAB(8);"YOU'RE BROKE":~::~:
```

cont. on next page

Listing 4-4—cont.

```
1050 PRINT TAB(5);"PLAY AGAIN? (Y/N)"::::
1060 CALL KEY(3,K,S)
1070 IF K=89 THEN 110
1080 IF K=78 THEN 1100
1090 GOTO 1060
1100 CALL CLEAR
1110 END
```

Explanation of Program

110-180	Set initial betting money and print dice games choice menu
190-240	Get user's choice input and go to the proper part of the program
250-370	Print game instructions, call wait subroutine between pages and before returning to choice menu
380-430	Input user's die choice for chuck-a-luck
440	Go to bet input subroutine
450-460	Set number of die to be thrown and go to die rolling subroutine
470-520	Check for matches to the user's choice, and if so, then add winnings to amount of money
530-540	If no match, then subtract bet from money
550-580	Print new amount of money, go to wait subroutine, and return to main choice menu
590-640	Print choice menu for high-low and input user's choice
650	Go to bet input subroutine
660-670	Set number of die to be thrown and go to die rolling subroutine
680	Compute die total
690-800	Check by menu choice number to see if user correct, if so add winnings to bet, go back to line 550 and print new amount of money, go to wait subroutine, and finally return to main choice menu
810-820	Since user wrong, subtract bet from money, and go to line 550
830-860	Wait subroutine—wait for input before continuing the program
870-930	Bet subroutine—input user's bet
940-1030	Die rolling subroutine—generates random die numbers and prints them to the screen

1040-1090	Print your broke message and user option to re-play the game
1100-1110	Program exit

Important Variables in Program

M	= Amount of money
C	= Number bet on in chuck-a-luck or choice in high-low game
N	= Number of die to roll
T	= Total of die in high-low
B	= Amount of bet
D1,D2,D3	= Random values of die
K,S	= KEY subprogram variables

Sample Run of Program

```

WHICH DIE DO YOU WISH TO
BET ON? (1-6)? 6

HOW MUCH DO YOU WISH TO BET?
TYPE ( 1 - 100 )? 50
  HERE IS THE ROLL

      1 2 6

YOU NOW HAVE 150 DOLLARS
PRESS ANY KEY TO CONTINUE

```

Fig. 4-4.

Suggested Modifications and Projects

1. Add other dice games to this program that utilize the subroutines of this program.
2. Add sounds or songs for winning, losing, and other parts of the program.

SIMON SAYS

The colors and sounds of this game test your memory. Simply practice the colors a few times, tell the computer how many different colors you think you can remember, and enjoy the game.

Listing 4-5

```
100 REM SIMON SAYS
110 CALL CLEAR
120 RANDOMIZE
130 C$(1)="BLUE"
140 C$(2)="RED"
150 C$(3)="YELLOW"
160 C$(4)="GRAY"
170 PRINT TAB(4);"WELCOME TO SIMON SAYS":::
180 PRINT "YOU WILL TRY TO REPEAT":"THE COLORS THAT ARE
    SHOWN":"ON THE SCREEN":::
190 PRINT "YOU MAY GET ACQUAINTED":"WITH THE COLORS (1-
    4) AND":"SOUNDS AS MUCH AS YOU LIKE":::
200 FOR L=1 TO 4
210 PRINT TAB(7);L;"- ";C$(L)
220 NEXT L
230 PRINT
240 PRINT "PRESS A 1 TO 4 TO SEE A":"COLOR OR PRESS A 5
    TO":"START THE GAME":::
250 PRINT "PRESS YOUR CHOICE? (1-5)"
260 CALL KEY(3,K,S)
270 IF (S=0)+(K<49)+(K>53)THEN 260
280 CALL CLEAR
290 ON K-48 GOTO 300,330,360,390,420
300 CALL SCREEN(6)
310 CALL SOUND(200,200,0)
320 GOTO 200
330 CALL SCREEN(9)
340 CALL SOUND(150,400,0)
350 GOTO 200
360 CALL SCREEN(12)
370 CALL SOUND(150,800,0)
380 GOTO 200
390 CALL SCREEN(15)
400 CALL SOUND(150,1600,0)
410 GOTO 200
420 CALL CLEAR
430 CALL SCREEN(4)
440 PRINT " HOW MANY EVENTS (1-8)":" DO YOU WISH TO A
    TTEMPT":" PLACING IN ORDER?":::
450 PRINT " YOUR CHOICE? (1-8)"::::
460 CALL KEY(3,K,S)
470 IF (S=0)+(K<49)+(K>56)THEN 460
480 N=K-48
490 PRINT " PRESS ANY KEY TO START":::
500 CALL KEY(3,K,S)
510 IF S=0 THEN 500
520 FOR L=1 TO N
530 CALL CLEAR
540 C(L)=INT(RND*4)+1
550 CALL SCREEN((C(L)-1)*3+6)
560 CALL SOUND(150,(2^C(L))*100,0)
570 FOR L2=1 TO 200
580 NEXT L2
590 NEXT L
600 CALL SCREEN(4)
```

```

610 FOR L=1 TO N
620 CALL CLEAR
630 PRINT TAB(9);"1. BLUE":TAB(9);"2. RED":TAB(9);"3. Y
    ELLOW":TAB(9);"4. GRAY"::::
640 PRINT " WHAT WAS THE COLOR OF ":" SCREEN #";L;"? (
    1-4)"::::
650 CALL KEY(3,K,S)
660 IF (S=0)+(K<49)+(K>52)THEN 650
670 IF C(L)=K-48 THEN 710
680 CALL CLEAR
690 PRINT TAB(4);"WRONG ---- NUMBER";L::TAB(9);"WAS ";C
    $(C(L))::::
700 GOTO 740
710 NEXT L
720 CALL CLEAR
730 PRINT TAB(7);"CONGRATULATIONS": " YOU GOT THEM ALL
    RIGHT"::::
740 PRINT TAB(6);"PLAY AGAIN? (Y/N)"::::
750 CALL KEY(3,K,S)
760 IF K=78 THEN 790
770 IF K=89 THEN 420
780 GOTO 750
790 CALL CLEAR
800 END

```

Explanation of Program

110-160	Clear screen and load color names into array C\$(4)
170-250	Print instructions and color names to screen
260-290	User option to see colors or start game
300-410	Four short routines to show different colors and play different sounds so the user can get acquainted with them
420-480	Input number of events user wishes to attempt
490-510	Wait for input to start showing events
520-590	Loop to choose random colors for each event and to print each of them to screen
600-610	Start of loop for user to recall colors
620-670	Print color choices to screen, input user's guess for each event
680-700	If an error is made, then print error message and go to line number 740
710	Complete recall loop
720-730	Print all-correct message
740-780	User option to play another game
790-800	Program exit

Important Variables in Program

C\$(4)	= Array to hold color names
L,L2	= Loop variables

N = Number of events to attempt
C(8) = Array to hold color numbers for each event
K,S = KEY subprogram variables

Sample Run of Program

**HOW MANY EVENTS (1-8)
DO YOU WISH TO ATTEMPT
PLACING IN ORDER?**

**YOUR CHOICE? (1-8)
PRESS ANY KEY TO START**

- 1. BLUE**
- 2. RED**
- 3. YELLOW**
- 4. GRAY**

**WHAT WAS THE COLOR OF
SCREEN # 1?**

Fig. 4-5.

Suggested Modifications and Projects

1. Try dividing portions of the screen into four small blocks, one for each color, so that the whole screen is not filled with each color for the game play.
2. Use skill levels to vary the speed at which colors and sounds are shown to the player.

BLACKJACK

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

Listing 4-6

```
100 REM BLACKJACK
110 CALL CLEAR
120 RANDOMIZE
130 PRINT TAB(5);"WELCOME TO BLACKJACK":::
140 PRINT "THIS IS A STANDARD BLACKJACK":"CARD GAME":::
    THE DEALER MUST STAND ON
17": "AND ABOVE - HE MUST TAKE"
150 PRINT "ON 16 AND BELOW": "THE DEALER WINS ALL TIES"
    :::
160 PRINT TAB(8);"PLEASE WAIT....": "THE CARDS ARE BEI
    NG SHUFFLED"::::
```

```

170 DIM D$(52),V(52)
180 DATA ACE,11,KING,10,QUEEN,10,JACK,10,TEN,10,NINE,9,
    EIGHT,8,SEVEN,7,SIX,6,FIVE,5,FOUR,4,THREE,3,TWO,2
190 FOR L=1 TO 13
200 READ X$,X
210 D$(L)=X$&" OF SPADES"
220 D$(L+13)=X$&" OF CLUBS"
230 D$(L+26)=X$&" OF DIAMONDS"
240 D$(L+39)=X$&" OF HEARTS"
250 V(L)=X
260 V(L+13)=X
270 V(L+26)=X
280 V(L+39)=X
290 NEXT L
300 M=100
310 FOR L=1 TO 52
320 X$=D$(L)
330 X=V(L)
340 R=INT(RND*52)+1
350 D$(L)=D$(R)
360 V(L)=V(R)
370 D$(R)=X$
380 V(R)=X
390 NEXT L
400 C=0
410 CALL CLEAR
420 PRINT "WHAT IS YOUR BET?"
430 PRINT : "ENTER ( 1 -";M;") ";
440 INPUT B
450 IF (B<1)+(B>M)THEN 430
460 B=INT(B)
470 CALL CLEAR
480 PRINT "DEALERS HAND":::::::::"YOUR HAND":::::::::
    "YOU HAVE";M;"DOLLARS":::
490 GOSUB 860
500 GOSUB 960
510 GOSUB 960
520 IF PH<>21 THEN 580
530 P$="BLACKJACK !!!!! "
540 F=1
550 R=23
560 GOSUB 1060
570 GOTO 1200
580 P$="ANOTHER CARD? (Y/N)"
590 R=23
600 GOSUB 1060
610 CALL KEY(3,K,S)
620 IF K=78 THEN 790
630 IF K=89 THEN 650
640 GOTO 610
650 P$=" "
660 R=23
670 GOSUB 1060
680 GOSUB 960
690 IF PH>21 THEN 710
700 GOTO 580
710 IF PA=0 THEN 750

```

cont. on next page

Listing 4-6—cont.

```
720 PA=PA-1
730 PH=PH-10
740 GOTO 690
750 P$="YOU BUSTED ! ! ! !      "
760 R=23
770 GOSUB 1060
780 GOTO 1110
790 GOSUB 860
800 IF DH<17 THEN 790
810 IF (DH<22)+(DA=0) THEN 850
820 DA=DA-1
830 DH=DH-10
840 GOTO 800
850 GOTO 1100
860 C=C+1
870 DC=DC+1
880 DH$(DC)=D$(C)
890 DH=DH+V(C)
900 IF V(C)<>11 THEN 920
910 DA=DA+1
920 P$=DH$(DC)
930 R=2+DC
940 GOSUB 1060
950 RETURN
960 C=C+1
970 PC=PC+1
980 PH$(PC)=D$(C)
990 PH=PH+V(C)
1000 IF V(C)<>11 THEN 1020
1010 PA=PA+1
1020 P$=PH$(PC)
1030 R=12+PC
1040 GOSUB 1060
1050 RETURN
1060 FOR L2=1 TO LEN(P$)
1070 CALL HCHAR(R,L2+2,ASC(SEG$(P$,L2,1)))
1080 NEXT L2
1090 RETURN
1100 IF (PH>DH)+(DH>21) THEN 1200
1110 P$="DEALER WINS      "
1120 R=23
1130 GOSUB 1060
1140 M=M-B
1150 P$="YOU HAVE "&STR$(M)&" DOLLARS      "
1160 R=21
1170 GOSUB 1060
1180 IF M<=0 THEN 1460
1190 GOTO 1270
1200 P$="YOU WIN ! ! ! ! !      "
1210 R=23
1220 GOSUB 1060
1230 M=M+B
1240 IF F=0 THEN 1260
1250 M=M+INT(B/2)
1260 GOTO 1150
```

```

1270 P$="PLAY AGAIN? (Y/N)  "
1280 R=23
1290 GOSUB 1060
1300 CALL KEY(3,K,S)
1310 IF K=78 THEN 1580
1320 IF K=89 THEN 1340
1330 GOTO 1300
1340 DH=0
1350 PH=0
1360 DC=0
1370 PC=0
1380 DA=0
1390 PA=0
1400 F=0
1410 IF C<42 THEN 410
1420 P$="RE-SHUFFLING CARDS  "
1430 R=23
1440 GOSUB 1060
1450 GOTO 310
1460 P$="YOU'RE BROKE !!!  "
1470 R=21
1480 GOSUB 1060
1490 P$="ANOTHER GAME? (Y/N)"
1500 R=23
1510 GOSUB 1060
1520 CALL KEY(3,K,S)
1530 IF K=78 THEN 1580
1540 IF K=89 THEN 1560
1550 GOTO 1520
1560 M=100
1570 GOTO 1340
1580 CALL CLEAR
1590 END

```

Explanation of Program

110-160	Clear screen and print instructions
170	Dimension arrays to hold card names and their values
180	Data of card names and values
190-290	Loop to read in and assign card names and values to the arrays
300	Set initial betting money
310-390	Routine to randomly shuffle cards
400	Set cards-dealt counter to zero
410-460	Input user's bet
470-480	Print hand titles and amount of money to screen
490	Go to dealer-gets-card subroutine
500-510	Go to player-gets-card subroutine twice
520-570	If player has blackjack, then print message and go to player-win routine
580-640	User option to get another card

650-740	Player gets another card, check to see if it is an ace to increment ace counter—check to see if player “busts”, and if not, go back and ask if another card is wanted
750-780	Print player-busted message, go to dealer-wins routine
790-850	Dealer gets cards until he has 17 or above, if he busts, then go to player-wins routine
860-950	Subroutine to give dealer another card
950-1050	Subroutine to give player another card
1060-1090	Subroutine to print messages and card names to screen
1100	If player has better hand then go to player-wins routine
1110-1190	Dealer wins routine—print dealer-wins message, subtract bet from money, print new amount of money, check to see if player out of money; if not, goes to play again option
1200-1260	Player-wins routine—print player-wins message, add winnings to money, go to print new amount part of the dealer-wins routine
1270-1330	User option to play another hand
1340-1400	If play another, then reset all hand variables
1410-1450	Check to see if cards need to be re-shuffled, if so print message and go to line 310, if not then go to line 410
1460-1480	Print your broke message
1490-1550	User option to play another game
1560-1570	If play another game, then reset betting money and go to line 1340 to reset the hand variables
1580-1590	Program exit

Important Variables in Program

D\$(52)	= Array to hold card names
V(52)	= Array to hold card values
L,L2	= Loop variables
X\$,X and V(52)	= Temporary variables used to fill in D\$(52)
M	= Amount of betting money
R	= Random number for shuffling cards
C	= Number of cards played from deck
B	= Player's bet
F	= Flag set if player gets blackjack
PH,DH	= Value of player's and dealer's hands
PA,DA	= Ace counts of player's and dealer's hands
PC,DC	= Number of cards in player's and dealer's hands
PH\$(10),DH\$(10)	= Card names in order for hands

P\$ = Message string for print subroutine
R = Row variable for print subroutine
K,S = KEY subprogram variables

Sample Run of Program

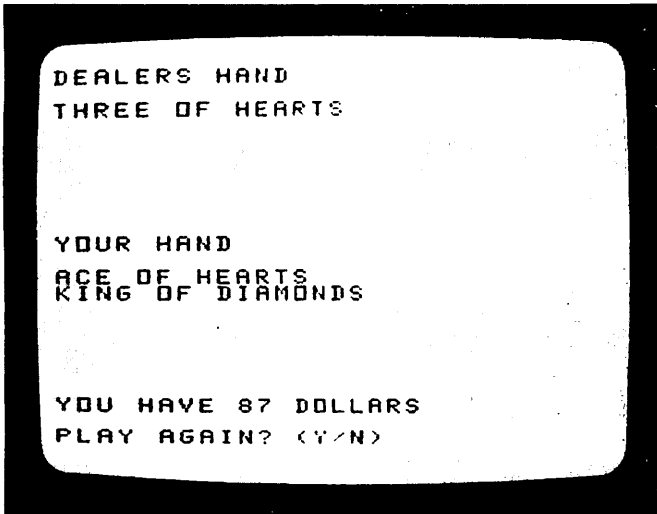


Fig. 4-6.

Suggested Modifications and Projects

1. See if you can make graphics work with this program so that cards are actually printed on the screen.
2. Try to program other favorite card games into your TI 99/4A using the routines 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 is user-friendly, it is praised for being an excellent teaching device. Computers can even teach people about computers. In this chapter there are seven programs for the TI-99/4A that teach subjects from geometry to French food names. There are two programs to help extend your knowledge of the BASIC language.

HISTORY

This program not only teaches you about history, but it also gives quizzes. This is accomplished by quizzing a person about certain historical events and the order in which they occurred.

Listing 5-1

```
100 REM HISTORY
110 CALL CLEAR
120 RANDOMIZE
130 PRINT TAB(5);"WELCOME TO HISTORY":::::
140 PRINT " IN THIS LESSON YOU WILL":" ATTEMPT TO PLA
    CE EVENTS":" IN CHRONOLOGICAL ORDER":::::
150 PRINT " PRESS ANY KEY TO START":::::
```

cont. on next page

Listing 5-1—cont.

```
160 CALL KEY(3,K,S)
170 N=0
180 W=0
190 RESTORE
200 IF S=0 THEN 160
210 CALL CLEAR
220 READ E$(1),E$(2),E$(3),T$(1),T$(2),T$(3)
230 N=N+2
240 E1=INT(3*RND)+1
250 E2=INT(3*RND)+1
260 IF E2=E1 THEN 250
270 E3=INT(3*RND)+1
280 IF (E3=E2)+(E3=E1) THEN 270
290 PRINT "1. ";E$(E1);;"2. ";E$(E2);;"3. ";E$(E3);;
:
300 PRINT "WHICH EVENT OCCURRED FIRST":"PRESS CHOICE? (
1,2,3)";:
310 CALL KEY(3,K,S)
320 IF (S=0)+(K<49)+(K>51) THEN 310
330 C=K-48
340 IF ((C=1)*(E1=1))+((C=2)*(E2=1))+((C=3)*(E3=1)) THEN
380
350 PRINT "WRONG":E$(1):"OCCURRED FIRST":;:
360 W=W+1
370 GOTO 390
380 PRINT "RIGHT":E$(1):"OCCURRED FIRST":;:
390 PRINT "WHICH EVENT OCCURRED SECOND?":"PRESS CHOICE?
(1,2,3)";:
400 CALL KEY(3,K,S)
410 IF (S=0)+(K<49)+(K>51) THEN 400
420 C=K-48
430 IF ((C=1)*(E1=2))+((C=2)*(E2=2))+((C=3)*(E3=2)) THEN
470
440 PRINT "WRONG":E$(2):"OCCURRED SECOND":;:
450 W=W+1
460 GOTO 480
470 PRINT "RIGHT":E$(2):"OCCURRED SECOND":;:
480 FOR D=1 TO 700
490 NEXT D
500 CALL CLEAR
510 PRINT "THE CORRECT ORDER IS":;:
520 FOR L=1 TO 3
530 PRINT E$(L):T$(L):;
540 NEXT L
550 PRINT :;"YOUR SCORE IS";(N-W);"OUT OF";N;:;:
560 IF N=40 THEN 820
570 PRINT "ATTEMPT ANOTHER? (Y/N)":;:
580 CALL KEY(3,K,S)
590 IF K=89 THEN 210
600 IF K=78 THEN 880
610 GOTO 580
620 DATA FRENCH AND INDIAN WAR,AMERICAN REVOLUTION,WAR
OF 1812,1755,1776,1812
630 DATA CIVIL WAR,EMANCIPATION PROCLAMATION,FOURTEENTH
AMENDMENT,1861,1863,1866
```

```

640 DATA LOUISIANA PURCHASE,WAR OF 1812,JACKSON ADMINIS
    TRATION,1803,1812,1820
650 DATA MEXICAN WAR,SPANISH-AMERICAN WAR,WORLD WAR I,1
    846,1898,1914
660 DATA INDEPENDENCE DECLARED,ARTICLES OF CONFEDERATION
    ,CONSTITUTION,1776,1781,1787
670 DATA BATTLE OF YORKTOWN,BATTLE OF THE MARNE,BATTLE
    OF THE BULGE,1781,1914,1944
680 DATA TREATY OF GHENT,WORLD WAR I,TREATY OF VERSAILLES,
    1814,1914,1919
690 DATA BOSTON TEA PARTY,INTOLERABLE ACTS,INDEPENDENCE
    DECLARED,1773,1774,1776
700 DATA CARTHAGE FOUNDED,FIRST OLYMPIC GAMES,ROME FOUNDED,
    800 BC,776 BC,753 BC
710 DATA BRITAIN TAKEN BY ROME,DEATH OF JULIUS CAESAR,JESUS
    BORN,84 BC,44 BC,4 BC
720 DATA DEATH OF ATILLA THE HUN,FALL OF ROME,BIRTH OF
    MUHAMMAD,453,455,570
730 DATA CLOVIS FIRST FRENCH KING,CHARLEMAGNE'S EMPIRE,
    EGBERT FIRST ENGLISH KING,481,800,828
740 DATA NORMAN INVASION,GENGHIS KHAN TAKES CHINA,KUBLAI
    KHAN IS GREAT KHAN,1066,1214,1260
750 DATA SIR RALEIGH IN VIRGINIA,JAMESTOWN FOUNDED,MAYFLOWER,
    1583,1607,1620
760 DATA SEVEN YEARS' WAR BEGINS,BRITISH TAKE OVER CANADA,
    AMERICAN REVOLUTION,1755,1763,1776
770 DATA CONSTITUTION SIGNED,RAID OF BASTILLE,LOUIS XVI
    BEHEADED,1787,1789,1793
780 DATA NAPOLEON FIGHTS IN EGYPT,NAPOLEON BECOMES EMPEROR,
    NAPOLEON FIGHTS IN RUSSIA,1798,1804,1812
790 DATA ABDICATION OF NAPOLEON,WATERLOO,TREATY OF VIENNA,
    1814,1815,1815
800 DATA CRIMEAN WAR,CIVIL WAR,BOER WAR,1854,1861,1899
810 DATA RUSSO-JAPANESE WAR,WORLD WAR I,RUSSIAN REVOLU-
    TION,1904,1914,1917
820 PRINT "THAT'S ALL THE QUESTIONS":
830 PRINT "RUN AGAIN? (Y/N)":
840 CALL KEY(3,K,S)
850 IF K=78 THEN 880
860 IF K=89 THEN 170
870 GOTO 840
880 CALL CLEAR
890 END

```

Explanation of Program

110-160	Clear screen, print instructions, wait for input to start quiz
170-180	Set number of questions asked and number of wrong guesses to zero
190	Restore the data pointer to first data statement
210	Clear the screen before each question in quiz
220	Read three event names and when they occurred
230	Increment number of questions asked
240-290	Determine random order and print events to screen
300-330	Input user choice for first event

340	Check to see if correct
350-370	If wrong, print wrong message and increment wrong counter
380	If right, print right message
390-420	Input user choice for second event
430	Check to see if correct
440-460	If wrong, print wrong message and increment wrong counter
470	If right, print right message
480-550	Delay, clear screen, print correct order, and print the score
560	Check to see if all data used, if so go to line 820
570-610	User option to attempt another question
620-810	Data statements
820-870	User option to run program again
880-890	Program exit

Important Variables in Program

E\$(3),T\$(3)	= Three event and date strings
E1,E2,E3	= Random order for printing events
N	= Number of questions asked
W	= Number of wrong answers
C	= Choice of events
D	= Delay loop variable
L	= Loop variable to print correct order
K,S	= KEY subprogram variables

Sample Run of Program

```
1. WAR OF 1812
2. FRENCH AND INDIAN WAR
3. AMERICAN REVOLUTION

WHICH EVENT OCCURRED FIRST
PRESS CHOICE? (1,2,3)

RIGHT
FRENCH AND INDIAN WAR
OCCURRED FIRST

WHICH EVENT OCCURRED SECOND
PRESS CHOICE? (1,2,3)

RIGHT
AMERICAN REVOLUTION
OCCURRED SECOND
```

Fig. 5-1.

Suggested Modifications and Projects

1. Add 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 be questioned on well-known events in history, while the expert should be presented with some lesser known events.

PLACES AND CAPITALS

If you have ever tried to learn which capitals are in which states and countries, then this program should be very helpful. This program allows you find out places or capitals you don't know, but it also quizzes you on what you do know.

Listing 5-2

```

100 REM PLACES AND CAPITALS
110 CALL CLEAR
120 RANDOMIZE
130 PRINT TAB(5);"PLACES AND CAPITALS":::
140 PRINT TAB(12);"MENU":::TAB(5);"1. FIND THE CAPITAL"
   ::TAB(5);"2. FIND THE STATE":TAB(8);"OR COUNTRY":::
150 PRINT TAB(5);"3. QUIZ ON CAPITALS":::TAB(5);"4. QUIZ
   ON PLACES":::TAB(5);"5. EXIT THE PROGRAM":::
160 PRINT " YOUR CHOICE? (1,2,3,4,5)":::
170 CALL KEY(3,K,S)
180 IF (S=0)+(K<49)+(K>53)THEN 170
190 CALL CLEAR
200 RESTORE
210 ON K-48 GOTO 220,330,430,450,940
220 PRINT "TYPE THE NAME OF A STATE"::"OR COUNTRY";
230 INPUT C$
240 CALL CLEAR
250 PRINT "SEARCHING DATA...":::
260 FOR L=1 TO 74
270 READ A$,B$
280 IF C$=A$ THEN 310
290 NEXT L
300 GOTO 730
310 PRINT B$;" IS THE"::"CAPITAL OF ";A$:::
320 GOTO 740
330 PRINT "TYPE THE NAME OF A"::"CAPITAL";
340 INPUT C$
350 CALL CLEAR
360 PRINT "SEARCHING DATA...":::
370 FOR L=1 TO 74
380 READ A$,B$
390 IF C$=B$ THEN 420
400 NEXT L
410 GOTO 730
420 GOTO 310

```

cont. on next page

Listing 5-2—cont.

```
430 F=1
440 GOTO 460
450 F=0
460 NQ=0
470 NR=0
480 CALL CLEAR
490 PRINT "SEARCHING DATA....":
500 RESTORE
510 R=INT(74*RND)+1
520 FOR L=1 TO R
530 READ A$,B$
540 NEXT L
550 NQ=NQ+1
560 IF F=1 THEN 600
570 PRINT B$;::"IS THE CAPITAL": "OF WHERE";
580 INPUT C$
590 IF C$=A$ THEN 630 ELSE 660
600 PRINT "THE CAPITAL OF": A$:: "IS ";
610 INPUT C$
620 IF C$=B$ THEN 630 ELSE 660
630 PRINT :: "RIGHT !!!":
640 NR=NR+1
650 GOTO 670
660 PRINT :: "WRONG !!!": B$; " IS THE ": "CAPITAL OF "
;A$::
670 PRINT "YOUR SCORE IS"; NR; "OUT OF"; NQ::
680 PRINT "ANOTHER QUESTION? (Y/N)":
690 CALL KEY(3,K,S)
700 IF K=78 THEN 110
710 IF K=89 THEN 480
720 GOTO 690
730 PRINT :: C$; " IS NOT": "IN THE COMPUTER'S": "DATA B
ANK":
740 PRINT "PRESS ANY KEY TO CONTINUE":
750 CALL KEY(3,K,S)
760 IF S=0 THEN 750
770 GOTO 110
780 DATA ALABAMA, BIRMINGHAM, ALASKA, JUNEAU, ARIZONA, PHOEN
IX, ARKANSAS, LITTLE ROCK, CALIFORNIA, SACRAMENTO
790 DATA COLORADO, DENVER, CONNECTICUT, HARTFORD, DELAWARE,
DOVER, FLORIDA, TALLAHASSEE, GEORGIA, ATLANTA
800 DATA HAWAII, HONOLULU, IDAHO, BOISE, ILLINOIS, SPRINGFIE
LD, INDIANA, INDIANAPOLIS, IOWA, DES MOINES
810 DATA KANSAS, TOPEKA, KENTUCKY, FRANKFORT, LOUISIANA, BAT
ON ROUGE, MAINE, AUGUSTA, MARYLAND, ANNAPOLIS
820 DATA MASSACHUSETTS, BOSTON, MICHIGAN, LANSING, MINNESOT
A, SAINT PAUL, MISSISSIPPI, JACKSON
830 DATA MISSOURI, JEFFERSON CITY, MONTANA, HELENA, NEBRASK
A, LINCOLN, NEVADA, CARSON CITY, NEW HAMPSHIRE, CONCORD
840 DATA NEW JERSEY, TRENTON, NEW MEXICO, SANTA FE, NEW YOR
K, ALBANY, NORTH CAROLINA, RALEIGH, NORTH DAKOTA, BISMAR
K
850 DATA OHIO, COLUMBUS, OKLAHOMA, OKLAHOMA CITY, OREGON, SA
LEM, PENNSYLVANIA, HARRISBURG, RHODE ISLAND, PROVIDENCE
```

```

860 DATA SOUTH CAROLINA,COLUMBIA,SOUTH DAKOTA,PIERRE,TE
NNESEE,NASHVILLE,TEXAS,AUSTIN,UTAH,SALT LAKE CITY
870 DATA VERMONT,MONTPELIER,VIRGINIA,RICHMOND,WASHINGTO
N,OLYMPIA,WEST VIRGINIA,CHARLESTON
880 DATA WISCONSIN,MADISON,WYOMING,CHEYENNE
890 DATA AFGANISTAN,KABUL,ARGENTINA,BUENOS AIRES,AUSTR
ALIA,CANBERRA,AUSTRIA,VIENNA,BELGIUM,BRUSSELS
900 DATA BRAZIL,BRASILIA,CANADA,OTTAWA,COLOMBIA,BOGATA,
CUBA,HAVANA,DENMARK,COPENHAGEN,FRANCE,PARIS
910 DATA EAST GERMANY,BERLIN,WEST GERMANY,BONN,GREECE,A
THENS,IRAN,TEHRAN,IRAQ,BAGHDAD,IRELAND,DUBLIN
920 DATA ITALY,ROME,JAPAN,TOKYO,NORWAY,OSLO,PORTUGAL,LI
SBON,SOVIET UNION,MOSCOW,SPAIN,MADRID
930 DATA GREAT BRITAIN,LONDON
940 CALL CLEAR
950 END

```

Explanation of Program

110-180	Clear screen, print choice menu, and get user's choice
190-200	Re-clear screen, restore data
210	Go to part of program as specified by user's choice
220-230	Input state or country name
240-290	Search data for match
300	If no match, print message at line 730
310-320	Print capital and corresponding state or country, go to wait routine at line 740
330-340	Input capital name
350-400	Search data for match
410	If no match, go to line 730
420	If match, go to line 310
430-450	Set flag to quiz for capitals or for states
460-470	Set number of questions and number right to zero
480-540	Search data for random question
550	Increment number of questions counter
570-590	Input for question on states, checking for correct or incorrect answer
600-620	Input for question on capitals, checking answer
630-650	Print right message, increment number right counter, and go print score
660	Print wrong message
670-720	Print score and user option to attempt another question
730	Print data not found message
740-770	Wait for input to continue, then go to next question
780-930	Data statements
940-950	Program exit

Important Variables in Program

A\$,B\$	= Names of states and capitals
C\$	= User's input of names
F	= Flag for guessing states or guessing capitals
NQ	= Number of questions asked
NR	= Number of right answers
R	= Random number for quizzes
L	= Loop variable to search data
K,S	= KEY subprogram variables

Sample Run of Program

```
TYPE THE NAME OF A STATE
OR COUNTRY? INDIANA
SEARCHING DATA...
INDIANAPOLIS IS THE
CAPITAL OF INDIANA
PRESS ANY KEY TO CONTINUE
```

Fig. 5-2.**Suggested Modifications and Projects**

1. Try adding more capitals from other countries and making the appropriate changes to the program to allow more data.
2. Use graphics to draw maps of the states and countries, with the respective locations of their capitals.

METRIC CONVERTER

Since the metric system is a more efficient, logical, and international measurement system than our own system, people should begin learning more about it. This program will help, since it converts units of measurement both ways.

Listing 5-3

```
100 REM METRIC CONVERTER
110 FOR L=1 TO 4
120 READ E$(L),M$(L),C(L)
130 NEXT L
140 CALL CLEAR
150 PRINT TAB(6);"METRIC CONVERTER":;;;
160 PRINT TAB(4);"1. ENGLISH TO METRIC":;TAB(4);"2. MET
    RIC TO ENGLISH":;TAB(4);"3. EXIT PROGRAM":;;;
170 PRINT TAB(4);"YOUR CHOICE? (1,2,3)":;;;
```

```

180 CALL KEY(3,K,S)
190 IF (S=0)+(K<49)+(K>51)THEN 180
200 N=K-48
210 IF N=3 THEN 540
220 CALL CLEAR
230 PRINT TAB(11);"UNITS":::
240 FOR L=1 TO 4
250 IF N=2 THEN 280
260 PRINT " ";STR$(L);". ";E$(L);" TO ";M$(L)::
270 GOTO 290
280 PRINT " ";STR$(L);". ";M$(L);" TO ";E$(L)::
290 NEXT L
300 PRINT "::TAB(3);"YOUR CHOICE? (1,2,3,4)":::
310 CALL KEY(3,K,S)
320 IF (S=0)+(K<49)+(K>54)THEN 310
330 N2=K-48
340 CALL CLEAR
350 PRINT "ENTER NUMBER OF":
360 IF N=2 THEN 390
370 PRINT E$(N2);
380 GOTO 400
390 PRINT M$(N2);
400 INPUT X
410 CALL CLEAR
420 IF N=2 THEN 450
430 PRINT INT(X*100)/100;E$(N2);::" IS EQUAL TO";::INT(
X*C(N2)*100)/100;M$(N2);::::
440 GOTO 460
450 PRINT INT(X*100)/100;M$(N2);::" IS EQUAL TO";::INT(
X*(1/C(N2))*100)/100;E$(N2);::::
460 PRINT " PRESS ANY KEY TO CONTINUE":
470 CALL KEY(3,K,S)
480 IF S=0 THEN 470
490 GOTO 140
500 DATA INCHES,CENTIMETERS,2.54
510 DATA FEET,METERS,.3048
520 DATA MILES,KILOMETERS,1.6093
530 DATA OUNCES,GRAMS,28.35
540 CALL CLEAR
550 END

```

Explanation of Program

110-130	Loop to read in unit names and conversion factors
140-200	Print main menu and input choice of options
210	Exit program if option three chosen
220-290	Print units menu according to the value of N (number of choice from main menu)
300-330	Input unit choice
340-400	Input value to be converted, also according to N
410-450	Print out converted number, also according to N
460-490	Wait for input before returning to main menu
500-530	Data statements
540-550	Program exit

Important Variables in Program

E\$(4),M\$(4)	= English and metric unit names
C(4)	= Conversion factor between units
L	= Loop variable for reading and printing unit names
N	= Choice for English to metric or metric to English, used as flag throughout program
N2	= Choice of units
X	= Number to convert
K,S	= KEY subprogram variables

Sample Run of Program

```
ENTER NUMBER OF
INCHES? 10

10 INCHES
IS EQUAL TO
25.4 CENTIMETERS

PRESS ANY KEY TO CONTINUE
```

Fig. 5-3.

Suggested Modifications and Projects

1. There are other metric measures for things like 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 next to a meter marked off in feet.

FRENCH FOOD

If you want to impress your friends by ordering in French, try out this program. You can become fluent with the phrases for popular food items. Regardless of whether this wows your friends, they will probably be impressed because you learned the words and phrases from a computer.

Listing 5-4

```
100 REM   FRENCH FOOD
110 CALL CLEAR
120 RANDOMIZE
130 PRINT TAB(5);"FRENCH FOOD READER"::::
140 PRINT TAB(7);"1. LESSONS"::TAB(7);"2. QUIZZES"::TAB
    (7);"3. EXIT PROGRAM"::::
```

```

150 PRINT TAB(4);"YOUR CHOICE? (1,2,3)"::::
160 CALL KEY(0,K,S)
170 IF (S=0)+(K<49)+(K>51)THEN 160
180 ON K-48 GOTO 250,190,920
190 CALL CLEAR
200 F=1
210 PRINT TAB(9);"QUIZZES":::
220 NR=0
230 NQ=0
240 GOTO 280
250 CALL CLEAR
260 F=0
270 PRINT TAB(9);"LESSONS":::
280 PRINT TAB(4);"1. MEAT AND POULTRY":TAB(4);"2. FRUITS":TAB(4);"3. VEGETABLES":
290 PRINT TAB(4);"4. FISH":TAB(4);"5. VARIOUS TERMS":TAB(4);"6. MAIN MENU"::::
300 PRINT TAB(4);"YOUR CHOICE? (1-6)"::::
310 CALL KEY(3,K,S)
320 IF (S=0)+(K<49)+(K>54)THEN 310
330 C=K-48
340 ON C GOTO 350,380,410,440,470,110
350 D=21
360 RESTORE 780
370 GOTO 490
380 D=8
390 RESTORE 820
400 GOTO 490
410 D=13
420 RESTORE 840
430 GOTO 490
440 D=8
450 RESTORE 870
460 GOTO 490
470 D=15
480 RESTORE 890
490 CALL CLEAR
500 IF F=1 THEN 590
510 FOR L=1 TO D
520 READ A$,B$
530 PRINT A$,B$
540 NEXT L
550 PRINT : "PRESS ANY KEY TO CONTINUE"
560 CALL KEY(3,K,S)
570 IF S=0 THEN 560
580 GOTO 250
590 CALL CLEAR
600 NQ=NQ+1
610 Q=INT(D*RND)+1
620 FOR L=1 TO Q
630 READ A$,B$
640 NEXT L
650 PRINT A$;" IS WHAT IN":
660 INPUT "ENGLISH? ":C$
670 IF C$=B$ THEN 750
680 PRINT : "WRONG !!!":A$;" IS ENGLISH": "FOR ";B$;
:::
690 PRINT "YOUR SCORE IS";NR;"OUT OF";NQ;:::

```

cont. on next page

Listing 5-4—cont.

```
700 PRINT "ANOTHER QUESTION? (Y/N)":::
710 CALL KEY(J,K,S)
720 IF K=78 THEN 190
730 IF K=89 THEN 340
740 GOTO 710
750 PRINT :::"RIGHT !!!":::
760 NR=NR+1
770 GOTO 690
780 DATA LA VIANDE,MEAT,L'AGNEAU,LAMB,LE BIFTECK,BEEFST
EAK,LE BEOUF,BEEF,LA COTELETTE,CHOP
790 DATA L'ENTRECOTE,RIB STEAK,L'ESCARGOT,SNAIL,LE FOIE
,LIVER,LE JAMBON,HAM,LE LAPIN,RABBIT,LE RAGOUT,STEW
800 DATA LE ROTI,ROAST,LA SUCISSE,SAUSAGE,LE VEAU,VEAL,
LA VOLAILLE,POULTRY,LE CANARD,DUCK,LE FAISAN,PHEASA
NT
810 DATA LE POULET,CHICKEN,A POINT,MEDIUM,BIEN CUIRE,WE
LL DONE,SAIGNANT,RARE
820 DATA L'ABRICOT,APRICOT,L'AMANDE,ALMOND,LA CERISE,CH
ERRY,LE CITRON,LEMON,LA FRAISE,STRAWBERRY
830 DATA LA FRAMBOISE,RASPBERRY,LA PECHE,PEACH,LE RAISE
N,GRAPE
840 DATA LE LEGUME,VEGETABLE,L'AIL,GARLIC,L'ARTICHAUT,A
RTICHOKE,L'ASPERGE,ASPARAGUS,LE CHAMPIGNON,MUSHROOM
850 DATA LE CHOU,CABBAGE,LES EPINARDS,SPINACH,BA LAITUE
,LETTUCE
860 DATA L'OIGNON,ONION,LE RADIS,RADISH
870 DATA LE POISSON,FISH,LA CREVETTE,SHRIMP,LE HARENG,H
ERRING,LE HOMARD,LOBSTER,L'HUITRE,OYSTER,LA MORUE,C
OD
880 DATA LE SAUMON,SALMON,LA TRUITE,TROUT
890 DATA LA BOISSON,DRINK,LE LAIT,MILK,LE VIN,WINE,LE P
AIN,BREAD,L'OEUF,EGG,LE RIZ,RICE,L'EPICE,SPICE
900 DATA LE SEL,SALT,LE POIVRE,PEPPER,LA MOUTARDE,MUSTA
RD,LE MIEL,HONEY,LA CREPE,PANCAKE,LE FLAN,CUSTARD
910 DATA LA PATISserie,PASTRY,LA RECETTE,RECIPE
920 CALL CLEAR
930 END
```

Explanation of Program

110-170	Clear screen, print choice menu, and get user's choice
180	Go to part of program as specified by user's choice
190-210	Clear screen, set flag for quizzes, print quiz title
220-230	Set number of questions and right answers to zero
240	Go print second menu
250-270	Clear screen, set flag for lessons, print lesson title
280-330	Print second menu and input choice
340	Go to lines according to second menu choice
350-480	Set number of data items and restore line for five different sets of data

490	Clear screen
500	If quiz go to line 590
510-540	Loop to list lessons to screen
550-580	Wait for input, then return to second menu
590	Clear screen
600	Increment number of questions asked
610-640	Get random data for quiz question
650-660	Input English guess
670	If right, then go to line 750
680-690	Print wrong message, correct answer and score
700-740	User option for another question
750-770	Print right message, increment right answer counter, go to line 690 to print score
780-910	Data statements
920-930	Program exit

Important Variables in Program

F	= Flag for lessons or quizzes
NQ,NR	= Number of questions and right answers
C	= Category choice
D	= Number of data items to read
L	= Loops to read and print data
A\$,B\$	= French and English food names
Q	= Random number for quiz question
K,S	= KEY subprogram variables

Sample Run of Program

LA VIANDE IS WHAT IN
ENGLISH? MEAT
RIGHT !!!
YOUR SCORE IS 1 OUT OF 1
ANOTHER QUESTION? (Y/N)

Fig. 5-4.

Suggested Modifications and Projects

1. You have probably realized that there are many other French words that could be learned. Think of other words that fit into categories and add them to this program.
2. Try making use of sentences within the quizzes, so that hints may be gained by seeing the word used in context.

GEOMETRY

Mathematics is simple for the TI-99/4A, but for those struggling through geometry in school, it may not seem so easy. This program will automatically calculate the area of two dimensional objects and calculate the area and volume of three dimensional objects.

Listing 5-5

```
100 REM      GEOMETRY
110 PI=3.1415
120 FOR L=1 TO 4
130 READ N$(1,L),N$(2,L)
140 NEXT L
150 CALL CLEAR
160 PRINT TAB(9);"GEOMETRY":;;;
170 PRINT TAB(7);"1. PLANES":TAB(7);"2. SOLIDS":TAB(7)
   );"3. EXIT PROGRAM":;;;
180 PRINT TAB(4);"YOUR CHOICE? (1,2,3)":;;;
190 CALL KEY(0,K,S)
200 IF (S=0)+(K<49)+(K>51)THEN 190
210 C=K-48
220 ON C GOTO 230,250,1090
230 F=1
240 GOTO 260
250 F=2
260 CALL CLEAR
270 PRINT TAB(9);"SHAPES":;;
280 FOR L=1 TO 4
290 PRINT TAB(7-F);STR$(L);". ";N$(F,L);;;
300 NEXT L
310 PRINT ::TAB(3+F);"YOUR CHOICE? (1-4)":;;
320 CALL KEY(3,K,S)
330 IF (S=0)+(K<49)+(K>52)THEN 320
340 C2=K-48
350 CALL CLEAR
360 ON C2+(4*(F-1))GOTO 370,370,500,610,660,790,880,970
370 INPUT "ENTER BASE? ":B
380 PRINT
390 IF B<0 THEN 370
400 INPUT "ENTER ALTITUDE? ":H
410 PRINT
420 IF H<0 THEN 400
430 A=B*H
440 IF C2=1 THEN 480
450 A=A/2
460 PRINT
470 IF R<0 THEN 610
480 PRINT ::"THE AREA IS";INT(A*100)/100;::;
490 GOTO 1030
500 INPUT "ENTER BASE1? ":B1
510 PRINT
520 IF B1<0 THEN 500
530 INPUT "ENTER BASE2? ":B2
```

```
540 PRINT
550 IF B2<0 THEN 530
560 INPUT "ENTER ALTITUDE? ":H
570 PRINT
580 IF H<0 THEN 560
590 A=(H/2)*(B1+B2)
600 GOTO 480
610 INPUT "RADIUS? ":R
620 PRINT
630 IF R<0 THEN 610
640 A=PI*R^2
650 GOTO 480
660 INPUT "ENTER LENGTH? ":L
670 PRINT
680 IF L<0 THEN 660
690 INPUT "ENTER WIDTH? ":W
700 PRINT
710 IF W<0 THEN 690
720 INPUT "ENTER HEIGHT? ":H
730 PRINT
740 IF H<0 THEN 720
750 V=L*W*H
760 S=2*((L*W)+(W*H)+(L*H))
770 PRINT :: "THE SURFACE AREA": "IS"; INT(S*100)/100; ::
    "THE VOLUME IS"; INT(V*100)/100; ::
780 GOTO 1030
790 INPUT "ENTER ALTITUDE? ":H
800 PRINT
810 IF H<0 THEN 790
820 INPUT "ENTER RADIUS? ":R
830 PRINT
840 IF R<0 THEN 820
850 V=PI*H*R^2
860 S=2*PI*R*(H+R)
870 GOTO 770
880 INPUT "ENTER ALTITUDE? ":H
890 PRINT
900 IF H<0 THEN 880
910 INPUT "ENTER RADIUS? ":R
920 PRINT
930 IF R<0 THEN 910
940 V=(PI/3)*H*R^2
950 S=PI*R*SQR(R^2+H^2)
960 GOTO 770
970 INPUT "ENTER RADIUS? ":R
980 PRINT
990 IF R<0 THEN 970
1000 V=(4/3)*PI*R^3
1010 S=PI*4*R^2
1020 GOTO 770
1030 PRINT "PRESS ANY KEY TO CONTINUE": ::
1040 CALL KEY(3,K,S)
1050 IF S=0 THEN 1040
1060 GOTO 150
1070 GOTO 1070
```

cont. on next page

Listing 5-5—cont.

```
1080 DATA RECTANGLE,RECTANGULAR SOLID,TRIANGLE,CYLINDER
      ,TRAPEZOID,CONC,CIRCLE,SPHERE
1090 CALL CLEAR
1100 END
```

Explanation of Program

110	Set value of PI
120-140	Loop to read shape names into array N\$(2,4)
150-210	Clear screen, print choice menu, and get user's choice
220	Go to part of program as specified by user's choice
230-250	Set flag for two or three dimensional shapes
260-340	Clear screen, print second choice menu according to flag, and get user's choice
350	Clear screen
360	Go to routine according to flag and choice two
370-420	Input base and altitude for rectangles and triangles
430-450	Figure area for rectangles or triangles
480-490	Print area to screen, go to wait routine
500-580	Input two bases and altitude for trapezoid
590-600	Figure area for trapezoid, go to line 480
610-630	Input radius for circle
640-650	Figure area for circle, go to line 480
660-740	Input length, width, and height for rectangular solid
750-760	Figure surface area and volume for rectangular solid
770-780	Print surface area and volume to screen, go to wait routine
790-840	Input altitude and radius for cylinder
850-870	Figure surface area and volume for cylinder, go to line 770
890-930	Input altitude and radius for cone
940-960	Figure surface area and volume for cone, go to line 770
970-990	Input radius for sphere
1000-1020	Figure surface area and volume for sphere, go to line 770
1030-1060	Wait for input then return to main menu
1080	Data statements
1090-1100	Program exit

Important Variables in Program

PI	= Value of PI
L	= Loop variable to read in shape names
N\$(2,4)	= Array to hold names of shapes

C	= Choice from main menu
F	= Flag from choice of main menu to decide which shape names are to be printed in second menu
C2	= Choice from second menu
A	= Area of two-dimensional shape
SA	= Surface area of three-dimensional shape
V	= Volume of three-dimensional shape
B,B1,B1	= Length of bases of shapes
H	= Altitude or heights of shapes
L,W	= Length and width of cubes
R	= Radius of shapes
K,S	= KEY subprogram variables

Sample Run of Program

```

ENTER BASE? 5
ENTER ALTITUDE? 5
THE AREA IS 25
PRESS ANY KEY TO CONTINUE

```

Fig. 5-5.

Suggested Modifications and Projects

1. Utilize graphics to draw the shapes to the screen.
2. Set up a quiz where the user has to input formulas for different shapes. This will teach the method used to attain areas and volumes of the different figures.

COMPUTER TUTOR #1

This program is designed to make people more familiar with the most important BASIC keywords that the TI-99/4A has. Once the computer is told which keyword is wanted, it will give a brief description and sometimes a demonstration of that keyword.

Listing 5-6

```

100 REM  COMPUTER TUTOR #1
110 CALL CLEAR
120 PRINT TAB(6);"COMPUTER TUTOR #1":::
130 PRINT "WHICH OF THE FOLLOWING WOULD":"YOU LIKE TO K
NOW MORE ABOUT?":::
140 PRINT TAB(8);"1. ABS":TAB(8);"2. ASC":TAB(8);"3. CH
R$:TAB(8);"4. DATA ":TAB(8);"5. FOR/NEXT"

```

cont. on next page

Listing 5-6—cont.

```
150 PRINT TAB(8);"6. GOSUB":TAB(8);"7. GOTO":TAB(8);"8.  
IF/THEN":TAB(8);"9. INPUT":TAB(8);"0. EXIT PROGRAM  
":  
160 PRINT TAB(5);"YOUR CHOICE? (0-9)":  
170 CALL KEY(3,K,S)  
180 IF (S=0)+(K<48)+(K>57)THEN 170  
190 CALL CLEAR  
200 IF K=48 THEN 1020  
210 ON K-48 GOTO 220,290,360,420,500,600,760,830,900  
220 PRINT "ABS=ABSOLUTE VALUE FUNCTION":  
230 PRINT "THE ABS COMMAND FUNCTION":"RETURNS THE ABSOL  
UTE VALUE":"OF A NUMBER. THIS MEANS "  
240 PRINT "THAT THE NUMBER YOU GIVE THE":"COMPUTER WILL  
BE RETURNED":"WITHOUT A SIGN. IF YOU GIVE"  
250 PRINT "A POSITIVE NUMBER, IT WILL":"GIVE BACK A POS  
ITIVE NUMBER.":"IF YOU GIVE A NEGATIVE"  
260 PRINT "NUMBER, THEN IT WILL GIVE":"BACK THE POSITIV  
E OF THE":"NUMBER. FOR EXAMPLE, THE"  
270 PRINT "STATEMENT [PRINT ABS(X)]":"WILL PRINT A POSI  
TIVE NUMBER":"REGARDLESS OF THE VALUE":"OF 'X'.":  
280 GOTO 980  
290 PRINT "ASC=ASCII FUNCTION VALUE":  
300 PRINT "TO DETERMINE THE ASCII VALUE":"OF A CHARACTE  
R, THE ASC":"FUNCTION MUST BE USED."  
310 PRINT "THE ASCII VALUE IS A NUMBER":"WHICH CORRESPO  
NDS TO THE":"COMPUTER'S CODE FOR THAT"  
320 PRINT "PARTICULAR CHARACTER. EACH":"CHARACTER ON T  
HE TI-99/4A":"HAS ITS OWN ASCII VALUE. IF"  
330 PRINT "YOU TYPED [PRINT ASC(,CHR$(34);"A";CHR$(34)  
;)],":"A '65' WOULD BE PRINTED"  
340 PRINT "SINCE THE LETTER 'A' HAS AN":"ASCII VALUE OF  
65.":  
350 GOTO 980  
360 PRINT "CHR$=CHARACTER STRING FCTN":  
370 PRINT "THE CHR$ COMMAND IS OPPOSITE":"OF THE ASCII  
FUNCTION. BY":"USING THE CHR$ FUNCTION,"  
380 PRINT "YOU MAY DETERMINE WHAT":"CHARACTER A CERTAIN  
ASCII":"NUMBER CORRESPONDS TO. FOR"  
390 PRINT "EXAMPLE, IF YOU TYPED":"[PRINT CHR$(65)], AN  
'A':"WOULD BE PRINTED SINCE A'S"  
400 PRINT "ASCII VALUE IS 65.":  
410 GOTO 980  
420 PRINT "DATA=DATA TO BE READ":  
430 PRINT "THE DATA STATEMENT IS USED":"TO STORE A LIST  
OF CONTANTS":"WHICH CAN BE EITHER WORDS"  
440 PRINT "OR NUMBERS. EACH CONSTANT":"MUST BE SEPARAT  
ED BY A":"COMA (,) SO THE COMPUTER"  
450 PRINT "WILL KNOW ONE CONSTANT FROM":"ANOTHER. TO R  
ETRIEVE THE"  
460 PRINT "DATA, A 'READ' COMMAND MUST":"BE USED ELSEWH  
ERE IN THE":"PROGRAM. THE FOLLOWING"  
470 PRINT "PROGRAM WILL READ THE NUMBER":"TEN FROM THE  
DATA STATEMENT":"AND PRINT IT ON THE SCREEN.":  
480 PRINT "10 READ A":"20 PRINT A":"30 DATA 10":  
490 GOTO 980
```

```

500 PRINT "FOR...NEXT=FOR/NEXT LOOP":
510 PRINT "THE FOR/NEXT LOOP IS SIMPLY":"A COUNTING DEV
    ICE. IF YOU":"WANT TO COUNT TO FIVE,"
520 PRINT "YOU WOULD SAY 'ONE','TWO','THREE','FOUR',
    'FIVE'. FOR":"THE COMPUTER TO COUNT TO"
530 PRINT "FIVE, YOU HAVE TO TELL IT":"THE STARTING NUM
    BER, THE":"ENDING NUMBER, AND THE STEP."
540 PRINT "THE STEP IS JUST HOW FAST":"THE COMPUTER COU
    NTS. IT":"COULD COUNT 1,2,3,4,5 OR IT"
550 PRINT "COULD COUNT 1,3,5. THE":"FIRST EXAMPLE WOULD
    BE A":"STEP OF 1 AND THE SECOND"
560 PRINT "EXAMPLE A STEP OF 2.":"PRESS ANY KEY TO CON
    TINUE":
570 CALL KEY(3,K,S)
580 IF S=0 THEN 570
590 CALL CLEAR
600 PRINT "TO TELL THE COMPUTER WHERE":"TO START AND ST
    OP, YOU":"MERELY SPECIFY THE NUMBER"
610 PRINT "IN THE FOR/NEXT LOOP. IF":"YOU WANT THE COM
    PUTER TO":"COUNT FROM 2 TO 8, YOU WOULD"
620 PRINT "TYPE [FOR A=2 TO 8]. YOU":"CAN USE ANY VARI
    ABLE YOU":"WANT ('A' IN THIS CASE) TO"
630 PRINT "DO THE COUNTING. IF YOU":"WANTED THE COMPUT
    ER TO COUNT":"IN STEPS OF 2 (2,4,6,8)"
640 PRINT "INSTEAD OF STEPS OF 1,":"TYPE [FOR A=2 TO 8
    STEP 2]."
650 PRINT "IF YOU WANT TO COUNT":"BACKWARDS (LIKE 10 TO
    1) YOU"
660 PRINT "WOULD USE A NEGATIVE STEP":"(LIKE -1).":
670 GOTO 980
680 PRINT "GOSUB=GO TO A SUBROUTINE":
690 PRINT "GOSUB IS A COMMAND TO TELL":"THE COMPUTER TO
    GO TO A":"DIFFERENT PART OF THE"
700 PRINT "PROGRAM, PERFORM A TASK, AND":"RETURN TO THE
    PLACE JUST":"AFTER WHERE THE SUBROUTINE"
710 PRINT "WAS ORDERED. THE FOLLOWING":"PROGRAM WOULD
    INSTRUCT THE":"COMPUTER TO GO A PRINTING"
720 PRINT "SUBROUTINE AT LINE 50, AND":"RETURN TO END T
    HE PROGRAM.":
730 PRINT "10 A=10":"20 GOSUB 50":"30 PRINT ";CHR$(34);
    "PROGRAM FINISHED";CHR$(34);"40 END"
740 PRINT "50 PRINT A":"60 RETURN":
750 GOTO 980
760 PRINT "GOTO=GO TO ANOTHER LINE":
770 PRINT "THE GOTO STATEMENT TELLS THE":"COMPUTER TO G
    O TO ANOTHER":"PART OF A PROGRAM. THE"
780 PRINT "GOTO STATEMENT IS FOLLOWED":"BY THE LINE NUM
    BER THE":"PROGRAM SHOULD GO TO. THE"
790 PRINT "FOLLOWING PROGRAM WILL PRINT":"'TI-99/4A' OV
    ER AND OVER,":"SINCE LINE 30 TELLS THE"
800 PRINT "COMPUTER TO GO TO THE":"BEGINNING OF THE PRO
    GRAM.":
810 PRINT "10 A$=";CHR$(34);"TI-99/4A";CHR$(34);"20 PR
    INT A$":"30 GOTO 10":
820 GOTO 980
830 PRINT "IF/THEN=LOGICAL DECISIONS":
840 PRINT "FOR THE COMPUTER TO MAKE A":"DECISION AND GO
    TO A RESULT,":"THE IF/THEN STATEMENT IS" cont. on next page

```

Listing 5-6—cont.

```
850 PRINT "USED. THE COMMAND SIMPLY":"TELLS THE COMPUT
ER TO GO":"SOMEWHERE 'THEN' IF 'IF'"
860 PRINT "AN EXPRESSION IS TRUE. THE":"EXPRESSION CAN
BE LOGICAL":"OR MATHEMATICAL. FOR"
870 PRINT "EXAMPLE, IF YOU WANTED THE":"COMPUTER TO GO
TO LINE 100":"IF 'A' IS EQUAL TO 10, THEN"
880 PRINT "YOU WOULD TYPE:":"IF A=10 THEN 100"::
890 GOTO 980
900 PRINT "INPUT=INPUT VARIABLES"::
910 PRINT "THE INPUT STATEMENT LETS A":"COMPUTER USER E
NTER DATA":"INTO THE COMPUTER, IN THE"
920 PRINT "FORM OF LETTERS, NUMBERS, OR":"BOTH. FOR TH
E COMPUTER TO":"ACCEPT NUMBERS, A NUMERIC"
930 PRINT "VARIABLE (LIKE B) MUST BE":"USED FOR THE INP
UT VARIABLE.":"FOR ALPHANUMERIC DATA TO BE"
940 PRINT "ACCEPTED, A STRING VARIABLE":"(LIKE B$) MUST
BE USED.":"SEVERAL PIECES OF DATA MAY"
950 PRINT "BE INPUT LIKE THIS:":"10 INPUT A,B$,C,D$"::
"THIS COMMAND WOULD INPUT TWO"
960 PRINT "NUMBERS AND TWO STRINGS."::
970 GOTO 980
980 PRINT "PRESS ANY KEY TO CONTINUE"::
990 CALL KEY(3,K,S)
1000 IF S=0 THEN 990
1010 GOTO 110
1020 END
```

Explanation of Program

110-180	Clear screen, print menu, get user's choice
190	Clear screen
200	Check for exit choice
210	Go to part of program as specified by user's choice
220-280	ABS
290-350	ASC
360-410	CHR\$
420-490	DATA
500-670	FOR/NEXT
680-750	GOSUB
760-820	GOTO
830-890	IF/THEN
900-970	INPUT
980-1010	Wait for input, go back to main menu
1020	Program exit

Important Variables in Program

K,S = KEY subprogram variables

Sample Run of Program**INPUT = INPUT VARIABLES**

**THE INPUT STATEMENT LETS A
COMPUTER USER ENTER DATA
INTO THE COMPUTER, IN THE
FORM OF LETTERS, NUMBERS, OR
BOTH. FOR THE COMPUTER TO
ACCEPT NUMBERS, A NUMERIC
VARIABLE (LIKE B) MUST BE
USED**

Fig. 5-6.**Suggested Modifications and Projects**

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

COMPUTER TUTOR #2

Computer Tutor #2 has the same purpose as Tutor #1, except that it covers nine different BASIC keywords.

Listing 5-7

```

100 REM   COMPUTER TUTOR #2
110 CALL CLEAR
120 PRINT TAB(6); "COMPUTER TUTOR #2":::
130 PRINT "WHICH OF THE FOLLOWING WOULD": "YOU LIKE TO K
    NOW MORE ABOUT?":::
140 PRINT TAB(8); "1. LEN": TAB(8); "2. PRINT": TAB(8); "3.
    READ": TAB(8); "4. REM": TAB(8); "5. RND"
150 PRINT TAB(8); "6. SEG$": TAB(8); "7. STR$": TAB(8); "8.
    TAB": TAB(8); "9. VAL": TAB(8); "0. EXIT PROGRAM":::
160 PRINT TAB(5); "YOUR CHOICE? (0-9)":::
170 CALL KEY(3, K, S)
180 IF (S=0)+(K<48)+(K>57) THEN 170
190 CALL CLEAR
200 IF K=48 THEN 850
210 ON K-48 GOTO 220, 280, 340, 410, 480, 540, 610, 680, 750
220 PRINT "LEN=LENGTH OF STRING":
230 PRINT "THE LEN FUNCTION IS USED TO": "FIND THE LENGTH
    H OF A STRING.": "THE STRING 'CLOCKWORK', FOR"
240 PRINT "EXAMPLE, WOULD HAVE A VALUE": "OF '9' IF YOU
    TYPED IN": "[PRINT LEN('CLOCKWORK')], "
250 PRINT "SINCE THE STRING HAS NINE": "CHARACTERS IN IT
    . IF A": "STRING CALLED 'B$' HAS TEN"

```

cont. on next page

Listing 5-7—cont.

```
260 PRINT "CHARACTERS IN IT, AND YOU": "TYPED [L=LEN(B$)
    ], THEN": "L WOULD BE EQUAL TO 10.":::
270 GOTO 810
280 PRINT "PRINT=DISPLAY SOMETHING"::
290 PRINT "THE PRINT COMMAND CAN BE": "USED TO DISPLAY V
    ARIABLES": "OR MESSAGES TO THE SCREEN."
300 PRINT "IF YOU WANT TO PRINT THE": "VALUE OF A VARIAB
    LE, THEN": "YOU WOULD TYPE": "PRINT A"::
310 PRINT "IF YOU WANT TO PRINT A": "MESSAGE TO THE SCRE
    EN, YOU": "WOULD TYPE"::
320 PRINT "PRINT "; CHR$(34); "THIS IS A MESSAGE"; CHR$(34
    );:::
330 GOTO 810
340 PRINT "READ=READ DATA"::
350 PRINT "THE READ STATEMENT IS USED": "ALONG WITH THE
    DATA": "STATEMENT. READ IS USED"
360 PRINT "TO READ DATA FROM ANOTHER": "PART OF THE PROG
    RAM,": "WHETHER THAT DATA IS WORDS"
370 PRINT "OR LETTERS. THIS EXAMPLE": "WOULD READ THE V
    ARIABLE 'A',": "READ THE WORD 'A$', AND"
380 PRINT "PRINT BOTH OF THEM"::
390 PRINT "10 READ A,A$": "20 PRINT A,A$": "30 DATA 12,TI
    -99/4A"::
400 GOTO 810
410 PRINT "REM=REMARK STATEMENT"::
420 PRINT "IF YOU WISH TO PUT A MESSAGE": "WITHIN A PROG
    RAM, FOR YOUR": "OWN USE OR FOR OTHERS,"
430 PRINT "THE REM STATEMENT IS USED.": "WHENEVER THE CO
    MPUTER": "ENCOUNTERS A REM STATEMENT,"
440 PRINT "IT SIMPLY GOES ON TO THE": "NEXT PROGRAM LINE
    AND"
450 PRINT "IGNORES EVERYTHING FOLLOWING": "THE REM ON TH
    AT LINE. YOU": "CAN THEREFORE PUT ANYTHING"
460 PRINT "YOU WISH AFTER A REM": "STATEMENT.": "10 REM
    THIS IS A REMARK"::
470 GOTO 810
480 PRINT "RND=GENERATE A RANDOM NUMBER"::
490 PRINT "RND IS A COMMAND THAT": "GENERATES A RANDON N
    UMBER": "BETWEEN 0 AND 1. IF YOU"
500 PRINT "WANT A WHOLE NUMBER, YOU": "MUST USE THE FOLL
    OWING": "EQUATION WITH 'A' BEING"
510 PRINT "THE UPPER RANGE OF THE": "NUMBER THAT YOU WAN
    T.": "R=INT(RND*A)+1": "R WOULD BE EQUAL TO A RANDO
    M"
520 PRINT "NUMBER BETWEEN 1 AND A."::
530 GOTO 810
540 PRINT "SEG$=PART OF STRING FUNCTION"::
550 PRINT "THE SEG$ FUNCTION GETS PARTS": "OF STRINGS.
    THE FORMAT": "FOR A SEG$ IS": "SEG$(STRING$, #1, #2)":
    :
560 PRINT "STRING$ SPECIFIES THE WHOLE": "STRING, #1 TEL
    LS THE NUMBER": "OF THE FIRST CHARACTER YOU"
570 PRINT "WANT TO EXTRACT, AND #2": "TELLS HOW LONG YOU
    WANT THE": "PART TO BE. THE FOLLOWING"
580 PRINT "PROGRAM EXTRACTS AND PRINTS": "PART OF A$"::
```

```

590 PRINT "10 A$=";CHR$(34);"DOGHOUSE";CHR$(34);;"20 T$
=SEG$(A$,4,5)";;"30 PRINT T$";:
600 GOTO 810
610 PRINT "STR$=NUMBER TO A STRING"::
620 PRINT "IF YOU HAVE A NUMBER AND":"WANT TO MAKE A ST
RING OUT":"OF IT, YOU MUST USE THE"
630 PRINT "STR$ COMMAND. ASSUME THAT":"A VARIABLE CALL
ED 'T' IS":"EQUAL TO 67 AND YOU WANT"
640 PRINT "TO MAKE A STRING WITH T'S":"VALUE. THE FOLL
OWING COMMAND":"WILL DO THIS"::
650 PRINT "T$=STR$(T)":"ONCE THIS HAS BEEN DONE,":"T W
ILL STILL EQUAL 67, BUT"
660 PRINT "T$ WILL BE EQUAL TO THE":"STRING '67'"::
670 GOTO 810
680 PRINT "TAB=MOVE CURSOR TO RIGHT"::
690 PRINT "JUST LIKE ON THE TYPEWRITER,":"THE TAB FUNCT
ION MOVES THE":"WHATEVER IS TO BE PRINTED"
700 PRINT "A CERTAIN NUMBER OF SPACES":"TO THE RIGHT.
THIS HELPS":"TO MAKE SCREENS APPEAR NEAT."
710 PRINT "THE FORMAT IS":"PRINT TAB(X);CHR$(34);"ME
SSAGE";CHR$(34);;"WITH 'X' BEING THE NUMBER"
720 PRINT "OF SPACES. A STATEMENT":"LIKE [PRINT TAB(15
)];CHR$(34);"HI";CHR$(34);"]"
730 PRINT "WOULD HAVE THE FOLLOWING":"RESULT"::TAB(15)
;"HI"::
740 GOTO 810
750 PRINT "VAL=VALUE OF A STRING"::
760 PRINT "TO FIND OUT WHAT THE VALUE":"OF A STRING IS,
YOU SHOULD":"USE THE VAL STATEMENT."
770 PRINT "STRINGS ARE NOT NUMBERS SO":"THIS CONVERSION
STATEMENT":"MUST BE USED. SUPPOSE A"
780 PRINT "STRING CALLED 'NM$' WAS":"'435'. TO FIND OU
T THE":"VALUE OF THAT STRING, YOU":"WOULD TYPE"::
790 PRINT "PRINT VAL(NM$)":"THE COMPUTER WOULD RETURN
A":"VALUE OF 435"::
800 GOTO 810
810 PRINT "PRESS ANY KEY TO CONTINUE"::
820 CALL KEY(3,K,S)
830 IF S=0 THEN 820
840 GOTO 110
850 END

```

Explanation of Program

110-180	Clear screen, print menu, get user's choice
190	Clear screen
200	Check for exit choice
210	Go to part of program as specified by user's choice
220-270	LEN
280-330	PRINT
340-400	READ
410-470	REM
480-530	RND
540-600	SEG\$

610-670	STR\$
680-740	TAB
750-800	VAL
810-840	Wait for input, go back to main menu
850	Program exit

Important Variables in Program

K,S = KEY subprogram variables

Sample Run of Program

**TAB = MOVE CURSOR TO RIGHT
JUST LIKE ON THE TYPEWRITER,
THE TAB FUNCTION MOVES
WHATEVER IS TO BE PRINTED
A CERTAIN NUMBER OF SPACES
TO THE RIGHT. THIS HELPS
THE SCREENS TO APPEAR NEAT.
THE FORMAT**

Fig. 5-7.

Suggested Modifications and Projects

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

Chapter 6

Programs for the Home

When you first bought your computer, the person selling the machine probably noted its uses for the home, such as checkbook balancing, recipe converting, and figuring out mortgages. However, you may have had some trouble getting your computer to do these things. In this chapter, three programs supply you with the tools to make your home more organized and efficient. There are, of course, countless applications of computers within the home, but these programs should give you a good idea how easily a TI-99/4A can accomplish some of these tasks.

CHECKBOOK

Have you ever wondered if the bank ever makes mistakes when it prints your bank statement? This program will give you a quick and easy way to double check the figures.

Listing 6-1

```
100 REM  CHECKBOOK
110 CALL CLEAR
120 PRINT TAB(3);"WELCOME TO CHECKBOOK":::
130 PRINT "THIS PROGRAM ALLOWS YOU":"DOUBLE CHECK YOUR
    BANK":"STATEMENT FOR ERRORS"::
```

cont. on next page

Listing 6-1—cont.

```
140 PRINT "ENTER THE FOLLOWING AMOUNTS": "FROM YOUR BANK
    STATEMENT": "DOLLAR SIGNS ARE NOT NEEDED":
150 PRINT "USE THE <FCTN><S> KEY TO": "BACK SPACE":
160 PRINT "    PRESS ANY KEY TO START":
170 CALL KEY(3,K,S)
180 IF S=0 THEN 170
190 CALL CLEAR
200 F=1
210 H=H+1
220 R=H*2+3
230 C=2
240 ON H GOTO 620,640,660,680,700,720,250
250 F=0
260 IF EB<>BB+TD-TW-SC+IN THEN 380
270 P$="THE STATEMENT IS CORRECT  "
280 R=20
290 C=2
300 GOSUB 1140
310 P$="PRESS ANY KEY TO CONTINUE"
320 R=22
330 C=2
340 GOSUB 1140
350 CALL KEY(3,K,S)
360 IF S=0 THEN 350
370 GOTO 1180
380 P$="AN ERROR HAS BEEN MADE"
390 R=20
400 C=2
410 GOSUB 1140
420 P$="ALL ENTRIES CORRECT? (Y/N)"
430 R=22
440 C=2
450 GOSUB 1140
460 CALL KEY(3,K,S)
470 IF K=78 THEN 500
480 IF K=89 THEN 1080
490 GOTO 460
500 P$="CHANGE WHICH ENTRY? (1-6)  "
510 R=22
520 C=2
530 GOSUB 1140
540 CALL KEY(3,K,S)
550 IF (S=0)+(K<49)+(K>54) THEN 540
560 H=K-48
570 P$="                                "
580 R=H*2+3
590 C=2
600 GOSUB 1140
610 ON H GOTO 620,640,660,680,700,720
620 P$="1. BEGINNING BAL?"
630 GOTO 730
640 P$="2. TOTAL DEPOSITS?"
650 GOTO 730
660 P$="3. TOTAL WDRAWALS?"
```

```

670 GOTO 730
680 P$="4. SERVICE CHARGE?"
690 GOTO 730
700 P$="5. INTEREST ?"
710 GOTO 730
720 P$="6. ENDING BALANCE?"
730 GOSUB 1140
740 C=22
750 A$=""
760 CALL HCHAR(R,C,94)
770 CALL HCHAR(R,C,32)
780 CALL KEY(3,K,S)
790 IF S=0 THEN 760
800 IF (K>47)*(K<58)*(C<32) THEN 850
810 IF (K=46)*(C<32) THEN 850
820 IF (K=8)*(C>22) THEN 890
830 IF K=13 THEN 920
840 GOTO 760
850 CALL HCHAR(R,C,K)
860 A$=A$&CHR$(K)
870 C=C+1
880 GOTO 760
890 A$=SEG$(A$,1,LEN(A$)-1)
900 C=C-1
910 GOTO 760
920 IF LEN(A$)>0 THEN 940
930 A$=""
940 ON H GOTO 950,970,990,1010,1030,1050
950 BB=VAL(A$)
960 GOTO 1060
970 TD=VAL(A$)
980 GOTO 1060
990 TW=VAL(A$)
1000 GOTO 1060
1010 SC=VAL(A$)
1020 GOTO 1060
1030 IN=VAL(A$)
1040 GOTO 1060
1050 EB=VAL(A$)
1060 IF F=1 THEN 210
1070 GOTO 260
1080 CALL CLEAR
1090 PRINT "PLEASE CONTACT YOUR BANK":;"TO HAVE THEM DO
      UBLE CKECK":;"YOUR BANK STATEMENT":;:
1100 PRINT "PRESS ANY KEY TO CONTINUE":;:
1110 CALL KEY(3,K,S)
1120 IF S=0 THEN 1110
1130 GOTO 1180
1140 FOR P=1 TO LEN(P$)
1150 CALL HCHAR(R,P+C,ASC(SEG$(P$,P,1)))
1160 NEXT P
1170 RETURN
1180 CALL CLEAR
1190 END

```

Explanation of Program

110-190	Clear screen, print instructions, wait for input to start
200	Set flag to run through all headings the first time
210-230	Set heading number to work on, set the row and column to print on screen for that heading
240	Go to different parts of program in input information depending on heading number
250	Set flag for individual heading changes
260	Check the amounts to see if statement correct
270-370	If statement correct, print correct message, wait for input to continue, and exit program
380-410	If statement incorrect, print error message
420-490	User option if all entries correct
500-560	If entry not correct, then user option to change an amount under any heading number
570-730	Print heading number to be changed
740-930	Routine to allow user to change amount on screen with a moving cursor—setting A\$ equal to current amount showing on screen
940-1050	Using heading number being worked on, set that heading amount equal to the value of A\$
1060	If flag for doing all headings, then go to line 210
1070	Otherwise, go to line 260
1080-1090	If still error and all entries correct, then print bank-made-error message, wait for input to continue, then exit program
1140-1170	Print subroutine—print message P\$ on row R starting at column C
1180-1190	Program exit

Important Variables in Program

F	= Flag set to one for first time through to change all six headings
H	= Heading number that is being worked on
R,C	= Row and column to print on
P\$	= Message string for print subroutine
A\$	= String variable to hold contents of value being entered on screen with moving cursor
BB	= Beginning balance
TD	= Total deposits
TW	= Total withdrawals

SC = Service charge
 IN = Interest
 EB = Ending balance
 P = Loop for print subroutine
 K,S = KEY subprogram variables

Sample Run of Program

```

1. BEGINNING BAL? 100
2. TOTAL DEPOSITS? 100
3. TOTAL WDRAWALS? 100
4. SERVICE CHARGE? 10
5. INTEREST? 10
6. ENDING BALANCE? 100

THE STATEMENT IS CORRECT

PRESS ANY KEY TO CONTINUE
  
```

Fig. 6-1.

Suggested Modifications and Projects

1. Modify the program so it also adds the check amounts and outstanding checks.
2. Make the program work to help the user balance his checkbook.

RECIPE CONVERTER

There are many recipes, but most of them are for the wrong number of servings to suit your needs. This program does a quick conversion on any recipe to how many servings you need.

Listing 6-2

```

100 REM  RECIPE CONVERTER
110 DIM I$(15),U$(15),N(15)
120 CALL CLEAR
130 PRINT TAB(6);"WELCOME TO THE":TAB(5);"RECIPE CONVER
    TER"::::
140 PRINT "THIS PROGRAM WILL HELP":"YOU IN THE KITCHEN
    BY":"CONVERTING RECIPES":::
150 PRINT "IT CAN BE USED TO EITHER":"INCREASE OR DECRE
    ASE THE":"NUMBER OF SERVINGS"::::
160 PRINT "  PRESS ANY KEY TO START":::
170 CALL KEY(3,K,S)
180 IF S=0 THEN 170
190 CALL CLEAR
  
```

cont. on next page

Listing 6-2—cont.

```

200 PRINT "TYPE THE NAME OF THE": "RECIPE";
210 INPUT N$
220 PRINT :: "HOW MANY INGREDIENTS ARE": "IN THIS RECIP
    E?":
230 INPUT "ENTER (1-15)? ": NI
240 PRINT
250 IF (NI<1)+(NI>15) THEN 230
260 FOR L=1 TO NI
270 CALL CLEAR
280 PRINT TAB(6); "INGREDIENT #"; L; ::
290 INPUT "NAME? ": I$(L)
300 PRINT ::
310 IF LEN(I$(L))<9 THEN 340
320 PRINT "TOO MANY LETTERS":
330 GOTO 290
340 INPUT "UNITS? ": U$(L)
350 PRINT ::
360 IF LEN(U$(L))<9 THEN 390
370 PRINT "TOO MANY LETTERS":
380 GOTO 340
390 INPUT "HOW MANY? ": N(L)
400 NEXT L
410 CALL CLEAR
420 PRINT :: "WHAT IS THE SUGGESTED NUMBER":
430 INPUT "OF SERVINGS? ": S1
440 IF S1<.1 THEN 420
450 PRINT :: "HOW MANY SERVINGS DO YOU":
460 INPUT "WANT TO MAKE? ": S2
470 IF S2<.1 THEN 450
480 CALL CLEAR
490 PRINT TAB(8); N$; ::
500 PRINT " FOR"; S2; "SERVINGS":
510 FOR L=1 TO NI
520 T=INT(N(L)/S1*S2*100)/100
530 IF T<100000 THEN 560
540 T$=" XXXXXXXX"
550 GOTO 580
560 T$="----- "&STR$(T)
570 T$=SEG$(T$,LEN(T$)-8,9)
580 PRINT SEG$(I$(L)&" -----",1,10); T$; " "; U$(L)
590 NEXT L
600 PRINT :: "ANOTHER RECIPE? (Y/N)":
610 CALL KEY(3,K,S)
620 IF K=89 THEN 120
630 IF K=78 THEN 650
640 GOTO 610
650 CALL CLEAR
660 END

```

Explanation of Program

- 110 Dimension arrays to hold ingredient names, units, and number needed in the recipe
- 120-180 Clear screen, print instructions, wait for input to start

190-210	Input recipe name
220-250	Input number of ingredients
260-400	Loop to input each ingredient name, its units, and amount needed for original recipe
410-440	Input suggested number of servings
450-470	Input desired number of servings
480-500	Clear screen, print recipe title, and print new number of servings
510-590	Loop to print information to screen using T and T\$ so that proper screen format is always maintained
600-640	User option to do another recipe
650-660	Program exit

Important Variables in Program

IS(15)	= Ingredient names
U\$(15)	= Unit names for each ingredient
N(15)	= Amount of ingredient in original recipe
N\$	= Name of recipe
NI	= Number of ingredients
L	= Loop variable
S1	= Suggested number of servings
S2	= Desired number of servings
T,T\$	= Temporary variables for proper screen format
K,S	= KEY subprogram variables

Sample Run of Program

```
TYPE THE NAME OF THE
RECIPE? CHICKEN SOUP

HOW MANY INGREDIENTS ARE
IN THIS RECIPE? 5

INGREDIENT # 1

NAME? CHICKEN
UNITS? POUNDS
HOW MANY? 1.5
```

Fig. 6-2.

Suggested Modifications and Projects

Try changing the program so you can store your recipes on cassette or disk for later reference.

MORTGAGE MACHINE

As we have seen before, the TI-99/4A is very good at performing mathematical calculations. Figuring out mortgage payments is a tedious job, but by using this program, you can quickly find out your monthly payment and interest paid over any twelve month period. If you happen to be working with very large amounts, the program will print out "XXXX.XX"s for the answer so the screen format will not be destroyed.

Listing 6-3

```

100 REM MORTGAGE MACHINE
110 DIM F(360,2)
120 CALL CLEAR
130 PRINT TAB(8);"WELCOME TO":TAB(4);"THE MORTGAGE MACH
    INE":::
140 PRINT "THIS PROGRAM WILL TELL YOU":"HOW MUCH PRINCIPAL AND":"INTEREST YOU PAY ON A LOAN"
150 PRINT "DURING ANY TWELVE MONTH":"PERIOD":"SIMPLY ENTER THE VALUES,":"AND THE TI-99/4A WILL DO"
160 PRINT "ALL THE WORK":::::"PRESS ANY KEY TO START":::::
170 CALL KEY(3,K,S)
180 IF S=0 THEN 170
190 CALL CLEAR
200 INPUT "AMOUNT OF LOAN? ":A
210 PRINT ::
220 IF A<1 THEN 200
230 INPUT "INTEREST RATE? ":I
240 PRINT ::
250 IF I<1 THEN 230
260 INPUT "LENGTH (YEARS)? ":Y
270 PRINT ::
280 IF (Y<=0)+(Y>30)THEN 260
290 CALL CLEAR
300 R=I/1200
310 M=Y*12
320 P=INT((A*(R/(1-(1/(1+R)^M)))+.005)*100)/100
330 PRINT :::"MONTHLY PAYMENT";P:::
340 PRINT "COMPUTING AMORTIZATION..."::::::::::
350 F(0,1)=A
360 FOR L=1 TO M
370 F(L,2)=INT((F(L-1,1)*R+.005)*100)/100
380 F(L,1)=F(L-1,1)-P+F(L,2)
390 NEXT L
400 PRINT "ENTER THE BEGINNING PAYMENT":"NUMBER OF ANY TWELVE MONTH":"PERIOD"
410 PRINT :::"TYPE (1-";STR$(M);")";
420 INPUT PD
430 IF (PD<1)+(PD>M)THEN 410
440 CALL CLEAR
450 PRINT "PMT-PRNCPL-INTERST--BALANCE":
460 TI=0
470 FOR L=PD TO PD+11

```

```

480 IF L>M THEN 700
490 TI=TI+F(L,2)
500 V=INT((P-F(L,2))*100)/100
510 V1=INT(F(L,2)*100)/100
520 V2=INT(F(L,1)*100)/100
530 IF V2>0 THEN 560
540 V=V+V2
550 V2=0
560 T=V
570 GOSUB 790
580 V$=T$
590 T=V1
600 GOSUB 790
610 V1$=T$
620 T=V2
630 FL=1
640 GOSUB 790
650 FL=0
660 V2$=T$
670 I$=" "&STR$(L)
680 I$=SEG$(I$,LEN(I$)-2,3)
690 PRINT I$;V$;V1$;V2$
700 NEXT L
710 PRINT "TOTAL INTEREST=";TI;:
720 PRINT "DO YOU WISH TO SEE INTEREST":"PAID FOR A DIF
    FERENT TWELVE":"MONTH PERIOD? (Y/N)":
730 CALL KEY(3,K,S)
740 IF K=78 THEN 1010
750 IF K=89 THEN 770
760 GOTO 730
770 CALL CLEAR
780 GOTO 400
790 IF FL=0 THEN 830
800 IF T<100000 THEN 860
810 T$=" XXXXX.XX"
820 GOTO 1000
830 IF T<10000 THEN 860
840 T$=" XXXX.XX"
850 GOTO 1000
860 T$=STR$(T)
870 T$=" "&T$
880 T$=SEG$(T$,LEN(T$)-6-FL,7+FL)
890 FOR L2=1 TO LEN(T$)
900 IF SEG$(T$,L2,1)="." THEN 960
910 NEXT L2
920 T$=T$&".00"
930 T$=SEG$(T$,4,7+FL)
940 T$=" "&T$
950 GOTO 1000
960 IF (L2=5)+((FL=1)*(L2=6))THEN 940
970 T$=T$&"0"
980 T$=SEG$(T$,2,7+FL)
990 GOTO 940
1000 RETURN
1010 CALL CLEAR
1020 PRINT "RUN PROGRAM AGAIN? (Y/N)":
1030 CALL KEY(3,K,S)
1040 IF K=78 THEN 1070

```

cont. on next page

Listing 6-3—cont.

```
1050 IF K=89 THEN 190
1060 GOTO 1030
1070 CALL CLEAR
1080 END
```

Explanation of Program

110	Dimension array to hold monthly balance and interest over period of loan
120-190	Clear screen, print instructions, wait for input to start, re-clear screen
200-220	Input amount of loan
230-250	Input interest rate
260-280	Input loan length in years
290	Clear screen
300	Convert interest rate for equations
310	Convert length to number of months
320-330	Compute and print monthly payment
340	Print message for user to read while amortization is calculated
350	Set month 0 balance to loan amount
360-390	Loop to compute monthly balance and interest payments
400-430	Input starting payment number of twelve month period
440-460	Print page heading, set total interest to zero
470	Start of loop to print out results
480	If last month of loan already done, then transfer out of loop
490	Keep running total of interest paid
500-550	Round off amounts to two decimal places
560-660	Set temporary variables, go to number formatting subroutine, and set V\$ variables to value of temporary variables
670-680	Format payment number for screen
690	Print amounts to screen using proper screen format
700	Complete loop, go back for next month of period
710	Print total interest paid
720-760	User option to see another twelve month period of same loan
770-780	If yes, clear screen, and go back to line 400
790-1000	Screen formatting routine—uses flag FL to allow one extra digit to print balance amount—takes temporary variable T and converts it to a string called T\$ that is 8 characters long with 2 digits to right of decimal point

1010-1060 User option to re-run program
 1070-1080 Program exit

Important Variables in Program

F(360,2) = Array to hold monthly balances and interest payments
 A = Amount of loan
 I = Interest rate
 R = Converted interest rate
 Y = Loan length in years
 M = Loan length in months
 P = Monthly loan payment
 PD = Starting payment number of twelve month period
 TI = Total interest paid in twelve month period
 L,L2 = Loop variables
 V,V1,V2 = Temporary variables for proper screen format
 V\$,V1\$,V2\$,I\$ = More variables for screen format
 T,T\$ = More variables for screen format
 FL = Flag to add one extra character to length of the balance amount
 K,S = KEY subprogram variables

Sample Run of Program

```

AMOUNT OF LOAN? 5000
INTEREST RATE? 14.5
LENGTH (IN YEARS)? 3

MONTHLY PAYMENT 172.10

COMPUTING AMORTIZATION...

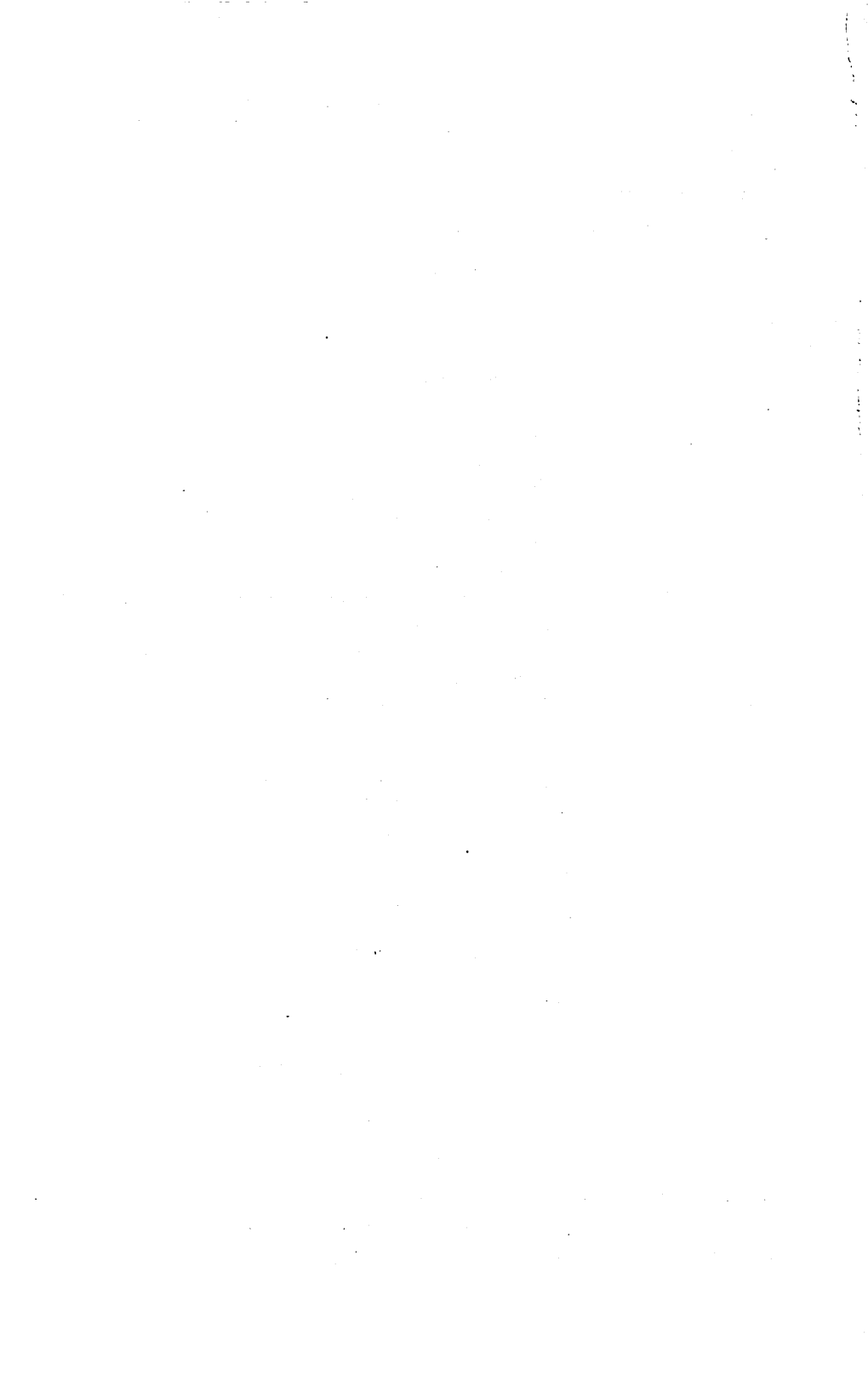
ENTER THE BEGINNING PAYMENT
NUMBER OF ANY TWELVE MONTH
PERIOD
TYPE (1-36)? 1

PMT-PRNCPL-INTERST—BALANCE
1 111.68 60.42 4888.32
2 113.03 59.07 4775.29
  
```

Fig. 6-3.

Suggested Modifications and Projects

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



Chapter 7

The Creation of a Program

Now that you have seen a number of programs for your TI-99/4A, perhaps you would like to understand just how a program is created. That is, how does a program evolve from a simple idea to a finished piece? In this last chapter, we will explore this process in some detail.

Basically, writing a program involves three steps. First, there is the conceptualizing and planning stage. Second, the program code must be written and typed into a computer. Last, the problems or “bugs” must be removed from the program. Following are some more detailed descriptions of these three important steps.

CONCEPTUALIZING AND PLANNING

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

Naturally, ideas that are not for 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 merge the power of the computer with people's needs, you are certain to get many fine ideas for programs.

Once you have an idea, there should be some planning. Flowcharts and a listings 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 how many points you will get for destroying certain aliens. If, on the other hand, you are writing a program to compose music, you should determine how many notes the program will hold, given the amount of available memory. Planning will save you a lot of time and trouble as you start to write your program.

WRITING A PROGRAM

Strange as it may seem, actually writing 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, you should consider the structure of the program carefully, use remarks to point to the important lines of the program, and keep a good record of all the variables. These steps will make the third part easier.

Sometimes it is easier to write programs in sections. That is, instead of sitting down and trying to write a whole program from the very beginning, many programmers write routines to do certain jobs. Then they use the routines together to create a program. For example, if you were writing a program that created special characters, you could write a routine that accepts input from the keyboard, a routine that takes data and forms characters on the screen, and a routine that saves character data. By breaking down big programs into several small programs, the job of creating computer software gets much easier. Plus, different routines from different programs can be combined to form new programs.

DEBUGGING AND FINISHING

Nearly every programmer would agree that debugging a program is the most exasperating, though sometimes gratifying, job that a programmer is asked to grapple with. The elusive bugs which find their way into nearly every program are often pesky, destructive, and hard to find. They will certainly eat away at your program and your sanity until you find them. Usually debugging the program takes longer than actually writing it.

Still, once you have finished the job of removing those bugs, a little polishing and finishing is in order. Getting a program exactly as you want it is rarely accomplished, since there always seem to be more features

available. Nevertheless, a programmer's finished product is always something to show off with pride.

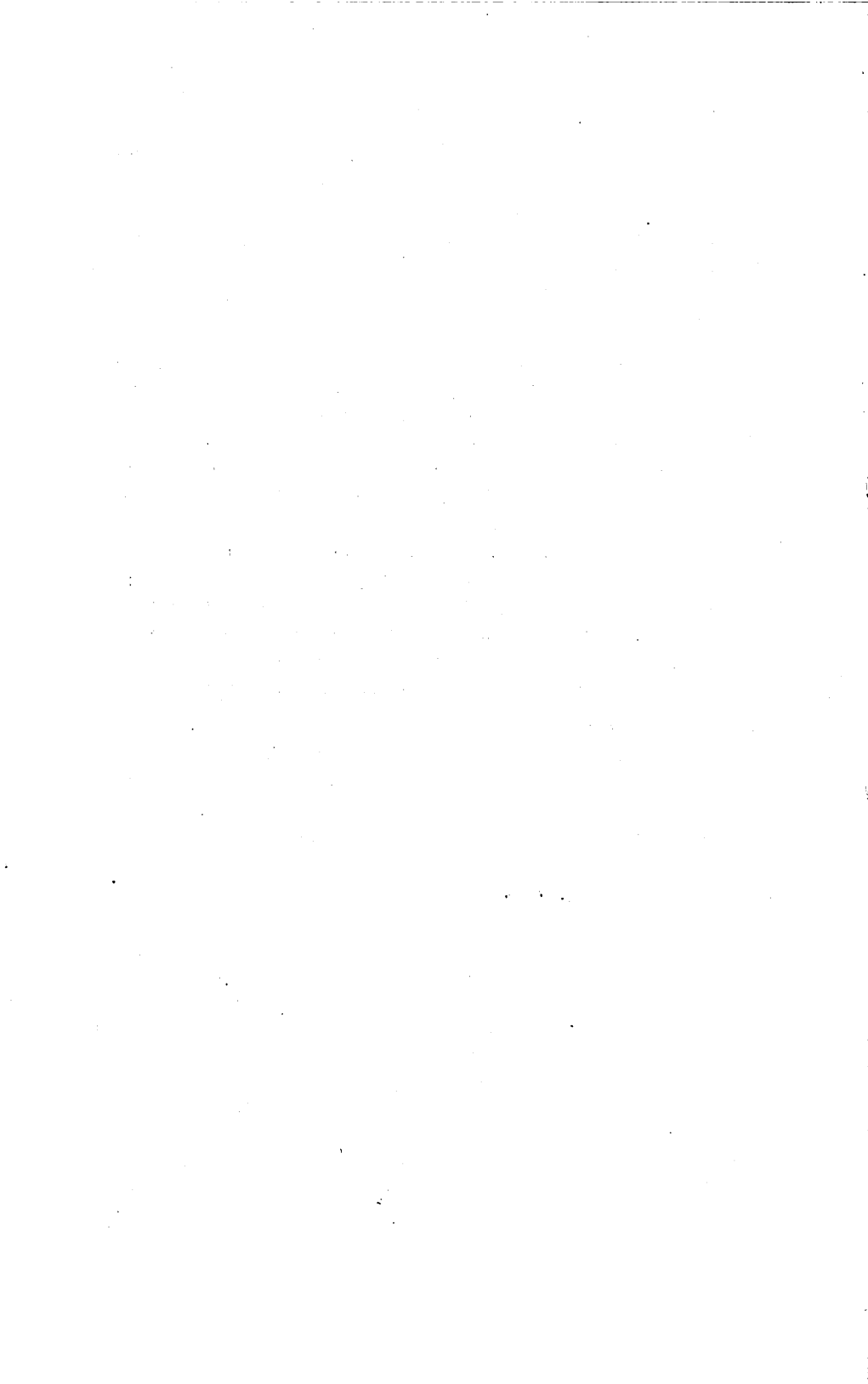
Because you have finished this book, you probably have a pretty good idea how to program in BASIC. In fact, if you understand most of the principles in this book, you have probably mastered BASIC and are ready to move onto a more advanced language. Before moving on, be sure that you have gained knowledge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

1. The subprograms of the TI-99/4A. These are extremely important and useful for just about every program application. Be sure to learn how they work so that you may use them with ease later.
2. Structure. If you see that your programs are not orderly or resemble a patchwork quilt, try to work on your structure. Before trying anything more complicated than BASIC, assure yourself that your programs are well-organized.
3. Know your computer. Understand how it can be expanded or modified, and know the basics of your computer's operation. Many programmers find themselves stalled because of minor problems like the printer or a disk drive not functioning properly. Knowing simple things like how to double check wire connections and how to clean a disk drive can save many hours of valuable programming time.

Once you feel you are ready to move on, there are several different ways to graduate from BASIC. There are many advanced languages available for the TI-99/4A. These are Extended BASIC, UCSD Pascal, PILOT, LOGO II, FORTH, and TMS9900 Assembly Language. Use of any of these languages will expand your computing horizons.

YOU AND YOUR COMPUTER

Since you are now a part of the computer revolution, take advantage of it. You certainly have the knowledge to do with your computer much of what you might wish. If you think your computer can help you run your house, give it a try. If you have a great game idea you believe can make a lot of money, go ahead and write it. If you think you can help others learn about programming, write your own book. Whatever you decide to do with you computer, remember that computers are made to help you. You should make every effort to use your computer to its full potential. The TI-99/4A can make your life easier, more enjoyable, and perhaps, even more profitable. Let your creativity be released through your machine, and keep on computing.



TI-99/4A™

BASIC Programs

- Gives the TI-99/4A 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

Howard W. Sams & Co., Inc.

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

\$9.95/22385

ISBN: 0-672-22385-6