BASIC Programs for Small Computers

Things To Do in 4K or Less

Over 30 programs using less than 4K of memory.
For the VIC-20,™ TRS-80 Model I® and Color
Computer,® TI-99/4A,® and other computers.

C. Regena

A **COMPUTE! Books** Publication

$12.95

BASIC Programs for Small Computers

Things To Do in 4K or Less

C. Regena

COMPUTE!™ Publications,Inc. abc
One of the ABC Publishing Companies
Greensboro, North Carolina

Printed in the United States of America

ISBN 0-942386-38-8

10 9 8 7 6 5 4 3 2 1

COMPUTE! Publications, Inc., Post Office Box 5406, Greensboro, NC 27403, (919) 275-9809, is one of the ABC publishing companies, and is not associated with any manufacturer of personal computers. VIC-20 is a trademark of Commodore Electronics Limited. TI-99/4 and TI-99/4A are trademarks of Texas Instruments, Inc. TRS-80 Color Computer, TRS-80 Model I, and MC-10 are trademarks of Tandy, Inc.

Contents

Acknowledgments

I want to offer special thanks to the staff of South Elementary School in Cedar City, Utah, who gave me the idea for this book. The school had purchased three computers, then received six hand-me-down computers from the high school. Although the teachers were excited, they asked, "What can we do with these computers?" This school, like many others, is anxious to give its students quality education, which includes computer experience.

In this book I offer some suggestions for what you *can* do with computers that have limited memory — and what you *can* do without having to spend any more money on peripherals.

I appreciate the help of my five children, Chery, Richard, Cindy, Bob, and Randy Whitelaw, who helped type the programs, then tested them to make sure they worked on several computers. My children and their teachers have continued to give me ideas for possible computer programs. I also acknowledge my husband, Chandler Whitelaw. He never complained as I outgrew my computer room and put computers in the kitchen, dining room, and bedroom.

Without the help of the COMPUTE! Book editors, this book would not be possible. I appreciate their suggestions and encouragement.

C. Regena

Foreword

If you own a VIC-20, a TRS-80 Color Computer, or another computer with limited memory, you probably know that it can be difficult to make it do complicated things. Memory restrictions simply prevent you from taking advantage of many popular commercial programs. Many useful programs seem to require more memory than your computer has.

However, there's a lot you can do with limited memory, and some of those huge software applications can be trimmed down or cut into miniprograms that run quite nicely on the computer you already have.

C. Regena is the popular author of the best-selling *Programmer's Reference Guide to the TI-99/4A* and *Programmer's Reference Guide to the Color Computer;* she also writes a column about the VIC-20 and Commodore 64 in *COMPUTE!'s Gazette* and one about the TI-99/4A in *COMPUTE! Magazine.* She has written and collected some of her best and most powerful programs for this book. All the programs take up less than 4K of memory in your computer—some of them considerably less. But they're all useful, from those which teach typing to those which test your knowledge of music or geography.

You'll see how to conserve memory in programs, letting you make the most of the memory your computer has. You'll even see how to translate programs from one computer's version of BASIC to another. Best of all, though, you'll have more than 100 programs to type in and use.

In fact, to make this book as useful as possible, many of the programs are accompanied by explanations of how the program works. Studying the programs and explanations will show you techniques that you can use as you write your own programs.

The programs range from games to loan and interest calculators, from teaching you the notes of a keyboard to plotting points on a graph. There are programs which turn your computer into a powerful calculator and programs which test your skill at solving mathematical word problems. You'll find uses for your computer again and again as you expand your library of programs.

Introduction

Many microcomputers used in homes and schools today have limited memory. Some companies started with a 4K model, then later added more memory and more capabilities. For example, Radio Shack attracted the new home consumer market with the TRS-80 Model I, a 4K black-and-white computer. Most dealers are now selling the Model IV (and others). Radio Shack's color computer line started with a 4K model and a 16K Extended BASIC model. That 4K Color Computer (affectionately called CoCo) has been replaced by the TRS-80 Micro Color Computer, Model MC-10, a smaller, compact microcomputer in the $100 range.

Commodore swept the home market with the 5K VIC-20 (actually 3583 bytes of free memory), an easy-to-use color computer with a full-size keyboard which now sells for under $100. Early Atari 400 and 800 owners probably purchased their computers with 8K. Sinclair owners could purchase a 1K or 2K computer or could make one from a kit.

Many people use their first computers to learn about computing, then upgrade to a more expensive computer with greater memory capacity. A lot of these early models are being shoved aside, given as hand-me-downs to another family member, or donated to schools. Many new computer owners are purchasing limited-memory computers because the price is attractive, and they figure *any* computer is better than *no* computer. Schools are purchasing low-end computers so they can get *more* computers and more students can have hands-on experience in class.

But you may have found that owning a computer is much like buying a house — you need to keep buying peripherals to go with it. In most cases, the computer peripherals cost more than the computer, and you may not be willing to add on those expenses. You want to make do with what you have.

But What Can You Do With 4K?

There is much you *can* do within 4K, and without buying extras. You still have a very powerful machine without having to add anything on. You *can* enjoy your computer just as it is.

I consider working within the memory limitations of a computer one of the challenges of that computer, just like having eight colors, a certain size screen, or particular graphics commands. You can't expect your microcomputer to do everything a mainframe computer can do — you didn't have to pay as much for your little computer. But it can do a few things that a big computer cannot do.

I hope these programs and ideas will be helpful to you and will whet your appetite so you can develop your own programs within the limitations of your computer. Although most of these programs are educational, I have tried to include a variety of subjects so you can get an idea of the many different applications for home computers.

I assume you have a computer, a television or monitor to see what you're doing, and a cassette recorder with blank tapes to save your programs, plus all the necessary cables to connect everything. You don't need to buy anything else to run the programs in this book.

The first section (Chapters 1-3) consists of programs that contain no graphics or sound statements. They are written in translatable BASIC, so you should be able to use them with any computer that uses BASIC. It was impossible to test the programs with *all* microcomputers, so explanations are included in case you need to make minor translations. Computers with larger memory capabilities may also use these programs.

The second section (Chapters 4-9) consists of programs for specific computers — TRS-80 Model I, TRS-80 Color Computer, MC-10 Micro Color Computer, unexpanded VIC-20, TI-99/4, and TI-99/4A. (Although the TI computers do not have the memory limitations of the other machines, program versions have been included. They can usually be entered in a limited amount of time, and are useful to any computer owner.)

The listings can even give you an idea of how to make these programs work on other computers.

Chapter 1
Conserving Memory

Chapter 1
Conserving Memory

If you program efficiently, you can really fit quite a bit into 4K or less. Here are some suggestions for memory-saving programming techniques.

• **Delete REMark statements.** As you are programming, REMark statements help to document your program so you can keep track of various sections of the program. However, if you need to conserve memory, the first and easiest step could be to remove all the REM statements.

• **Use short variable names.** Many computers recognize only the first two letters of the name anyway, so the rest take up valuable space. If you can use variables with one letter, you conserve even more memory than by using two-letter names. For example, REDSCORE and BLUESCORE may be shortened to *R* and *B*.

• **Repeat variable names where possible.** For example, in the beginning of the program you may read in ten variables:

```
1Ø FOR I=1 TO 1Ø
2Ø READ V
3Ø PRINT V,V*V
4Ø NEXT I
```

Later in the program you have another counter loop. Use the variable *I* again:

```
6ØØ FOR I=1 TO 1ØØØ
61Ø NEXT I
```

Many computers allow you to leave off the index in a NEXT statement:

```
5Ø FOR J=1 TO 5Ø
6Ø NEXT
```

• **Use FOR-NEXT or GOSUB.** Anytime you have repetitious code, see if a FOR-NEXT loop or a GOSUB procedure will work.

• **Plan all GOTO logic.** If you have many GOTO statements in your program, it may be much more efficient to rearrange your program.

• **Use the DRAW command, if your computer allows, to create graphic displays.** Subroutines or FOR-NEXT loops can save lines when used with the DRAW command.

• **DIMension wisely.** If you need to use an array with 25 names, you waste valuable memory if you arbitrarily DIMension for 40 names. Many computers automatically DIMension 11 elements whenever you use a subscripted number without a previous

DIMension statement. If you really need fewer subscripts, be sure to use the DIMension statement to save memory. Also remember to use the zero element — you don't need to start with A(1).

• **READ and DATA.** If you have several assignment statements, a DATA and READ procedure may be more efficient:

```
10 A=1
20 B=3
30 C=2
40 D=5
50 E=7
```

is equivalent to

```
10 READ A,B,C,D,E
20 DATA 1,3,2,5,7
```

• **Use ON-GOTO or ON-GOSUB instead of several logical IF-THEN statements.** For example:

```
10 IF I=1 THEN 640
20 IF I=2 THEN 730
30 IF I=3 THEN 870
40 IF I=4 THEN 930
```

is equivalent to

```
10 ON I GOTO 640,730,870,930
```

You may first need to make sure that variable I is a positive integer less than 5.

• **Leave out spaces if your computer allows it.** The statements are harder to read, but use less space. If a BASIC word follows a variable name, a space may be required between the variable and the next word.

• **If your computer allows it, use multistatement lines.** The most common statement separator is the colon. Some computers use the slash. Use abbreviations, if your computer allows them, to crowd more on each line. For example, many computers use ? for PRINT and ! or ' (apostrophe) for REMark. Abbreviations alone will not save memory, but by using them you can put more statements and commands on a line, saving a small number of bytes otherwise used for additional line numbers. You need to be careful when you're combining statements if an IF-THEN statement is involved or if any program transfer statements go to one of the statements to be combined.

• **Use low line numbers.** Often during program development it is wise to number your statements by tens so you can later insert statements. I often plan different sections of the program starting with the thousands. After a program is complete, I save memory by renumbering the lines — perhaps using 10, 20, 30, 40, and so on, or, even better, 1, 2, 3, 4, etc.

Before you begin entering any of the programs in this book, take a look at Appendix A, "A Beginner's Guide to Typing In Programs," and Appendix B, "How to Type In Programs." These will help you understand how to enter the listings in this book into your own computer.

Nouns

Computers: VIC-20; TRS-80 Model I; TRS-80 16K Color Computer; MC-10; TI-99/4; TI-99/4A

Following is an example of how to conserve memory by combining lines. The first program, "Nouns," works as is on a VIC-20. To add another section of DATA or to add graphics and sound, this program needs to take up less memory.

Combining commands for one statement number usually saves five bytes per line (depending on your computer). In conserving memory for Nouns, I went through the listing and marked every line that was referred to by another command — such as GOTO, GOSUB, or IF-THEN, plus every statement after an IF-THEN statement. I had to make sure those lines contained the correctly referenced line numbers.

The second listing of Nouns shows how the lines were combined. For instance, notice that lines 4 through 30 in the VIC program were combined into lines 2 and 3 in the TRS-80 version. Lines 32 through 48 in the VIC version were combined into two lines, lines 4 and 5, in the TRS-80 program. Commands such as GOSUB were carefully modified to reflect the new line numbers in the second, crunched version of the program. Line 48 in the VIC program, for example, read GOSUB200. This is the last expression in line 5 of the TRS-80 program, but it was changed to GOSUB 22 because of the new line numbering of this shorter listing. Combining lines in this way saves memory, but you need to be careful to prevent errors in line numbers.

The second listing works on the TRS-80 Model I, TRS-80 16K Color Computer, and MC-10 computer. The third listing is the TI-99/4 or TI-99/4A version. Note that TI BASIC does not allow combining commands on one line, but you can combine PRINT statements.

Nouns presents a quiz to identify nouns. A word is chosen randomly and printed on the screen. The user presses *1* if the word is a person, *2* if the word is a place, *3* if the word is a thing, or *4* if the word is not a noun. After ten words a score is given. Since the answers are multiple-choice, the program does not continue until the correct number is chosen. The score is based on correct answers for the first response.

Program 1-1. Nouns

VIC-20

```
4 PRINT"{CLR}"
6 PRINTTAB(5);"*********"
8 PRINTTAB(5);"* NOUNS *"
1Ø PRINTTAB(5);"*********"
12 PRINT
14 PRINT"A NOUN IS A WORD THAT"
16 PRINT"IS A PERSON,{2 SPACES}PLACE,"
18 PRINT"OR THING."
2Ø PRINT
22 PRINT"YOU WILL BE SHOWN A"
24 PRINT"WORD.{2 SPACES}PRESS THE"
26 PRINT"RIGHT NUMBER FOR"
28 PRINT"PERSON, PLACE, THING,"
3Ø PRINT"OR NOT A NOUN."
32 DIMW$(1ØØ),W(1ØØ)
34 FORC=ØTO1ØØ
36 READW$(C),W(C)
38 NEXT C
4Ø N$(1)="PERSON"
42 N$(2)="PLACE"
44 N$(3)="THING"
46 N$(4)="NOT A NOUN"
48 GOSUB2ØØ
49 D=Ø
5Ø SC=Ø
52 FORI=1TO1Ø
54 PRINT"{CLR}"
56 FL=Ø
58 R=INT(RND(Ø)*1Ø1)
6Ø IFW(R)=ØTHEN58
62 PRINT"{2 SPACES}** ";W$(R);" **"
64 PRINT
66 FORC=1TO4
68 PRINTC;N$(C)
7Ø NEXTC
72 PRINT
74 PRINT"?"
76 GETE$
78 IFE$=""THEN76
8Ø E=ASC(E$)-48
```

```
82 IF(E<1)+(E>4)THEN76
84 PRINTN$(E)
86 IFE=W(R)THEN96
88 PRINT"SORRY, TRY AGAIN."
90 PRINT
92 FL=1
94 GOTO76
96 PRINT"CORRECT!"
98 IF FL=1 THEN 104
100 SC=SC+1
102 W(R)=0
104 GOSUB200
106 NEXT I
107 PRINT"{CLR}"
108 PRINTTAB(5);"NOUNS"
110 PRINT
112 PRINT"SCORE:"
114 PRINTSC;"CORRECT"
116 PRINTTAB(5);"OUT OF 10 WORDS"
118 PRINT
120 PRINT
121 D=D+1:IFD>8THEN300
122 PRINT"PRESS 1 TRY AGAIN"
124 PRINTTAB(6);"2 END"
126 GETE$
128 IF E$="1"THEN 50
130 IF E$="2"THEN300
132 GOTO126
200 PRINT
202 PRINT"PRESS RETURN"
204 GETE$
206 IFE$=""THEN204
208 IFASC(E$)<>13 THEN 204
210 RETURN
220 DATAMAN,1,BOY,1,DOG,3,TEXAS,2,ROGER,1,WALL,3,W
    AS,4,UTAH,2,IDAHO,2,CHAIR,3,HAT,3
222 DATAHAND,3,GIRL,1,LADY,1,JILL,1,CINDY,1,JOE,1,
    BOB,1,COWBOY,1,TOWN,2,SKY,2,HOME,2
224 DATARIVER,3,HILL,3,GLASS,3,CUP,3,DISH,3,WENT,4
    ,BEFORE,4,AFTER,4,WHEN,4,NOW,4,THEN,4
226 DATADRESS,3,PANTS,3,SHIRT,3,SOCKS,3,SHOES,3,MA
    P,3,FLAG,3,MOTHER,1,FATHER,1,DAD,1
228 DATASISTER,1,BROTHER,1,BABY,1,DESK,3,TABLE,3,W
    ATER,3,FLOWER,3,OHIO,2,IOWA,2,MAINE,2
230 DATABOOK,3,PAPER,3,PENCIL,3,BOX,3,TREE,3,BEAR,
    3,FOX,3,BALL,3,QUIET,4,QUICK,4
232 DATABLUE,4,GREEN,4,BEADS,3,ANNE,1,ABLE,4,ABOUT
    ,4,ABOVE,4,ACROSS,4,ACTOR,1,AGAIN,4
234 DATABACON,3,BAKE,4,BIRD,3,BANANA,3,BEACH,2,PAR
    IS,2,LONDON,2,JAPAN,2,MEXICO,2
```

```
236 DATAAFRICA,2,CANADA,2,CAR,3,NEVER,4,WALKED,4,J
    UMPED,4,OF,4,ARE,4,BY,4,ATE,4,BIG,4
238 DATAINTO,4,HAPPY,4,SAD,4,ALICE,1,SMALL,4,EGYPT
    ,2,TEACHER,1,IS,4
3ØØ END
```

Program 1-2. Nouns

TRS-80 Model I and Color Computer; MC-10

```
2 CLS:PRINTTAB(12);"*********":PRINTTAB(12);"* NOU
  NS *":PRINTTAR(12);"*********":PRINT:PRINT"A NOU
  N IS A WORD THAT IS A":PRINT"PERSON, PLACE, OR T
  HING."
3 PRINT:PRINT"YOU WILL BE SHOWN A WORD.":PRINT"PRE
  SS THE RIGHT NUMBER FOR":PRINT"PERSON, PLACE, TH
  ING, OR":PRINT"NOT A NOUN."
4 DIM W$(1ØØ),W(1ØØ)
5 FORC=ØTO1ØØ:READW$(C),W(C):NEXTC:N$(1)="PERSON":
  N$(2)="PLACE":N$(3)="THING":N$(4)="NOT A NOUN":G
  OSUB22
6 S=Ø:FORI=1TO1Ø
7 CLS:F=Ø
8 R=RND(1ØØ):IF W(R)=Ø THEN 8
9 PRINT"   ** ";W$(R);" **":PRINT:FORC=1TO4:PRINTC;
  N$(C):NEXT:PRINT:PRINT"?"
1Ø E$=INKEY$:IFE$=""THEN1Ø
11 E=ASC(E$)-48:IF(E<1)+(E>4)THEN1Ø
12 PRINTN$(E):IFE=W(R)THEN14
13 PRINT"SORRY, TRY AGAIN.":PRINT:F=1:GOTO1Ø
14 PRINT"CORRECT!":IFF=1THEN16
15 S=S+1:W(R)=Ø
16 GOSUB22:NEXT
17 CLS:PRINTTAB(5);"NOUNS":PRINT:PRINT"SCORE:":PRI
   NTS;"CORRECT":PRINTTAB(5);"OUT OF 1Ø WORDS":PRI
   NT:PRINT:D=D+1:IFD>8THEN33
18 PRINT"PRESS 1 TRY AGAIN":PRINTTAB(6);"2 END"
19 E$=INKEY$:IFE$="1"THEN6
2Ø IFE$="2"THEN33
21 GOTO19
22 PRINT:PRINT"PRESS ENTER"
23 E$=INKEY$:IFE$=""THEN23
24 IF ASC(E$)<>13THEN23
25 RETURN
26 DATAMAN,1,BOY,1,DOG,3,TEXAS,2,ROGER,1,WALL,3,WA
   S,4,UTAH,2,IDAHO,2,CHAIR,3,HAT,3,HAND,3,GIRL,1,
   LADY,1,JILL,1,CINDY,1,JOE,1
27 DATABOB,1,COWBOY,1,TOWN,2,SKY,2,HOME,2,RIVER,3,
   HILL,3,GLASS,3,CUP,3,DISH,3,WENT,4,BEFORE,4,AFT
   ER,4,WHEN,4,NOW,4,THEN,4
```

```
28 DATADRESS,3,PANTS,3,SHIRT,3,SOCKS,3,SHOES,3,MAP
   ,3,FLAG,3,MOTHER,1,FATHER,1,DAD,1,SISTER,1,BROT
   HER,1,BABY,1,DESK,3,TABLE,3
29 DATAWATER,3,FLOWER,3,OHIO,2,IOWA,2,MAINE,2,BOOK
   ,3,PAPER,3,PENCIL,3,BOX,3,TREE,3,BEAR,3,FOX,3,B
   ALL,3,QUIET,4,QUICK,4
3Ø DATABLUE,4,GREEN,4,BEADS,3,ANNE,1,ABLE,4,ABOUT,
   4,ABOVE,4,ACROSS,4,ACTOR,1,AGAIN,4,BACON,3,BAKE
   ,4,BIRD,3,BANANA,3,BEACH,2
31 DATAPARIS,2,LONDON,2,JAPAN,2,MEXICO,2,AFRICA,2,
   CANADA,2,CAR,3,NEVER,4,WALKED,4,JUMPED,4,OF,4,A
   RE,4,BY,4,ATE,4,BIG,4
32 DATAINTO,4,HAPPY,4,SAD,4,ALICE,1,SMALL,4,EGYPT,
   2,TEACHER,1,IS,4
33 END
```

Program 1-3. Nouns

TI-99/4A

```
11Ø CALL CLEAR
12Ø PRINT TAB(5);"*********":TAB(5);"* NOUNS *":TA
    B(5);"*********"
13Ø PRINT :"YOU WILL BE SHOWN A WORD.{3 SPACES}PRE
    SS THE RIGHT NUMBER FOR  PERSON, PLACE, THING,
     OR{4 SPACES}NOT A NOUN."
14Ø DIM W$(1ØØ),W(1ØØ)
15Ø FOR C=Ø TO 1ØØ
16Ø READ W$(C),W(C)
17Ø NEXT C
18Ø N$(1)="PERSON"
19Ø N$(2)="PLACE"
2ØØ N$(3)="THING"
21Ø N$(4)="NOT A NOUN"
22Ø GOSUB 57Ø
23Ø D=Ø
24Ø SC=Ø
25Ø FOR I=1 TO 1Ø
26Ø CALL CLEAR
27Ø FL=Ø
28Ø RANDOMIZE
29Ø R=INT(1Ø1*RND)
3ØØ IF W(R)=Ø THEN 29Ø
31Ø PRINT "   ** ";W$(R);" **"::
32Ø FOR C=1 TO 4
33Ø PRINT C;N$(C)
34Ø NEXT C
35Ø PRINT :"?"
36Ø CALL KEY(Ø,K,S)
37Ø IF (K<49)+(K>52)THEN 36Ø
38Ø PRINT :N$(K-48)
```

```
390 IF K-48=W(R)THEN 430
400 PRINT "SORRY, TRY AGAIN."::
410 FL=1
420 GOTO 360
430 PRINT "CORRECT!"
440 IF FL=1 THEN 470
450 SC=SC+1
460 W(R)=0
470 GOSUB 570
480 NEXT I
490 CALL CLEAR
500 PRINT TAB(5);"NOUNS"::"SCORE:":SC;"CORRECT":TA
    B(5);"OUT OF 10 WORDS":::
510 D=D+1
520 IF D>8 THEN 690
530 PRINT "PRESS 1 TRY AGAIN":TAB(7);"2 END"
540 CALL KEY(0,K,S)
550 IF K=49 THEN 240
560 IF K=50 THEN 690 ELSE 540
570 PRINT :"PRESS ENTER"
580 CALL KEY(0,K,S)
590 IF K<>13 THEN 580
600 RETURN
610 DATA MAN,1,BOY,1,DOG,3,TEXAS,2,ROGER,1,WALL,3,
    WAS,4,UTAH,2,IDAHO,2,CHAIR,3,HAT,3,HAND,3,GIRL
    ,1,LADY,1
620 DATA JILL,1,CINDY,1,JOE,1,BOB,1,COWBOY,1,TOWN,
    2,SKY,2,HOME,2,RIVER,3,HILL,3,GLASS,3,CUP,3,DI
    SH,3,WENT,4
630 DATA BEFORE,4,AFTER,4,WHEN,4,NOW,4,THEN,4,DRES
    S,3,PANTS,3,SHIRT,3,SOCKS,3,SHOES,3,MAP,3,FLAG
    ,3,MOTHER,1
640 DATA FATHER,1,DAD,1,SISTER,1,BROTHER,1,BABY,1,
    DESK,3,TABLE,3,WATER,3,FLOWER,3,OHIO,2,IOWA,2,
    MAINE,2
650 DATA BOOK,3,PAPER,3,PENCIL,3,BOX,3,TREE,3,BEAR
    ,3,FOX,3,BALL,3,QUIET,4,QUICK,4,BLUE,4,GREEN,4
    ,BEADS,3
660 DATA ANNE,1,ABLE,4,ABOUT,4,ABOVE,4,ACROSS,4,AC
    TOR,1,AGAIN,4,BACON,3,BAKE,4,BIRD,3,BANANA,3,B
    EACH,2
670 DATA PARIS,2,LONDON,2,JAPAN,2,MEXICO,2,AFRICA,
    2,CANADA,2,CAR,3,NEVER,4,WALKED,4,JUMPED,4,OF,
    4,ARE,4
680 DATA BY,4,ATE,4,BIG,4,INTO,4,HAPPY,4,SAD,4,ALI
    CE,1,SMALL,4,EGYPT,2,TEACHER,1,IS,4
690 END
```

Chapter 2
Translating BASIC

Chapter 2
Translating BASIC

Translating programs from one version of BASIC to another computer's BASIC can become a skill of its own. Some BASIC programs are translatable, but others aren't. It all depends on how complex the program is and what special features of a computer it uses.

Many of the programs in this book are computer-specific. In other words, the program will work on only one computer, using that machine's version of BASIC. Others, however, can be translated from one computer to another. The short programs listed in this chapter, and in Chapter 3, are good examples of "translatable" programs. All of these programs can be entered into almost any home computer which uses BASIC. This is possible because the programs are simple and use no special graphic or sound features peculiar to only one or two computers.

Before each program in this book is a note indicating which computers can use the program as it's listed. Sometimes this note will list various computers. Other times it will simply read *translatable*. When you see this word, you should be able to type the program in and RUN it on your own computer, no matter what model it is. All you have to do is look the program over carefully and make a few simple changes. The most important commands that must be changed from one computer to another are listed below:

To clear the screen:

```
TRS-80, MC-10: CLS
VIC-20, 64, PET: PRINT "{CLR}" (SHIFT and CLR/HOME)
TI-99/4, TI-99/4A: CALL CLEAR
Apple: HOME
Atari: PRINT "   " (ESC CTRL CLR)
Timex/Sinclair: CLS
```

Random number from 1 to A:

```
TRS-80, MC-10: RND(A)
TRS-80, Extended BASIC: R=RND(-TIMER):R=RND(A)
VIC-20, 64, PET: INT(A*RND(0)+1)
TI-99/4, TI-99/4A: RANDOMIZE
                   INT(A*RND+1)
Apple: RND(A)+1
Apple IIe: INT(A*RND(1)+1)
Atari: INT(A*RND(1)+1)
Timex/Sinclair: INT(A*RND+1)
```

To detect a key being pressed:

```
TRS-80, MC-10: E$=INKEY$
VIC-20, 64, PET: GET E$
TI-99/4, TI-99/4A: CALL KEY(0,K,S)
Apple: GET E$
Atari: E=PEEK(764)
Timex/Sinclair: LET E$=INKEY$
```

Not all of the computers referred to here have programs specific to them in the book, but I've included the commands to help you as you translate the programs in this chapter and in Chapter 3 to your own computer.

Besides these changes, you may want to add graphics and sound to enhance the program as it is printed here. Commands involving string variables may need to be changed as well. You will probably also want to adjust the PRINT statements to reflect the number of columns your own monitor screen has.

If the note listing the computers a program can run on indicates it works for the TRS-80 Color Computer, it will work for both the 4K and 16K models, as well as with the 16K Extended BASIC computer.

Some of the VIC-20 programs in this book have a short line before the listing, stating that lines must be entered using abbreviations. If you don't abbreviate the keywords in these lines, the program will not work. Be sure to use abbreviations for all BASIC keywords when you type in the lines marked with an asterisk (*). If you don't, those programs probably won't run; if they do, they won't work correctly. Most of the abbreviations are simply the first letter, then SHIFT, and then the second letter of the keyword. Refer to Appendix C, "How to Abbreviate Keywords," for a complete list of BASIC keywords and their abbreviations on the VIC-20.

If you need to edit a line, be sure the line also contains abbreviations when you are through. For example, if you LIST a line that you previously typed with abbreviations, the full keyword is displayed. If you then edit that line on the screen, part of it may be cut off, since the VIC-20 will show a maximum of only 87 characters on a line. Your program may not run if this happens. You need to either retype the entire line or replace all the keywords in the line you LISTed with abbreviations. This is especially important with all lines marked with an asterisk (*) in the VIC-20 program listings.

Roman Numerals

Computers: VIC-20; TRS-80 Color Computer; TRS-80 Model I; MC-10; TI-99/4; TI-99/4A

"Roman Numerals" illustrates how a standard BASIC program can be adapted for other computers. The first listing works exactly as is on the TRS-80 Model I, TRS-80 Color Computer, and MC-10. The second listing shows how the program was changed for the VIC-20. The commands for clearing the screen, choosing a random number, and detecting a key pressed were changed, and some of the PRINT statements were adapted for the 22-column screen and color capabilities of the VIC-20. The third listing is the TI-99/4 or TI-99/4A version.

Here's how Roman Numerals works:

Line	Function
110–150	Reads from DATA the Roman numeral values for the Arabic numbers in hundreds H$, tens T$, and ones S$.
160	Clear the screen.
170	Initialize R$ to be the null string.
180–190	Choose a random number from 1 to 1999 and set NN equal to N.
200–220	If the number starts with 1000, set R$ to "M" and subtract 1000 from N.
230–260	Calculate the hundreds value for the Roman numeral.
270–300	Calculate the tens value for the Roman numeral.
310–320	Calculate the ones value for the Roman numeral.
330–340	PRINT the Roman numeral, R$.
350–380	Ask for the corresponding number as INPUT.
390	Test user's answer.
400	PRINT correct answer if user's answer is incorrect.
410–420	If user's answer is incorrect, PRINT "PRESS ENTER" message, then return for another problem.
430–440	PRINT message for correct answer.
450–490	PRINT option for another problem and branch appropriately.
500	END.

Program 2-1. Roman Numerals

TRS-80 Color Computer; MC-10; TRS-80 Model I

```
11Ø FOR I=1 TO 9
12Ø READ H$(I),T$(I),S$(I)
13Ø NEXT I
```

```
140 DATA C,X,I,CC,XX,II,CCC,XXX,III,CD,XL,IV,D,L,V
150 DATA DC,LX,VI,DCC,LXX,VII,DCCC,LXXX,VIII,CM,XC
    ,IX
160 CLS
170 R$=""
180 N=RND(1999)
190 NN=N
200 IF N<1000 THEN 230
210 R$="M"
220 N=N-1000
230 IF N<100 THEN 270
240 NR=INT(N/100)
250 R$=R$+H$(NR)
260 N=N-NR*100
270 IF N<10 THEN 310
280 NR=INT(N/10)
290 R$=R$+T$(NR)
300 N=N-NR*10
310 IF N=0 THEN 330
320 R$=R$+S$(N)
330 PRINT "GIVEN THE ROMAN NUMERAL"
340 PRINT R$
350 PRINT "WHAT IS THE CORRESPONDING"
360 PRINT "NUMBER?"
370 INPUT ANS
380 PRINT
390 IF ANS=NN THEN 430
400 PRINT "THE NUMBER IS";NN
410 INPUT "PRESS <ENTER>";E$
420 GOTO 160
430 PRINT "CORRECT!"
440 PRINT
450 PRINT "ANOTHER PROBLEM?"
460 PRINT "(Y/N)"
470 E$=INKEY$
480 IF E$="Y" THEN 160
490 IF E$<>"N" THEN 470
500 END
```

Program 2-2. Roman Numerals

VIC-20

```
110 FOR I=1 TO 9
120 READ H$(I),T$(I),S$(I)
130 NEXT I
140 DATA C,X,I,CC,XX,II,CCC,XXX,III,CD,XL,IV,D,L,V
150 DATA DC,LX,VI,DCC,LXX,VII,DCCC,LXXX,VIII,CM,XC
    ,IX
160 PRINT"{CLR}"
170 R$=""
```

```
180 N=INT(1999*RND(Ø))+1
190 NN=N
200 IF N<1ØØØ THEN 23Ø
210 R$="M"
22Ø N=N-1ØØØ
23Ø IF N<1ØØ THEN 27Ø
24Ø NR=INT(N/1ØØ)
25Ø R$=R$+H$(NR)
26Ø N=N-NR*1ØØ
27Ø IF N<1Ø THEN 31Ø
28Ø NR=INT(N/1Ø)
29Ø R$=R$+T$(NR)
3ØØ N=N-NR*1Ø
31Ø IF N=Ø THEN 33Ø
32Ø R$=R$+S$(N)
33Ø PRINT"{BLU}GIVEN THE ROMAN":PRINT"NUMERAL{RED}
    "
34Ø PRINT R$
35Ø PRINT:PRINT"{BLU}WHAT IS THE"
36Ø PRINT"CORRESPONDING NUMBER?{BLK}"
37Ø INPUT ANS
38Ø PRINT
39Ø IF ANS=NN THEN 43Ø
4ØØ PRINT"{BLU}THE NUMBER IS";NN
41Ø PRINT:INPUT"{GRN}PRESS <RETURN>";E$
42Ø GOTO 16Ø
43Ø PRINT"{BLU}CORRECT!"
44Ø PRINT
45Ø PRINT "ANOTHER PROBLEM?"
46Ø PRINT "(Y/N)"
47Ø GET E$
48Ø IF E$="Y" THEN 16Ø
49Ø IF E$<>"N" THEN 47Ø
5ØØ END
```

Program 2-3. Roman Numerals

TI-99/4; TI-99/4A

```
11Ø FOR I=1 TO 9
12Ø READ H$(I),T$(I),S$(I)
13Ø NEXT I
14Ø DATA C,X,I,CC,XX,II,CCC,XXX,III,CD,XL,IV,D,L,V
15Ø DATA DC,LX,VI,DCC,LXX,VII,DCCC,LXXX,VIII,CM,XC
    ,IX
16Ø CALL CLEAR
17Ø R$=""
175 RANDOMIZE
18Ø N=INT(1999*RND)+1
19Ø NN=N
2ØØ IF N<1ØØØ THEN 23Ø
```

```
21Ø R$="M"
22Ø N=N-1ØØØ
23Ø IF N<1ØØ THEN 27Ø
24Ø NR=INT(N/1ØØ)
25Ø R$=R$&H$(NR)
26Ø N=N-NR*1ØØ
27Ø IF N<1Ø THEN 31Ø
28Ø NR=INT(N/1Ø)
29Ø R$=R$&T$(NR)
3ØØ N=N-NR*1Ø
31Ø IF N=Ø THEN 33Ø
32Ø R$=R$&S$(N)
33Ø PRINT "GIVEN THE ROMAN NUMERAL"
34Ø PRINT R$
35Ø PRINT :"WHAT IS THE CORRESPONDING"
36Ø PRINT "NUMBER?"
37Ø INPUT ANS
38Ø PRINT
39Ø IF ANS=NN THEN 43Ø
4ØØ PRINT "THE NUMBER IS";NN
41Ø INPUT "PRESS <ENTER>":E$
42Ø GOTO 16Ø
43Ø PRINT "CORRECT!"
44Ø PRINT
45Ø PRINT "ANOTHER PROBLEM?"
46Ø PRINT "(Y/N)"
47Ø CALL KEY(Ø,K,S)
48Ø IF K=89 THEN 16Ø
49Ø IF K<>78 THEN 47Ø
5ØØ END
```

Adverbs

Computers: Translatable

Here is a program written in portable or translatable BASIC. You should be able to translate this program to the version of BASIC your computer uses without much difficulty. You can probably tell which lines to change for your particular computer. The listing works as is on the TRS-80 Model I, TRS-80 Color Computer, and MC-10.

Here's how Adverbs works:

Line	Function
110–200	Clear screen; PRINT instructions.
210–260	READ words in as data (article, noun, verb, and adverb for nine of each type of word).
270–320	Wait for user to press ENTER.

330	Clear screen.
340	Initialize score.
350	Perform quiz ten times.
360	Clear screen.
370–400	Randomly choose words for sentence.
410–470	Randomly choose order for sentence and print sentence.
480–500	Enter answer.
510–530	If answer is incorrect, PRINT correct answer.
540–550	If answer is correct, PRINT message and increment score.
560–610	Wait for user to press ENTER for next problem.
620–680	Clear screen and PRINT score.
690–720	PRINT option to try again and branch appropriately.
730–740	Clear screen and END.

Program 2-4. Adverbs

Translatable

```
11Ø CLS
12Ø PRINT TAB(8);"***********"
13Ø PRINT TAB(8);"* ADVERBS *"
14Ø PRINT TAB(8);"***********"
15Ø PRINT
16Ø PRINT
17Ø PRINT "YOU WILL BE SHOWN A SENTENCE."
18Ø PRINT
19Ø PRINT "TYPE THE ADVERB"
2ØØ PRINT "THEN PRESS <ENTER>."
21Ø FOR C=Ø TO 9
22Ø READ A$(C),B$(C),C$(C),D$(C)
23Ø NEXT C
24Ø DATA THE,CAT,CRAWLED,QUICKLY,A,DOG,JUMPED,QUIE
    TLY,MY,DEER,RAN,HAPPILY,YOUR,COW,LOPED,SLYLY
25Ø DATA HIS,FOX,WIGGLED,SLOWLY,HER,WOLF,GALLOPED,
    JOYFULLY,ITS,BOY,SPED,RAPIDLY,OUR,GIRL,CREEPED
    ,SILENTLY
26Ø DATA THAT,BUG,HURRIED,CALMLY,ONE,BEAR,MOVED,SW
    IFTLY
27Ø PRINT
28Ø PRINT
29Ø PRINT "PRESS <ENTER> TO START."
3ØØ E$=INKEY$
31Ø IF E$="" THEN 3ØØ
32Ø IF ASC(E$)<>13 THEN 3ØØ
33Ø CLS
34Ø S=Ø
```

```
350 FOR T=1 TO 10
360 CLS
370 A=RND(10)-1
380 B=RND(10)-1
390 C=RND(10)-1
400 D=RND(10)-1
410 I=RND(3)
420 ON I GOTO 430,450,470
430 PRINT A$(A);" ";B$(B);" ";C$(C);" ";D$(D);"."
440 GOTO 480
450 PRINT A$(A);" ";B$(B);" ";D$(D);" ";C$(C);"."
460 GOTO 480
470 PRINT D$(D);" ";A$(A);" ";B$(B);" ";C$(C);"."
480 PRINT
490 INPUT "ADVERB ";V$
500 PRINT
510 IF V$=D$(D) THEN 540
520 PRINT "THE ADVERB IS: ";D$(D)
530 GOTO 560
540 PRINT "CORRECT!"
550 S=S+1
560 PRINT
570 PRINT "PRESS <ENTER>."
580 E$=INKEY$
590 IF E$="" THEN 580
600 IF ASC(E$)<>13 THEN 580
610 NEXT T
620 CLS
630 PRINT "YOUR SCORE IS"
640 PRINT TAB(5);"RIGHT",S
650 PRINT TAB(5);"WRONG",10-S
660 PRINT
670 PRINT
680 PRINT
690 PRINT "TRY AGAIN? (Y/N)"
700 E$=INKEY$
710 IF E$="Y" THEN 330
720 IF E$<>"N" THEN 700
730 CLS
740 END
```

Chapter 3
Quick Calculations

Chapter 3
Quick Calculations

You can use your computer to perform important calculations. In fact, I know a bank that bought a microcomputer system (for about $6000). When asked what they did with their computer, they said it was used for amortization. You can use your own $100 computer to do the same thing.

You can use your computer as a glorified calculator — figuring conversions and formulas. The difference between a calculator and a computer is that with a computer you can put the formula in a program once and then not worry about it again. You can RUN the program and input known values. The program can give you the final answer almost instantly. With a calculator, you need to remember the equation or formula for each problem.

Let's look at some examples. The simple program below converts an angle in degrees to an angle in radians. The general conversion formula is 180 degrees= π radians.

```
110 PRINT
120 INPUT "NUMBER OF DEGREES = ";D
130 PRINT
140 PRINT "RADIANS: ";D*3.14159/180
150 GOTO 110
```

Type in the program, then RUN. The computer will ask for the number of degrees. Enter a number, such as 45. The equivalent number of radians in decimal form will be printed. The program continues until you press RUN/STOP or BREAK.

Many microcomputers allow you to *define* a function at the beginning of the program, calling it a variable name. Later, any calculations using that variable will evaluate the function. There are several restrictions, depending on the type of computer you have. One of the requirements is that the computer must process the line defining the function before the function is actually used. In some computers the definition statement must have a lower line number than any line in which the function is used. To play it safe, put definitions at the beginning of the program.

There are several forms of the definition statement. The Commodore, Apple, and TRS-80 use DEF FN name (variable), where a variable is specified in parentheses and the name is often a letter. Some computers allow functions to be redefined anywhere in the program. An example is:

```
100 DEF FN A(R)=3.14159*R*R
200 PRINT FN A(5)
```

The formula with the value of 5 for R is evaluated and printed.

In TI BASIC, leave out FN. More than one variable is permitted, and you don't have to have a dependent variable:

```
100 DEF A(R)=3.14159*R*R
200 PRINT A(5)
```

or

```
100 DEF R16=INT(16*RND+1)
200 CALL COLOR(5,R16,R16)
```

Each time R16 is used, a random number from 1 to 16 is chosen. Within one command you can use the function several times:

```
210 CALL HCHAR(R16,R16,R16+63,R16)
```

Suppose you want to evaluate the expression $10x^2+7x+1$ for values of x equal to 0, 1, 3, 5, -1, and -5. Here is one way it could be done:

```
100 DEF FNA(X)=10*X*X+7*X+1
110 PRINT
120 FOR I=1 TO 6
130 READ X
140 PRINT "X = ";X,"A(X) = ";FNA(X)
150 NEXT I
160 PRINT
170 DATA 0,1,3,5,-1,-5
180 END
```

Line 100 defines the function A(X). The space between FN and A(X) is optional, so to save memory you can leave it out, as above. Line 130 reads the values for X from the DATA statement in line 170. Line 140 prints the X value and the evaluated function.

The following programs illustrate how you can use your computer to take the drudgery out of calculations. These programs can be translated for your computer quite easily. If you want, you can combine lines as shown in Chapter 1 to make the listings shorter. Remember to use your own computer's commands for such things as clearing the screen, creating random numbers, or detecting a key pressed.

Conversions

Computers: Translatable

This program includes conversion factors for linear measurements. First, choose which measurement you need to convert, then enter a number. The equivalent measurement is printed. You can use this general form to write your own program for the conversions you often need.

Program 3-1. Conversions

Translatable

```
10 CLS
20 PRINT "** CONVERSIONS **"
30 PRINT
40 PRINT "CHOOSE:"
50 PRINT " 1 CENTIMETER TO FOOT"
60 PRINT " 2 FOOT TO CENTIMETER"
70 PRINT " 3 CENTIMETER TO INCH"
80 PRINT " 4 INCH TO CENTIMETER"
90 PRINT " 5 INCH TO FOOT"
100 PRINT " 6 FOOT TO INCH"
110 PRINT " 7 METER TO FOOT"
120 PRINT " 8 FOOT TO METER"
130 PRINT " 9 FEET TO MILE"
140 PRINT "10 MILE TO FEET"
150 PRINT "11 MILE TO KILOMETER"
160 PRINT "12 KILOMETER TO MILE"
170 PRINT "13 METER TO YARD"
180 PRINT "14 YARD TO METER"
190 INPUT A
200 IF A>14 THEN 10
210 IF A<1 THEN 10
220 PRINT
230 PRINT
240 ON A GOTO 250,300,350,400,450,500,550,600,650,
    700,750,800,850,900
250 PRINT "1 CENTIMETER = .032808 FOOT"
260 PRINT
270 INPUT "CENTIMETERS = ";N
280 PRINT .032808*N;" FOOT"
290 GOTO 940
300 PRINT "1 FOOT = 30.48006 CENTIMETER"
310 PRINT
320 INPUT "NUMBER OF FEET = ";N
330 PRINT 30.48006*N;" CENTIMETERS"
340 GOTO 940
350 PRINT "1 CENTIMETER = .39370 INCH"
360 PRINT
370 INPUT "NUMBER OF CENTIMETERS = ";N
380 PRINT N*.39370;" INCH"
390 GOTO 940
400 PRINT "1 INCH = 2.540005 CENTIMETER"
410 PRINT
420 INPUT "NUMBER OF INCHES = ";N
430 PRINT N*2.540005;" CENTIMETERS"
```

```
44Ø GOTO 94Ø
45Ø PRINT "1 INCH = 1/12 FOOT"
46Ø PRINT
47Ø INPUT "NUMBER OF INCHES = ";N
48Ø PRINT N/12;" FOOT"
49Ø GOTO 94Ø
5ØØ PRINT "1 FOOT = 12 INCHES"
51Ø PRINT
52Ø INPUT "NUMBER OF FEET = ";N
53Ø PRINT 12*N;" INCHES"
54Ø GOTO 94Ø
55Ø PRINT "1 METER = 3.28Ø833 FEET"
56Ø PRINT
57Ø INPUT "METERS = ";N
58Ø PRINT 3.28Ø833*N;" FEET"
59Ø GOTO 94Ø
6ØØ PRINT "1 FOOT = .3Ø48ØØ6 METER"
61Ø PRINT
62Ø INPUT "NUMBER OF FEET = ";N
63Ø PRINT .3Ø48ØØ6*N;" METER"
64Ø GOTO 94Ø
65Ø PRINT "1 FOOT = 1/528Ø MILE"
66Ø PRINT
67Ø INPUT "NUMBER OF FEET = ";N
68Ø PRINT N/528Ø;" MILE"
69Ø GOTO 94Ø
7ØØ PRINT "1 MILE = 528Ø FEET"
71Ø PRINT
72Ø INPUT "MILES = ";N
73Ø PRINT 528Ø*N;" FEET"
74Ø GOTO 94Ø
75Ø PRINT "1 MILE = 1.6Ø935 KILOMETERS"
76Ø PRINT
77Ø INPUT "MILES = ";N
78Ø PRINT 1.6Ø935*N;" KILOMETERS"
79Ø GOTO 94Ø
8ØØ PRINT "1 KILOMETER = .62137 MILE"
81Ø PRINT
82Ø INPUT "KILOMETERS = ";N
83Ø PRINT N*.62137;" MILES"
84Ø GOTO 94Ø
85Ø PRINT "1 METER = 1.Ø93611 YARDS"
86Ø PRINT
87Ø INPUT "METERS = ";N
88Ø PRINT N*1.Ø93611;" YARDS"
89Ø GOTO 94Ø
9ØØ PRINT "1 YARD = .9144Ø183 METER"
91Ø PRINT
92Ø INPUT "YARDS = ";N
```

```
930 PRINT N*.91440183;" METERS"
940 PRINT
950 PRINT "CHOOSE:"
960 PRINT " 1 ANOTHER PROBLEM"
970 PRINT " 2 END PROGRAM"
980 A$=INKEY$
990 IF A$="1" THEN 10
1000 IF A$<>"2" THEN 980
1010 PRINT
1020 END
```

Distance

Computers: Translatable

Any type of formula can be put into a simple program. This program uses the formula Distance = Rate * Time. It allows you to choose which item becomes the unknown. You input the known values, and the unknown value will be printed.

Program 3-2. Distance

Translatable

```
10 CLS
20 PRINT "**DISTANCE=RATE*TIME**"
30 PRINT
40 PRINT
50 PRINT "WHICH IS THE UNKNOWN?"
60 PRINT " 1 DISTANCE"
70 PRINT " 2 RATE"
80 PRINT " 3 TIME"
90 A$=INKEY$
100 IF A$="2" THEN 210
110 IF A$="3" THEN 290
120 IF A$<>"1" THEN 90
130 PRINT
140 PRINT
150 PRINT "DISTANCE = RATE * TIME"
160 PRINT
170 INPUT "RATE = ";R
180 INPUT "TIME = ";T
190 PRINT "DISTANCE =";R*T
200 GOTO 360
210 PRINT
220 PRINT
230 PRINT "RATE = DISTANCE/TIME"
240 PRINT
250 INPUT "DISTANCE = ";D
260 INPUT "TIME = ";T
270 PRINT "RATE =";D/T
280 GOTO 360
```

```
290 PRINT
300 PRINT
310 PRINT "TIME = DISTANCE/RATE"
320 PRINT
330 INPUT "DISTANCE = ";D
340 INPUT "RATE = ";R
350 PRINT "TIME =";D/R
360 PRINT
370 PRINT
380 PRINT "CHOOSE:"
390 PRINT " 1 ANOTHER PROBLEM"
400 PRINT " 2 END PROGRAM"
410 A$=INKEY$
420 IF A$="1" THEN 30
430 IF A$<>"2" THEN 410
440 PRINT
450 END
```

Squares

Computers: Translatable

"Squares" calculates the area, side, or perimeter of a square if you know one of these quantities. Line 10 clears the screen. Lines 90, 260, and 520 detect a key pressed. You may need to change these lines to GET A$ or CALL KEY(0,K,S) for your particular computer.

Program 3-3. Squares

Translatable

```
10 CLS
20 PRINT "** SQUARES **"
30 PRINT
40 PRINT
50 PRINT "WHICH IS UNKNOWN?"
60 PRINT " 1 AREA"
70 PRINT " 2 SIDE"
80 PRINT " 3 PERIMETER"
90 A$=INKEY$
100 IF A$="2" THEN 200
110 IF A$="3" THEN 410
120 IF A$<>"1" THEN 90
130 PRINT
140 PRINT
150 PRINT "AREA = SIDE * SIDE"
160 PRINT
170 INPUT "SIDE = ";S
180 PRINT "AREA =";S*S
190 GOTO 470
```

```
200 PRINT
210 PRINT
220 PRINT "NEED TO FIND SIDE"
230 PRINT "WHAT IS GIVEN?"
240 PRINT " 1 AREA"
250 PRINT " 2 PERIMETER"
260 A$=INKEY$
270 IF A$="1" THEN 350
280 IF A$<>"2" THEN 260
290 PRINT
300 PRINT "SIDE = PERIMETER/4"
310 PRINT
320 INPUT "PERIMETER = ";P
330 PRINT "SIDE =";P/4
340 GOTO 470
350 PRINT
360 PRINT "SIDE = SQR ROOT OF AREA"
370 PRINT
380 INPUT "AREA = ";A
390 PRINT "SIDE =";SQR(A)
400 GOTO 470
410 PRINT
420 PRINT
430 PRINT "PERIMETER = 4*SIDE"
440 PRINT
450 INPUT "SIDE = ";S
460 PRINT "PERIMETER =";4*S
470 PRINT
480 PRINT
490 PRINT "CHOOSE:"
500 PRINT " 1 ANOTHER PROBLEM"
510 PRINT " 2 END PROGRAM"
520 A$=INKEY$
530 IF A$="1" THEN 10
540 IF A$<>"2" THEN 520
550 PRINT
560 END
```

Rectangles

Computers: Translatable

"Rectangles" also illustrates plane geometry principles. The computer will calculate area, perimeter, or the length of the unknown side if all other necessary quantities are given. Lines 90, 340, and 500 detect a key pressed. Change these to fit your computer by using GET A$ or CALL KEY(0,K,S).

Program 3-4. Rectangles

Translatable

```
1Ø CLS
2Ø PRINT "** RECTANGLES **"
3Ø PRINT
4Ø PRINT
5Ø PRINT "WHAT IS UNKNOWN?"
6Ø PRINT " 1 AREA"
7Ø PRINT " 2 PERIMETER"
8Ø PRINT " 3 ONE SIDE"
9Ø A$=INKEY$
1ØØ IF A$="2" THEN 21Ø
11Ø IF A$="3" THEN 29Ø
12Ø IF A$<>"1" THEN 9Ø
13Ø PRINT
14Ø PRINT
15Ø PRINT "AREA = LENGTH * WIDTH"
16Ø PRINT
17Ø INPUT "LENGTH = ";L
18Ø INPUT "WIDTH = ";W
19Ø PRINT "AREA =";L*W
2ØØ GOTO 45Ø
21Ø PRINT
22Ø PRINT
23Ø PRINT "PERIMETER = 2*LENGTH + 2*WIDTH"
24Ø PRINT
25Ø INPUT "LENGTH = ";L
26Ø INPUT "WIDTH = ";W
27Ø PRINT "PERIMETER =";2*L+2*W
28Ø GOTO 45Ø
29Ø PRINT
3ØØ PRINT "WHAT IS GIVEN?"
31Ø PRINT " 1 AREA"
32Ø PRINT " 2 PERIMETER"
33Ø PRINT
34Ø A$=INKEY$
35Ø IF A$="2" THEN 42Ø
36Ø IF A$<>"1" THEN 34Ø
37Ø INPUT "AREA = ";A
38Ø INPUT "GIVEN SIDE = ";S
39Ø IF A$<>"1" THEN 37Ø
4ØØ PRINT "SIDE =";A/S
41Ø GOTO 45Ø
42Ø INPUT "PERIMETER = ";P
43Ø INPUT "GIVEN SIDE = ";S
44Ø PRINT "SIDE ="; (P-2*S)/2
45Ø PRINT
46Ø PRINT
47Ø PRINT "CHOOSE:"
```

```
480 PRINT " 1 ANOTHER PROBLEM"
490 PRINT " 2 END PROGRAM"
500 A$=INKEY$
510 IF A$="1" THEN 10
520 IF A$<>"2" THEN 500
530 PRINT
540 END
```

Circles

Computers: Translatable

"Circles" uses the formulas for the radius, diameter, area, or circumference of a circle. First, decide which is the unknown, or the answer you'll want. Second, determine the given quantity and input that value. The answer will then be printed.

The program uses INPUT, rather than detecting a key press, to receive a menu choice.

(Note: This program will not work on the TRS-80 4K Color Computer because the program uses the square root function, which is not available in 4K Color BASIC. You may, however, adapt the program by leaving out all sections involving the square root and by combining lines and leaving out all spaces. For a program which calculates a number's square root on the Color Computer, refer to Program 3-6.)

Program 3-5. Circles

Translatable

```
10 PRINT"** CIRCLES **"
20 PRINT
30 PRINT"WHICH IS UNKNOWN?"
40 PRINT" 1 RADIUS"
50 PRINT" 2 DIAMETER"
60 PRINT" 3 AREA"
70 PRINT" 4 CIRCUMFERENCE"
80 INPUT A
90 IF A>=1 THEN 120
100 PRINT"ENTER 1, 2, 3, OR 4"
110 GOTO 80
120 IF A>4 THEN 100
130 PRINT
140 ON A GOTO 150,410,670,930
150 PRINT"WHAT IS GIVEN?"
160 PRINT" 1 DIAMETER"
170 PRINT" 2 AREA"
180 PRINT" 3 CIRCUMFERENCE"
190 INPUT A
200 IF A>=1 THEN 230
```

```
210 PRINT"ENTER 1, 2, OR 3"
220 GOTO 190
230 IF A>3 THEN 210
240 PRINT
250 ON A GOTO 260,310,360
260 PRINT"RADIUS = DIAMETER/2"
270 PRINT
280 INPUT "DIAMETER = ";D
290 PRINT"RADIUS = ";D/2
300 GOTO 1180
310 PRINT"RADIUS = SQR(AREA/PI)"
320 PRINT
330 INPUT "AREA = ";A
340 PRINT"RADIUS = ";SQR(A/3.14159)
350 GOTO 1180
360 PRINT"RADIUS = CIRCUM/2 PI"
370 PRINT
380 INPUT "CIRCUMFERENCE = ";C
390 PRINT"RADIUS = ";C/(2*3.14159)
400 GOTO 1180
410 PRINT"WHAT IS GIVEN?"
420 PRINT" 1 RADIUS"
430 PRINT" 2 AREA"
440 PRINT" 3 CIRCUMFERENCE"
450 INPUT A
460 IF A>=1 THEN 490
470 PRINT"ENTER 1, 2, OR 3"
480 GOTO 450
490 IF A>3 THEN 470
500 PRINT
510 ON A GOTO 520,570,620
520 PRINT"DIAMETER = 2*RADIUS"
530 PRINT
540 INPUT "RADIUS = ";R
550 PRINT"DIAMETER = ";2*R
560 GOTO 1180
570 PRINT"DIAMETER = 2*SQR(AREA/PI)
580 PRINT
590 INPUT "AREA = ";A
600 PRINT"DIAMETER = ";2*SQR(A/3.14159)
610 GOTO 1180
620 PRINT"DIAMETER = CIRCUM/PI"
630 PRINT
640 INPUT "CIRCUMFERENCE = ";C
650 PRINT"DIAMFTER = ";C/3.14159
660 GOTO 1180
670 PRINT"WHICH IS GIVEN?"
680 PRINT" 1 RADIUS"
690 PRINT" 2 DIAMETER"
```

```
700 PRINT" 3 CIRCUMFERENCE"
710 INPUT A
720 IF A>=1 THEN 750
730 PRINT"ENTER 1, 2, OR 3"
740 GOTO 710
750 IF A>3 THEN 730
760 PRINT
770 ON A GOTO 780,830,880
780 PRINT"AREA = PI*RADIUS SQUARED"
790 PRINT
800 INPUT "RADIUS = ";R
810 PRINT"AREA = ";3.14159*R*R
820 GOTO 1180
830 PRINT"AREA = PI * DIAM SQUARED/4"
840 PRINT
850 INPUT "DIAMETER = ";D
860 PRINT"AREA = ";3.14159*D*D/4
870 GOTO 1180
880 PRINT"AREA = CIRCUM SQUARED/4 PI"
890 PRINT
900 INPUT "CIRCUMFERENCE = ";C
910 PRINT"AREA = ";C*C/4/3.14159
920 GOTO 1180
930 PRINT"WHAT IS GIVEN?"
940 PRINT" 1 RADIUS"
950 PRINT" 2 DIAMETER"
960 PRINT" 3 AREA"
970 INPUT A
980 IF A>=1 THEN 1010
990 PRINT"ENTER 1, 2, OR 3"
1000 GOTO 970
1010 IF A>3 THEN 990
1020 PRINT
1030 ON A GOTO 1040,1090,1140
1040 PRINT"C=2 PI*RADIUS"
1050 PRINT
1060 INPUT "RADIUS = ";R
1070 PRINT"CIRCUMFERENCE = ";2*3.14159*R"
1080 GOTO 1180
1090 PRINT"C = PI*DIAMETER"
1100 PRINT
1110 INPUT "DIAMETER = ";D
1120 PRINT"CIRCUMFERENCE = ";3.14159*D
1130 GOTO 1180
1140 PRINT"C = 2*SQR(AREA*PI)"
1150 PRINT
1160 INPUT "AREA = ";A
1170 PRINT"CIRCUMFERENCE = ";2*SQR(A*3.14159)
1180 PRINT
```

```
1190 PRINT"CHOOSE:"
1200 PRINT" 1 ANOTHER PROBLEM"
1210 PRINT" 2 END PROGRAM"
1220 INPUT A
1230 IF A=1 THEN 20
1240 IF A<>2 THEN 1220
1250 END
```

Square Root

Computers: Translatable

If you have the TRS-80 4K Color Computer or another computer which does not have the square root function SQR(*n*), you can use this program to calculate the square root of a number *n*. You could convert this program to a subroutine to be called whenever you need a square root.

Program 3-6. Square Root

Translatable

```
10 CLS
20 PRINT "** SQUARE ROOT **"
30 PRINT
40 PRINT
50 INPUT "NUMBER = ";N
60 PRINT
70 IF N<>0 THEN 100
80 PRINT "SQUARE ROOT = 0"
90 GOTO 250
100 IF N>0 THEN 160
110 PRINT
120 PRINT "PLEASE ENTER NUMBER"
130 PRINT "GREATER THAN ZERO."
140 PRINT
150 GOTO 250
160 PRINT "SQUARE ROOT =";
170 S=N/2
180 T=0
190 U=(N/S-S)/2
200 IF (U=0)+(U=T) THEN 240
210 S=S+U
220 T=U
230 GOTO 190
240 PRINT S
250 PRINT
260 PRINT "CHOOSE:"
270 PRINT " 1 ANOTHER PROBLEM"
280 PRINT " 2 END PROGRAM"
290 A$=INKEY$
```

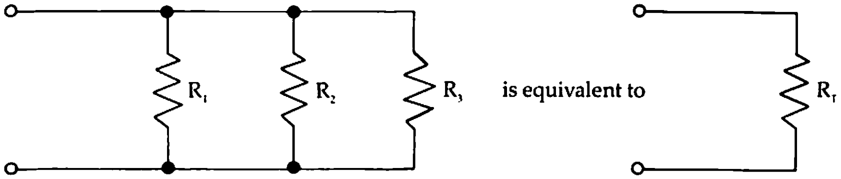
```
300 IF A$="1" THEN 10
310 IF A$<>"2" THEN 290
320 PRINT
330 END
```

Parallel Resistance

Computers: Translatable

"Parallel Resistance" is an electrical engineering program that illustrates how quickly your computer can calculate a complex equation. In electrical engineering circuit analysis, a number of resistors can be combined in parallel to create a total resistance value. The formula states that the reciprocal of the total is equal to the sum of the reciprocals of each one — in other words, an equation involving fractions.

For example:



where:

$1/R_T = 1/R_1 + 1/R_2 + 1/R_3$

For this program, enter the number of resistors, then enter the resistance for each one. The total resistance is calculated and printed.

Program 3-7. Parallel Resistance

Translatable

```
10 CLS
20 PRINT "** PARALLEL RESISTANCE **"
30 PRINT
40 PRINT " 1{4 SPACES}1{4 SPACES}1{4 SPACES}1"
50 PRINT "-- = -- + -- + -- + ..."
60 PRINT "RT{3 SPACES}R1{3 SPACES}R2{3 SPACES}R3
70 PRINT
80 PRINT "YOUR PROBLEM:"
90 INPUT "HOW MANY RESISTORS? ";N
100 N=INT(N)
110 IF N>0 THEN 140
```

```
12Ø PRINT "SORRY, ONE OR MORE"
13Ø GOTO 9Ø
14Ø IF N<51 THEN 18Ø
15Ø PRINT "IF YOU REALLY HAVE MORE"
16Ø PRINT "THAN 5Ø, SOLVE IN STEPS."
17Ø GOTO 9Ø
18Ø PRINT
19Ø D=Ø
2ØØ FOR I=1 TO N
21Ø PRINT "R";I;"=";
22Ø INPUT R
23Ø IF R<>Ø THEN 26Ø
24Ø PRINT "SORRY, ZERO NOT ALLOWED"
25Ø GOTO 21Ø
26Ø D=D+1/R
27Ø NEXT I
28Ø PRINT
29Ø PRINT "TOTAL R = ";1/D
3ØØ PRINT
31Ø END
```

Arithmetic Mean

Computers: Translatable

All types of statistical programs can be done with your computer — all you have to do is pick the appropriate formula or procedure and write a short program. This program illustrates the calculation of the arithmetic mean or average of a group of numbers. First, enter the number of items, then enter each value. The total and the average are printed.

Program 3-8. Arithmetic Mean

Translatable

```
1Ø CLS
2Ø PRINT "** ARITHMETIC MEAN **"
3Ø PRINT
4Ø INPUT "HOW MANY ITEMS";N
5Ø N=INT(N)
6Ø IF N>Ø THEN 1ØØ
7Ø PRINT "PLEASE ENTER A WHOLE NUMBER"
8Ø PRINT "GREATER THAN ZERO."
9Ø GOTO 4Ø
1ØØ T=Ø
11Ø FOR I=1 TO N
12Ø PRINT " VALUE";I;" = ";
13Ø INPUT A
14Ø T=T+A
15Ø NEXT I
```

```
160 PRINT
170 PRINT "TOTAL = ";T
180 PRINT "AVERAGE = ";T/N
190 PRINT
200 END
```

Simple Interest

Computers: Translatable

Suppose you have a savings account that pays simple interest. If you deposit a certain amount now, how much will it be worth later? In "Simple Interest," you enter your principal deposit, the interest rate in percent, and the number of years. The future value will be printed.

Another use for this program would be to calculate a simple interest loan. If you borrowed money at simple interest, what would be the total amount due *n* years from now? Entering the principal loan, the interest rate, and the number of years will produce the total amount due at the end of the loan's term.

Program 3-9. Simple Interest

Translatable

```
10 CLS
20 PRINT "** SIMPLE INTEREST **"
30 PRINT
40 PRINT "ENTER PRINCIPAL DEPOSIT."
50 INPUT P
60 IF P>0 THEN 90
70 PRINT "MUST BE MORE THAN ZERO."
80 GOTO 30
90 PRINT
100 PRINT "ENTER INTEREST RATE IN PERCENT"
110 PRINT "SUCH AS 6 FOR 6 PERCENT."
120 INPUT I
130 PRINT
140 INPUT "HOW MANY YEARS";N
150 IF N>0 THEN 180
160 PRINT "MUST BE MORE THAN ZERO."
170 GOTO 130
180 PRINT
190 A=P*(1+N*(I/100))
200 PRINT "VALUE OF";P
210 PRINT "AT INTEREST";I;"PERCENT"
220 PRINT "FOR";N;"YEARS = "
230 PRINT A
240 PRINT
250 END
```

Compound Interest

Computers: Translatable

Most banks and savings associations now use compound interest. This program calculates the future value of a present deposit earning compound interest. You will be asked to enter the principal deposit, the interest rate in percent, how many times interest is compounded per year (quarterly would be 4, daily would be 365), and how many years in the analysis. The equation in line 240 involves a power or exponent symbol. Enter your computer's symbol for this function. For example, a VIC-20 would show an arrow (↑) here. Exponentiation is not available on the TRS-80 Color Computer unless Extended Color BASIC is installed. The future value will be printed.

As in Program 3-9, Simple Interest, you could use this program to calculate a compound interest loan's total amount due. Just enter the principal of the loan, the interest rate, times compounded, and the number of years for the loan. The total value will be shown. You could use this to find out how much interest you'll pay on a car loan, for instance.

Program 3-10. Compound Interest

Translatable

```
1Ø CLS
2Ø PRINT "** COMPOUND INTEREST **"
3Ø PRINT
4Ø PRINT "ENTER PRINCIPAL DEPOSIT"
5Ø INPUT P
6Ø IF P>Ø THEN 9Ø
7Ø PRINT "MUST BE MORE THAN ZERO."
8Ø GOTO 3Ø
9Ø PRINT
1ØØ PRINT "ENTER INTEREST RATE IN PERCENT"
11Ø INPUT I
12Ø PRINT
13Ø PRINT "COMPOUNDED HOW MANY TIMES"
14Ø INPUT "PER YEAR";Q
15Ø IF Q>=Ø THEN 18Ø
16Ø PRINT "MUST BE MORE THAN ZERO."
17Ø GOTO 12Ø
18Ø PRINT
19Ø INPUT "HOW MANY YEARS";N
2ØØ IF N>Ø THEN 23Ø
21Ø PRINT "MUST BE MORE THAN ZERO."
22Ø GOTO 18Ø
23Ø PRINT
24Ø A=P*(1+(I/1ØØ)/Q)^(N*Q)
25Ø PRINT "PRINCIPAL";P
```

```
260 PRINT "AT INTEREST";I;"PERCENT"
270 PRINT "COMPOUNDED";Q;"TIMES PER YEAR"
280 PRINT "FOR";N;"YEARS ="
290 PRINT A
300 PRINT
310 END
```

Single Payment Present Worth

Computers: Translatable

Have you ever wanted to know how much money you'd have to deposit *today* to have $10,000 for your retirement? You can find out by using this program. Enter the future sum (10000, for example), the number of years you have until age 65, and a realistic interest rate in percent. The amount you would have to have right now will be calculated.

Line 220 contains a "raise to power" (*exponent*) symbol. Make sure you enter the proper symbol for exponentiation for your computer here. Note that the TRS-80 Color Computer does not have the exponentiation function unless you have Extended Color BASIC installed.

Program 3-11. Single Payment

Translatable

```
10 CLS
20 PRINT "** SINGLE PAYMENT **"
30 PRINT "** PRESENT WORTH **"
40 PRINT
50 PRINT "GIVEN A FUTURE SUM,"
60 PRINT "FIND ITS VALUE TODAY,"
70 PRINT "N PERIODS EARLIER."
80 PRINT
90 PRINT "WHAT IS THE FUTURE SUM?"
100 INPUT S
110 IF S>=0 THEN 140
120 PRINT "MUST BE MORE THAN ZERO."
130 GOTO 80
140 PRINT
150 INPUT "HOW MANY PERIODS";N
160 IF N>0 THEN 190
170 PRINT "MUST BE GREATER THAN ZERO"
180 GOTO 140
190 PRINT
200 PRINT "ENTER INTEREST RATE IN PERCENT"
210 INPUT I
220 P=S*(1/((1+I/100)^N))
230 PRINT
240 PRINT "PRESENT VALUE =";P
```

Uniform Series Compound Amount

Computers: Translatable

Suppose you have a savings account earning compound interest. You deposit the same amount each month. After *n* months, how much will your account be worth?

"Uniform Series" calculates the result of a uniform series of payments. Enter the payment amount, the number of payments, and the interest in percent. The final total, plus interest, will be printed on the screen.

Line 240 contains a power or exponent symbol. Use your computer's symbol here. For the TRS-80 Color Computer, Extended Color BASIC is required.

Program 3-12. Uniform Series

Translatable

```
1Ø CLS
2Ø PRINT "** UNIFORM SERIES **"
3Ø PRINT "** COMPOUND AMOUNT **"
4Ø PRINT
5Ø PRINT "GIVEN A UNIFORM SERIES OF"
6Ø PRINT "END-OF-PERIOD PAYMENTS,"
7Ø PRINT "WHAT WILL N PAYMENTS"
8Ø PRINT "ACCUMULATE TO AT"
9Ø PRINT "COMPOUND INTEREST I?"
1ØØ PRINT
11Ø INPUT "ENTER PAYMENT AMOUNT";R
12Ø IF R>Ø THEN 15Ø
13Ø PRINT "MUST BE MORE THAN ZERO"
14Ø GOTO 1ØØ
15Ø PRINT
16Ø INPUT "HOW MANY PAYMENTS";N
17Ø IF N>Ø THEN 2ØØ
18Ø PRINT "MUST BE MORE THAN ZERO"
19Ø GOTO 15Ø
2ØØ PRINT
21Ø PRINT "ENTER INTEREST IN PERCENT"
22Ø INPUT "PER PERIOD";I
23Ø I=I/1ØØ
24Ø S=R*(((1+I)^N-1)/I)
25Ø PRINT
26Ø PRINT "FINAL VALUE =";S
27Ø PRINT
28Ø END
```

Sinking Fund Deposit

Computers: Translatable

You need a certain sum in a future year and want to know how much to deposit each month to reach that amount. Your account earns compound interest. Enter the desired future sum, the number of time periods, and the interest expressed as a percent per period. The program will then calculate the monthly deposit needed.

Make sure you enter your computer's exponent symbol in line 250. Exponentiation is not available on the TRS-80 Color Computer unless Extended Color BASIC is installed.

Program 3-13. Sinking Fund

Translatable

```
10 CLS
20 PRINT "** SINKING FUND DEPOSIT **"
30 PRINT
40 PRINT "WHAT UNIFORM SERIES OF"
50 PRINT "END-OF-PERIOD DEPOSITS"
60 PRINT "MUST BE MADE FOR N PERIODS"
70 PRINT "AT COMPOUND INTEREST I TO"
80 PRINT "GET A FUTURE SUM?"
90 PRINT
100 PRINT "ENTER DESIRED FUTURE SUM."
110 INPUT S
120 IF S>=0 THEN 150
130 PRINT "MUST BE MORE THAN ZERO"
140 GOTO 90
150 PRINT
160 INPUT "HOW MANY PERIODS";N
170 IF N>=1 THEN 200
180 PRINT "MUST BE AT LEAST ONE"
190 GOTO 150
200 PRINT
210 PRINT "ENTER INTEREST AS PERCENT"
220 INPUT "PER PERIOD";I
230 I=I/100
240 PRINT
250 R=S*(I/((1+I)^N-1))
260 PRINT "SINKING FUND DEPOSITS ="
270 PRINT R
280 PRINT
290 END
```

Loan Payments

Computers: Translatable

You would like to borrow a sum of money from a bank that charges compound interest. What would the monthly payments be? Using this program, you can find out.

Enter the amount you want to borrow, and the number of years for the loan. For example, home mortgages may be 25 or 30 years, but second mortgages are usually only 10 or 15 years. Next, enter the annual interest rate in percent — the interest rate the bank quoted you. The program will compute your monthly payment. By running this program you can quickly see the difference in your monthly payment with different interest rates — or different times — for the same amount of money.

Line 230 contains an exponent symbol. Enter your computer's symbol in this line. Exponentiation is not available on TRS-80 Color Computers unless Extended Color BASIC is installed.

Program 3-14. Loan Payments

Translatable

```
1Ø CLS
2Ø PRINT "** LOAN PAYMENTS **"
3Ø PRINT
4Ø PRINT "YOU WANT TO BORROW A"
5Ø PRINT "CERTAIN SUM OF MONEY."
6Ø PRINT "IF INTEREST IS COMPOUNDED,"
7Ø PRINT "WHAT IS THE MONTHLY PAYMENT?"
8Ø PRINT
9Ø INPUT "AMOUNT BORROWED";P
1ØØ IF P>Ø THEN 13Ø
11Ø PRINT "AMOUNT MORE THAN ZERO PLEASE"
12Ø GOTO 8Ø
13Ø PRINT
14Ø INPUT "HOW MANY YEARS";Y
15Ø IF Y>Ø THEN 18Ø
16Ø PRINT "MUST BE MORE THAN ZERO"
17Ø GOTO 13Ø
18Ø N=12*Y
19Ø PRINT
2ØØ PRINT "WHAT IS THE INTEREST RATE"
21Ø INPUT "IN PERCENT";I
22Ø I=I/12ØØ
23Ø F=(1+I)^N
24Ø M=P*(I*F/(F-1))
25Ø M=(INT(1ØØ*(M+.ØØ5)))/1ØØ
26Ø PRINT
27Ø PRINT "MONTHLY PAYMENT =";M
```

```
280 PRINT
290 END
```

Amortization

Computers: Translatable

"Amortization" is a continuation of Program 3-14, "Loan Payments." After your monthly payment is calculated, you can use this program to see an amortization schedule — a list by month of the amount of principal paid, the amount of interest paid, and the current balance.

Enter the amount borrowed, the number of years for the loan, and the interest rate in percent. The computer will display the monthly payment. The program then gives you the option to see the amortization schedule or end the program. Enter *1* to see the amortization. Pressing 2 will end the program. Since most computers display words faster than we can read, only one line at a time is shown, representing one month's payment. Press any key to see the next line. (For the VIC-20 you'll need to adjust the printing to fit 22 column lines.)

Line 230 contains a power, or exponent, symbol. Use your computer's symbol here. This program cannot be used on TRS-80 Color Computers unless Extended Color BASIC is installed. This is because the exponentiation function is not available in standard Color BASIC.

Program 3-15. Amortization

Translatable

```
10 CLS
20 PRINT "** AMORTIZATION **"
30 PRINT
40 PRINT "YOU WANT TO BORROW A"
50 PRINT "CERTAIN SUM OF MONEY."
60 PRINT "IF INTEREST IS COMPOUNDED,"
70 PRINT "WHAT IS THE MONTHLY PAYMENT?"
80 PRINT
90 INPUT "AMOUNT BORROWED";P
100 IF P>0 THEN 130
110 PRINT "AMOUNT MORE THAN ZERO PLEASE"
120 GOTO 80
130 PRINT
140 INPUT "HOW MANY YEARS";Y
150 IF Y>0 THEN 180
160 PRINT "MUST BE MORE THAN ZERO"
170 GOTO 130
```

```
18Ø N=12*Y
19Ø PRINT
2ØØ PRINT "WHAT IS THE INTEREST RATE"
21Ø INPUT "IN PERCENT";I
22Ø I=I/12ØØ
23Ø F=(1+I)^N
24Ø M=P*(I*F/(F-1))
25Ø M=(INT(1ØØ*(M+.ØØ5)))/1ØØ
26Ø PRINT
27Ø PRINT "MONTHLY PAYMENT =";M
28Ø PRINT
29Ø PRINT "CHOOSE:"
3ØØ PRINT " 1 PRINT AMORTIZATION"
31Ø PRINT " 2 END PROGRAM"
32Ø INPUT A
33Ø IF A=2 THEN 51Ø
34Ø IF A<>1 THEN 32Ø
35Ø PRINT
36Ø PRINT "WHILE THE SCHEDULE IS BEING"
37Ø PRINT "PRINTED, PRESS ANY KEY TO"
38Ø PRINT "SEE THE NEXT LINE."
39Ø PRINT
4ØØ PRINT "N  PRINCIPAL  INTEREST  BALANCE"
41Ø P=(INT(1ØØ*(P+.ØØ5)))/1ØØ
42Ø FOR C=1 TO N
43Ø J=(INT(1ØØ*(I*P+.ØØ5)))/1ØØ
44Ø A=(INT(1ØØ*((M-J)+.ØØ5)))/1ØØ
45Ø B=(INT(1ØØ*((P-A)+.ØØ5)))/1ØØ
46Ø P=B
47Ø PRINT C;TAB(4);A;TAB(14);J;TAB(23);B
48Ø A$=INKEY$
49Ø IF A$="" THEN 48Ø
5ØØ NEXT C
51Ø END
```

Chapter 4
Graphics

Chapter 4
Graphics

The rest of the programs in this book are computer-specific. Each program has been written for a particular machine, and its listing shows what should be entered for that computer. Many of the programs have been translated into other computers' versions of BASIC. Unlike the previous chapters, you won't have to go through the programs and locate commands and symbols to change. It's been done for you.

As before, each program notes which computer or computers the program will RUN on. In many cases, there will be only one explanation of the program's operation, but there will be several program listings. Each listing will be identified with the computer it will operate on. Most of the rest of the programs are listed for one of three computers: the VIC-20, the TRS-80 Color Computer, or the TI-99/4A. Variations of the Color Computer, such as the TRS-80 Model I and the MC-10, or variations of the TI-99/4, such as the TI-99/4A, are included in the list of possible computers where they apply.

This doesn't mean that you can't translate these programs to your computer, even if it is not one of these three machines. It just means that translation may be more difficult. Depending on the program and the computer features it uses, a translation may be simple, or next to impossible. Most of the commands in BASIC are the same for the different microcomputers. Sometimes the punctuation or functions used are slightly different, but in general most of the commands are similar enough to be translated from one computer to another without much problem. The main difference in variations in BASIC is found in the graphic and sound commands. These are very difficult to translate, for each computer uses its own graphics and sound commands. This chapter and Chapter 5 provide programs written for specific computers — graphics and sound commands have made the programs so different that separate listings are given for each computer.

Graphics Demo

Computers: VIC-20; TRS-80 Color Computer; MC-10; TI-99 /4A

"Graphics Demo" is a short program that randomly places squares of color on the screen while playing random tones. The program continues until you press any key.

Program 4-1. Graphics Demo

VIC-20

```
20 PRINT"{CLR}PRESS A KEY TO STOP."
30 S=36876:POKE36878,15
40 P=INT(5Ø6*RND(Ø))
50 T=INT(254*RND(Ø)+1)
60 POKEP+768Ø,16Ø:POKEP+384ØØ,INT(8*RND(Ø))
70 POKES,T
80 GET E$:IF E$=""THEN4Ø
90 POKES,Ø
100 END
```

Program 4-2. Graphics Demo

TRS-80 Color Computer; MC-10

```
2Ø CLS
3Ø PRINT "PRESS A KEY TO STOP."
4Ø I=RND(64)-1
5Ø J=RND(32)-1
6Ø C=RND(4)
7Ø SET(I,J,C)
8Ø SOUND RND(255),1
9Ø A$=INKEY$
1ØØ IF A$="" THEN 4Ø
11Ø END
```

Program 4-3. Graphics Demo

TI-99/4A

```
11Ø CALL CLEAR
12Ø PRINT "PRESS ANY KEY TO STOP."
13Ø FOR C=1 TO 16
14Ø CALL COLOR(C,C,C)
15Ø NEXT C
16Ø RANDOMIZE
17Ø CALL SOUND(-5Ø,INT(16ØØ*RND+11Ø),2)
18Ø CALL HCHAR(INT(24*RND+1),INT(32*RND+1),8*INT(1
    6*RND+1)+31)
19Ø CALL KEY(Ø,K,S)
2ØØ IF S<1 THEN 16Ø
21Ø END
```

Doodle

Computers: VIC-20; TRS-80 Color Computer; MC-10; TRS-80 Model I; TI-99/4A

"Doodle" lets you draw on the screen by pressing the arrow or cursor keys. The cursor starts in the middle of the screen. As you press a cursor or arrow key, the cursor moves in that direction and leaves a trail. Pressing the space bar changes colors. If you can't see the cursor, it may be the same color as the background screen, or it may be on a color previously drawn.

Here's how the VIC-20 version of Doodle works:

Line	Function
2	Clear screen.
3–5	PRINT instructions.
6–7	Wait for beginning key to be pressed.
8	Clear screen.
10	Initialize variables. P is screen position, S is character screen code number for graphics, and C is color number.
20	Draw square of color.
30	Blink square while waiting for input.
40–160	Depending on cursor key pressed, change screen position.
170–180	If space bar is pressed, change color.
190	RETURN to line 20 to draw new square.
200	END.

To move left or up, remember that you'll have to press SHIFT, then the appropriate cursor key.

Program 4-4. Doodle

VIC-20

```
2 PRINT "{CLR}"
3 PRINT"PRESS THE CRSR KEYS{3 SPACES}TO DRAW."
4 PRINT"{2 DOWN}PRESS THE SPACE BAR{3 SPACES}TO CH
  ANGE COLORS."
5 PRINT"{4 DOWN}PRESS F1 TO START."
6 GETE$:IFE$=""THEN6
7 IFASC(E$)<>133THEN6
8 PRINT"{CLR}"
10 P=7932:S=160:C=0
20 R=P+30720:POKEP,S:POKER,C
30 POKER,1:POKER,C:GETE$:IFE$=""THEN30
40 E=ASC(E$)
50 IFE<>17THEN80
60 IFP+22>8185THEN30
70 P=P+22:GOTO20
80 IFE<>29THEN110
90 IFP+1>8185THEN30
100 P=P+1:GOTO20
110 IFE<>145THEN140
120 IFP-22<7680THEN30
130 P=P-22:GOTO20
140 IFE<>157THEN170
```

```
15Ø IFP-1<768ØTHEN3Ø
16Ø P=P-1:GOTO2Ø
17Ø IFE<>32THEN3Ø
18Ø C=C+1:IFC>7THENC=Ø
19Ø GOTO2Ø
2ØØ END
```

Doodle works a bit differently on the Color Computer and MC-10:

Line	Function
20	Clear screen.
30–40	PRINT instructions.
50–70	Wait for key to be pressed.
80	Clear screen.
90	Initialize variables. P is screen position, C is color number.
100–110	Calculate character number R.
120–130	Blink square while waiting for input.
140–260	Depending on arrow key pressed, change screen position.
270–280	If space bar is pressed, change color.
290	RETURN to line 100 to draw new square.
300	END.

Program 4-5. Doodle

TRS-80 Color Computer; MC-10

```
2Ø CLS
3Ø PRINT "PRESS THE ARROW KEYS TO DRAW."
4Ø PRINT:PRINT "PRESS THE SPACE BAR":PRINT "TO CHA
   NGE COLORS."
5Ø PRINT:PRINT"PRESS <ENTER> TO START."
6Ø E$=INKEY$:IF E$="" THEN 6Ø
7Ø IF ASC(E$)<>13 THEN 6Ø
8Ø CLS
9Ø P=24Ø:C=4
1ØØ R=128+16*(C-1)+15
11Ø IF R=127 THEN R=128
12Ø PRINT@P,CHR$(128);:PRINT@P,CHR$(R);
13Ø E$=INKEY$:IF E$=""THEN 12Ø
14Ø E=ASC(E$)
15Ø IF E<>1Ø THEN 18Ø
16Ø IF P+32>51Ø THEN 12Ø
17Ø P=P+32:GOTO 12Ø
18Ø IF E<>94 THEN 21Ø
19Ø IF P-32<Ø THEN 12Ø
2ØØ P=P-32:GOTO 12Ø
```

```
21Ø IF E<>8 THEN 24Ø
22Ø IF P-1<Ø THEN 12Ø
23Ø P=P-1:GOTO 12Ø
24Ø IF E<>9 THEN 27Ø
25Ø IF P+1>51Ø THEN 12Ø
26Ø P=P+1:GOTO 12Ø
27Ø IF E<>32 THEN 12Ø
28Ø C=C-1:IF C<Ø THEN C=4
29Ø GOTO 1ØØ
3ØØ END
```

Doodle works like this on the Model I:

Line	Function
20	Clear screen.
30–40	PRINT instructions.
50–70	Wait for beginning key to be pressed.
80	Clear screen.
90	Initialize starting coordinates X and Y.
100–110	Blink square while waiting for input.
120–240	Depending on key pressed, change screen position.
250	END.

Program 4-6. Doodle

TRS-80 Model I

```
2Ø CLS
3Ø PRINT "PRESS THE ARROW KEYS TO DRAW."
4Ø PRINT:PRINT:PRINT:PRINT
5Ø PRINT:PRINT "PRESS <ENTER> TO START."
6Ø E$=INKEY$:IF E$="" THEN 6Ø
7Ø IF ASC(E$)<>13 THEN 6Ø
8Ø CLS
9Ø X=64:Y=23
1ØØ RESET(X,Y):SET(X,Y)
11Ø E$=INKEY$:IF E$="" THEN 1ØØ
12Ø E=ASC(E$)
13Ø IF E<>91 THEN 16Ø
14Ø IF Y-1<Ø THEN 1ØØ
15Ø Y=Y-1:GOTO 1ØØ
16Ø IF E<>8 THEN 19Ø
17Ø IF X-1<Ø THEN 1ØØ
18Ø X=X-1:GOTO 1ØØ
19Ø IF E<>9 THEN 22Ø
2ØØ IF X+1>127 THEN 1ØØ
21Ø X=X+1:GOTO 1ØØ
22Ø IF E<>1Ø THEN 1ØØ
23Ø IF Y+1>47 THEN 1ØØ
24Ø Y=Y+1:GOTO 1ØØ
25Ø END
```

Here's how the TI-99/4A version works:

Line	Function
110	Clear screen.
120–130	PRINT instructions.
140–160	Wait for key to be pressed.
170	Clear screen.
180–200	Define 16 colors.
210–230	Define variables. X and Y are screen positions, and C is color number.
240	Calculate character number R.
250–270	Blink square while waiting for input.
280–430	Depending on arrow key pressed, change screen position.
440–480	If space bar is pressed, change colors.
490–530	Check edge condition, then branch to line 250 to draw new square.
540	END.

Program 4-7. Doodle

TI-99/4A

```
11Ø CALL CLEAR
12Ø PRINT "PRESS ARROW KEYS TO DRAW."
13Ø PRINT ::"PRESS THE SPACE BAR"::"TO CHANGE COLO
    RS."
14Ø PRINT :::"PRESS <ENTER> TO START."
15Ø CALL KEY(Ø,K,S)
16Ø IF K<>13 THEN 15Ø
17Ø CALL CLEAR
18Ø FOR I=1 TO 16
19Ø CALL COLOR(I,I,I)
2ØØ NEXT I
21Ø X=12
22Ø Y=15
23Ø C=16
24Ø R=31+8*C
25Ø CALL HCHAR(X,Y,8)
26Ø CALL HCHAR(X,Y,R)
27Ø CALL KEY(Ø,K,S)
28Ø IF K<>69 THEN 32Ø
29Ø DX=-1
3ØØ DY=Ø
31Ø GOTO 49Ø
32Ø IF K<>68 THEN 36Ø
33Ø DX=Ø
34Ø DY=1
35Ø GOTO 49Ø
```

```
360 IF K<>88 THEN 400
370 DX=1
380 DY=0
390 GOTO 490
400 IF K<>83 THEN 440
410 DX=0
420 DY=-1
430 GOTO 490
440 IF K<>32 THEN 250
450 C=C-1
460 IF C>0 THEN 240
470 C=16
480 GOTO 240
490 X=X+DX
500 Y=Y+DY
510 X=INT(24*((X-1)/24-INT((X-1)/24)))+1
520 Y=INT(32*((Y-1)/32-INT((Y-1)/32)))+1
530 GOTO 250
540 END
```

Color Code

Computers: VIC-20; TRS-80 Color Computer; TI-99/4A

"Color Code" is a graphic guessing game, similar to the popular game "Mastermind." The computer chooses four colors and places them in a certain order. Each position may be one of the particular computer's available colors. You then try to guess the computer's color code by pressing the color numbers shown at the top of the screen in the correct order. If you make a mistake before you press the fourth color number, you may erase your guess and start over.

The computer checks your guess. You'll see one kind of marker for each correct color that is in the correct position and a different marker for a color that is in the series but is *not* in the correct position. The markers shown are not in any particular order — you may not know which color goes with which marker. Try to guess the color code in as few tries as possible. If you don't discover the color code before you reach the bottom of the screen, the computer will show you the correct answer.

The VIC version of Color Code works like this:

Line	Function
10–16	PRINT instructions; initialize sound variables.
18	Initialize color numbers.
20–24	Wait for user to press f1 key.
26	PRINT color numbers and color across top of screen.

28–30	Draw playing board.
32	Choose series of four colors.
34	Initialize N for number of tries.
36	Label player's guess.
38–46	Receive user's four guesses of color numbers.
48–54	Check for correct color in correct position.
56–60	Check for correct color in any other position.
62	Increment number of guesses; return for next try.
64–66	Show correct answer.
68	Play tune for correct answer.
70–76	PRINT option to try again and branch appropriately.
78	Subroutine for sound delay loop.
80	Clear screen and END.

Program 4-8. Color Code

VIC-20

(In this program, make sure to use abbreviations for all BASIC keywords in the lines marked with an asterisk (*). If you don't, the program will not run correctly. Refer to Appendix C, "BASIC Keyword Abbreviations," for the proper abbreviations.)

```
 10 PRINT"{CLR}":PRINTTAB(6);"{BLU}COLOR CODE":PRIN
    T"{DOWN}GUESS THE CORRECT"
*12 PRINT"SEQUENCE OF FOUR":PRINT"COLORS.":PRINT"
    {DOWN}GUESS BY PRESSING":PRINT"FOUR COLOR NUMBE
    RS."
 14 PRINT"PRESS Ø TO ERASE.":PRINT"{2 DOWN}{GRN} Z
    {SPACE}{BLK}RIGHT COLOR ONLY"
 16 PRINT"{DOWN}{RED} S {BLK}RIGHT COLOR IN":PRINT"
    {3 SPACES}CORRECT POSITION":S=36876:POKE36878,1
    5
 18 Z(1)=1:Z(2)=3:Z(3)=6:Z(4)=7:Z(5)=8
 2Ø PRINT"{4 DOWN}{GRN}PRESS {BLU}F1{GRN} TO START.
    "
 22 GETE$:IFE$=""THEN22
 24 IFASC(E$)<>133THEN22
 26 PRINT"{CLR}{BLU}1{BLK}⁅K⁆{BLU}3{RED}⁅K⁆
    {BLU}6{GRN}⁅K⁆{BLU}7⁅K⁆8{YEL}⁅K⁆{BLU}
    {3 SPACES}Ø←"
 28 FORP=7725TO8165STEP22:FORA=PTOP+8:POKEA,1Ø2:POK
    EA+3Ø72Ø,Ø:NEXTA,P
 3Ø FORP=7748TO8144STEP44:FORA=PTOP+6STEP2:POKEA,16
    Ø:POKEA+3Ø72Ø,1:NEXTA,P
 32 FORA=1TO4:J=INT(5*RND(Ø)+1):X(A)=Z(J):NEXT
 34 N=1
 36 M=77Ø2+44*N:L=M+3Ø72Ø:POKEM,48+N:POKEL,6
 38 FORJ=1TO4:POKES,225:GOSUB78:POKES,Ø
 4Ø GETE$:IFE$<>"1"ANDE$<>"3"ANDE$<>"6"ANDE$<>"7"AN
    DE$<>"8"ANDE$<>"Ø"THEN4Ø
 42 IFE$<>"0"THEN46
 44 POKEL+2,1:POKEL+4,1:POKEL+6,1:GOTO38
 46 Y(J)=VAL(E$):POKEL+J*2,Y(J)-1:NEXTJ
 48 J=1
 50 FORA=1TO4:W(A)=X(A):IFX(A)<>Y(A)THEN54
 52 POKEM+12+J,83:POKEL+12+J,2:Y(A)=0:W(A)=9:J=J+1
 54 NEXTA:IFJ=5THEN68
 56 FORA=1TO4:FORA1=1TO4:IFW(A)<>Y(A1)THEN60
 58 POKEM+12+J,90:POKEL+12+J,5:Y(A1)=0:W(A)=9:J=J+1
 60 NEXTA1,A
 62 N=N+1:IFN<10THEN36
 64 POKES,159:GOSUB78:POKES,135:GOSUB78:POKES,0
 66 L=38862:FORA=1TO4:POKEL+2*A,X(A)-1:NEXT:GOTO70
*68 FORA=1TO3:POKES,195:GOSUB78:POKES,207:GOSUB78:P
    OKES,215:GOSUB78:POKES,225:GOSUB78:GOSUB78:POKE
    S,0:NEXTA
 70 PRINT"{22 DOWN}TRY AGAIN? (Y/N)";
 72 GETE$:IFE$="Y"THEN26
 74 IFE$="N"THEN80
 76 GOTO72
 78 FORT=1TO99:NEXT:RETURN
 80 PRINT"{CLR}":END
```

Color Code works differently on the Color Computer:

Line	Function
10–60	PRINT instructions.
70–80	Define character numbers for colors C(I).
90–110	Wait for user to press ENTER to start.
120–130	PRINT color numbers and colors across top of screen.
140–160	Draw playing board.
170	Choose series of four colors; initialize N as number of tries.
180–280	Receive user's four guesses of color numbers.
290–320	Check for correct color in correct position.
330–350	Check for correct color in any other position.
360	Increment number of guesses; return for next try.
370–380	Show correct answer.
390	Play tune for correct answer.
400–420	PRINT option to try again and branch appropriately.
430	END.

Program 4-9. Color Code

TRS-80 Color Computer

```
10 CLS:PRINT TAB(6);"** COLOR CODE **"
20 PRINT:PRINT "GUESS THE CORRECT SEQUENCE"
```

```
3Ø PRINT "OF FOUR COLORS.":PRINT"GUESS BY PRESSING
    FOUR"
4Ø PRINT "COLOR NUMBERS.":PRINT"PRESS THE BACK ARR
   OW TO ERASE."
5Ø PRINT:PRINT"!  RIGHT COLOR ONLY"
6Ø PRINT"*  RIGHT COLOR, CORRECT POSITION"
7Ø FOR I=2 TO 8:READC(I):NEXT
8Ø DATA 159,175,191,2Ø7,223,239,255
9Ø PRINT:PRINT"PRESS <ENTER> TO START."
1ØØ E$=INKEY$:IF E$=""THEN1ØØ
11Ø IF ASC(E$)<>13 THEN 1ØØ
12Ø CLS
13Ø FOR I=2TO8:PRINT STR$(I);CHR$(C(I));:NEXT
14Ø A$=CHR$(128):B$=CHR$(131)
15Ø FOR I=65 TO 417 STEP 32:PRINT@I,A$+B$+A$+B$+A$
    +B$+A$+B$+A$:NEXT
16Ø PRINT @449,A$+A$+A$+A$+A$+A$+A$+A$+A$
17Ø FOR A=1 TO 4:X(A)=RND(6)+2:NEXT:N=1
18Ø FOR A=1 TO 4
19Ø SOUND 2ØØ,1
2ØØ M=32+32*N+A*2
21Ø PRINT@M,CHR$(128);:PRINT@M,CHR$(131);
22Ø E$=INKEY$:IF E$=""THEN 21Ø
23Ø IF ASC(E$)<>8 THEN 25Ø
24Ø PRINT @34+32*N,B$+A$+B$+A$+B$;:A=1:GOTO 19ØGOT
    O 19Ø
25Ø IF ASC(E$)<5Ø THEN 21Ø
26Ø IF ASC(E$)>56 THEN 21Ø
27Ø E=VAL(E$):Y(A)=E:PRINT@M,CHR$(C(E)-12);
28Ø NEXT A
29Ø J=1:M=M+5
3ØØ FORA=1TO4:W(A)=X(A):IFX(A)<>Y(A)THEN 32Ø
31Ø PRINT@M,"*";:Y(A)=Ø:W(A)=9:J=J+1:M=M+1
32Ø NEXTA:IFJ=5THEN39Ø
33Ø FORA=1TO4:FORA1=1TO4:IFW(A)<>Y(A1)THEN35Ø
34Ø PRINT@M,"!";:Y(A1)=Ø:W(A)=9:J=J+1:M=M+1
35Ø NEXTA1,A
36Ø N=N+1:IFN<12THEN18Ø
37Ø SOUND125,3:SOUND89,3
38Ø FORA=1TO4:PRINT@416+2*A,CHR$(C(X(A))-12);:NEXT
    :GOTO4ØØ
39Ø FORA=1TO3:SOUND89,1:SOUND125,1:SOUND147,1:SOUN
    D176,2:NEXTA
4ØØ PRINT@48Ø,"TRY AGAIN? (Y/N)";
41Ø E$=INKEY$:IFE$="Y"THEN12Ø
42Ø IFE$<>"N"THEN41Ø
43Ø END
```

The same game program, this time on the TI-99/4A, works like this:

Line	Function
100–200	PRINT instructions and define graphics characters.
210–320	Define graphics, colors, and character numbers for colors.
330–350	Wait for user to press any key to start.
360–390	Draw playing board.
400–430	Choose series of four colors.
440–500	PRINT color numbers and colors across top of screen.
510–520	Initialize I for number of tries.
530	If number of guesses is greater than 9, end.
540–550	Label number of try.
560–710	Receive user's four guesses of color numbers.
720–810	Check for correct color in correct position.
820–910	Check for correct color in any other position.
920	Return for next try.
930–980	Show correct answer.
990–1030	Play tune for correct answer.
1040–1070	PRINT option to try again and branch appropriately.
1080–1090	Clear screen and END.

Program 4-10. Color Code

TI-99/4A

```
100 CALL CLEAR
110 PRINT TAB(7);"COLOR CODE"
120 CALL CHAR(93,"3C7EFFFFFFFF7E3C")
130 PRINT ::::"GUESS THE CORRECT SEQUENCE"
140 PRINT "OF FOUR COLORS."
150 CALL CHAR(120,"3C7EFFFFFFFF7E3C")
160 PRINT :"GUESS BY PRESSING THE"
170 PRINT "COLOR NUMBERS."
180 CALL COLOR(12,16,1)
190 PRINT :::CHR$(93);" RIGHT COLOR AND POSITION"
200 PRINT :CHR$(120);" RIGHT COLOR ONLY"
210 FOR I=1 TO 3
220 CALL CHAR((11+I)*8,"0")
230 CALL CHAR((11+I)*8+1,"FFFFFFFFFFFFFFFF")
240 NEXT I
250 FOR I=1 TO 6
260 READ C(I)
270 NEXT I
280 DATA 96,97,104,105,112,113
290 CALL COLOR(9,3,16)
300 CALL COLOR(10,5,12)
310 CALL COLOR(11,7,14)
```

```
320 CALL CHAR(92,"FFFFFFFFFFFFFFFF")
330 PRINT :::"PRESS ANY KEY TO START.";
340 CALL KEY(Ø,K,S)
350 IF S<1 THEN 34Ø
360 CALL CLEAR
370 FOR I=9 TO 17
380 CALL VCHAR(4,I,92,2Ø)
390 NEXT I
400 FOR I=1 TO 4
410 RANDOMIZE
420 A(I)=INT(RND*6+1)
430 NEXT I
440 FOR I=1 TO 6
450 CALL HCHAR(1,3*I-1,48+I)
460 CALL HCHAR(1,3*I,C(I))
470 NEXT I
480 FOR I=1 TO 7
490 CALL HCHAR(1,21+I,ASC(SEG$("Ø-ERASE",I,1)))
500 NEXT I
510 I=Ø
520 I=I+1
530 IF I>9 THEN 93Ø
540 IA=3+2*I
550 CALL HCHAR(IA,8,48+I)
560 FOR J=1 TO 4
570 JA=8+2*J
580 CALL SOUND(1ØØ,1497,2)
590 CALL KEY(Ø,K,S)
600 CALL HCHAR(IA,JA,92)
610 CALL HCHAR(IA,JA,32)
620 IF S<1 THEN 59Ø
630 IF (K>47)*(K<55)<>1 THEN 59Ø
640 IF K<>48 THEN 69Ø
650 FOR K=J TO 1 STEP -1
660 CALL HCHAR(IA,8+K*2,92)
670 NEXT K
680 GOTO 56Ø
690 B(J)=K-48
700 CALL HCHAR(IA,JA,C(K-48))
710 NEXT J
720 L=1
730 FOR J=1 TO 4
740 D(J)=A(J)
750 IF A(J)<>B(J)THEN 8ØØ
760 CALL HCHAR(IA,21+L,93)
770 B(J)=Ø
780 D(J)=8
790 L=L+1
800 NEXT J
```

```
81Ø IF L=5 THEN 99Ø
82Ø FOR J=1 TO 4
83Ø FOR J1=1 TO 4
84Ø IF D(J)<>B(J1)THEN 9ØØ
85Ø CALL HCHAR(IA,21+L,12Ø)
86Ø B(J1)=Ø
87Ø D(J)=8
88Ø L=L+1
89Ø IF L=5 THEN 52Ø
9ØØ NEXT J1
91Ø NEXT J
92Ø GOTO 52Ø
93Ø CALL SOUND(15Ø,33Ø,2)
94Ø CALL SOUND(15Ø,262,2)
95Ø FOR J=1 TO 4
96Ø CALL HCHAR(23,8+2*J,C(A(J)))
97Ø NEXT J
98Ø GOTO 1Ø4Ø
99Ø FOR J=1 TO 3
1ØØØ CALL SOUND(1ØØ,262,2)
1Ø1Ø CALL SOUND(1ØØ,33Ø,2)
1Ø2Ø CALL SOUND(1ØØ,392,2)
1Ø3Ø NEXT J
1Ø4Ø PRINT "TRY AGAIN? (Y/N)";
1Ø5Ø CALL KEY(Ø,K,S)
1Ø6Ø IF K=89 THEN 36Ø
1Ø7Ø IF K<>78 THEN 1Ø5Ø
1Ø8Ø CALL CLEAR
1Ø9Ø END
```

Matching Shapes

Computers: VIC-20; TRS-80 Color Computer; MC-10; TRS-80 Model I;
TI-99/4A

"Matching Shapes" uses the color and graphics capabilities of the computer to present a teaching game for preschool children. This program, a drill of matching shapes, begins with a random shape displayed. Four more random shapes are then drawn. One of the four matches the first shape. Press the matching shape's number as your answer. The answer must be correct to continue. Remember to look at the shape, not at the color. After ten problems you have the option to try again.

Each computer's program version is listed below.

Here's how the VIC version works:

Line	Function
2	Initialize variables Y for color and T for sound, turn volume on, and branch past subroutines.

4–24 Subroutines to draw shapes.
26 Subroutine for sound delay.
28–30 Read graphics coordinate position W from DATA.
32–74 Perform quiz ten times.
34 PRINT question.
36 Randomly choose shape N and color C, set X equal to the coordinate for drawing, and draw shape to be matched.
38 PRINT answer numbers.
40 Randomly choose answer number.
42–58 Randomly draw four shapes for the four possible answers. Color C is chosen randomly. X is the coordinate for drawing. Lines 44–46 draw the correct answer shape.
60–62 Play a tone, then wait for any key to be pressed.
64–66 If answer is incorrect, play sound effect and return for another response.
68–74 If answer is correct, PRINT three hearts under the correct shape, play a musical sound effect, and return for next problem.
76–80 PRINT option to try again and branch appropriately.
82 Clear screen and END.

Program 4-11. Matching Shapes

VIC-20

(In this program, make sure to use abbreviations for all BASIC keywords in the lines marked with an asterisk (*). If you don't, the program will not run correctly. Refer to Appendix C, "BASIC Keyword Abbreviations," for the proper abbreviations.)

```
 2 Y=3Ø72Ø:T=36876:POKE36878,15:GOTO28
 4 POKEX,16Ø:POKEX+21,16Ø:POKEX+22,16Ø:POKEX+23,16Ø
   :POKEX+44,16Ø
 6 Z=X+Y:POKEZ,C:POKEZ+21,C:POKEZ+22,C:POKEZ+23,C:P
   OKEZ+44,C:RETURN
*8 FORI=X-1TOX+1:POKEI,16Ø:POKEI+22,16Ø:POKEI+44,16
   Ø:POKEI+Y,C:POKEI+Y+22,C:POKEI+Y+44,C:NEXT:RETUR
   N
 1Ø POKEX,233:POKEX+1,223:POKEX+21,233:POKEX+22,16Ø
    :POKEX+23,16Ø:POKEX+24,223
 12 Z=X+Y:POKEZ,C:POKEZ+1,C:POKEZ+21,C:POKEZ+22,C:P
    OKEZ+23,C:POKEZ+24,C:RETURN
 14 POKEX,223:POKEX+22,16Ø:POKEX+23,223:POKEX+44,16
    Ø:POKEX+45,16Ø:POKEX+46,223
 16 Z=X+Y:POKEZ,C:POKEZ+22,C:POKEZ+23,C:POKEZ+44,C:
    POKEZ+45,C:POKEZ+46,C:RETURN
 18 POKEX-1,233:POKEX,16Ø:POKEX+1,223:POKEX+21,16Ø:
    POKEX+22,16Ø:POKEX+23,16Ø
```

```
20 POKEX+43,95:POKEX+44,16Ø:POKEX+45,1Ø5:Z=X+Y
22 FORI=Z-1TOZ+1:POKEI,C:POKEI+22,C:POKEI+44,C:NEX
   T:RETURN
24 FORI=XTOX+66STEP22:POKEI,16Ø:POKEI+Y,C:POKEI+1,
   16Ø:POKEI+Y+1,C:NEXT:RETURN
26 FORD=1TO6Ø:NEXT:RETURN
28 FORI=ØTO4:READW(I):NEXT
3Ø DATA7754,7947,7952,7957,7962
32 FORP=1TO1Ø
34 PRINT"{CLR}{DOWN}WHICH SHAPE MATCHES?"
36 N=INT(6*RND(Ø)+1):X=W(Ø):C=INT(6*RND(Ø)+2):ONN
   {SPACE}GOSUB4,8,1Ø,14,18,24
38 PRINT"{9 DOWN}{3 SPACES}1{4 SPACES}2{4 SPACES}3
   {4 SPACES}4"
4Ø A=INT(4*RND(Ø)+1)
42 FORJ=1TO4:X=W(J):C=INT(6*RND(Ø)+2)
44 IFJ<>A THEN48
46 K(J)=N:ONN GOSUB4,8,1Ø,14,18,24:GOTO58
48 B=INT(6*RND(Ø)+1):IFB=N THEN48
5Ø IFJ=1THEN56
52 FORL=1TOJ-1:IFB=K(L)THEN48
54 NEXTL
56 K(J)=B:ONB GOSUB4,8,1Ø,14,18,24
58 NEXTJ
6Ø POKET,225:GOSUB26:POKET,Ø
62 GETE$:IFE$=""THEN62
64 IFVAL(E$)=A THEN68
66 POKET,159:GOSUB26:POKET,135:GOSUB26:POKET,Ø:GOT
   O62
68 PRINTTAB(A*5-3);"{5 DOWN}{RED}SSS{BLU}"
7Ø POKET,195:GOSUB26:POKET,2Ø7:GOSUB26:POKET,215:G
   OSUB26
72 POKET,225:GOSUB26:GOSUB26:POKET,Ø
74 NEXTP
76 PRINT"{2 DOWN}TRY AGAIN? (Y/N)"
78 GETE$:IFE$="Y"THEN32
8Ø IFE$<>"N"THEN78
82 PRINT"{CLR}":END
```

The Color Computer version looks a bit different:

Line	Function
20	Branch past subroutines.
30–410	Subroutine to draw shapes.
420–430	Initialize coordinates for positions of shapes.
440	Perform quiz ten times.
450	Clear screen and PRINT question.
460	Randomly choose shape number N and color C and draw shape to be matched.

470 Randomly choose answer number.
480–570 Randomly draw four shapes for the four possible answers. Color C is chosen randomly. X and Y are coordinates for drawing.
580–590 Play a tone, then wait for any key to be pressed.
600–610 If answer is incorrect, play musical tone and return for another response.
620–640 If answer is correct, PRINT message, play musical effect, and return for next problem.
650–670 PRINT option to try again and branch appropriately.
680 Clear screen and END.

Program 4-12. Matching Shapes

TRS-80 Color Computer; MC-10

```
20 GOTO420
30 L1=X:L2=X+1
40 FOR J=Y TO Y+4 STEP 2
50 FOR I=L1 TO L2
60 SET(I,J,C):SET(I,J+1,C)
70 NEXT I
80 L1=L1-2:L2=L2+2
90 NEXT J
100 RETURN
110 FOR J=Y TO Y+5
120 FOR I=X-4 TO X+5
130 SET(I,J,C)
140 NEXT I,J
150 RETURN
160 L2=X+1
170 FOR J=Y TO Y+4 STEP 2
180 FOR I=X TO L2
190 SET(I,J,C):SET(I,J+1,C)
200 NEXT I
210 L2=L2+2
220 NEXT J
230 RETURN
240 FOR I=X TO X+1
250 FOR J=Y TO Y+5
260 SET(I,J,C)
270 NEXT J,I
280 FOR I=X-4 TO X+5
290 SET(I,Y+2,C):SET(I,Y+3,C)
300 NEXT I
310 RETURN
320 FOR J=Y TO Y+4 STEP 2
330 SET(X,J,C):SET(X,J+1,C)
340 SET(X+1,J,C):SET(X+1,J+1,C)
```

```
350 X=X+2
360 NEXT J:X=X-2
370 SET(X,Y,C):SET(X,Y+1,C)
380 SET(X+1,Y,C):SET(X+1,Y+1,C)
390 SET(X-4,Y+4,C):SET(X-3,Y+4,C)
400 SET(X-4,Y+5,C):SET(X-3,Y+5,C)
410 RETURN
420 FORI=0TO4:READA(I),B(I):NEXT
430 DATA30,2,8,18,24,18,38,18,54,18
440 FORP=1TO10
450 CLS:PRINT"WHICH SHAPE MATCHES?"
460 N=RND(5):X=A(0):Y=B(0):C=RND(3)+1:ONN GOSUB30,
    110,160,240,320
470 D=RND(4)
480 FORM=1TO4:X=A(M):Y=B(M):C=RND(3)+1
490 IFM<>D THEN510
500 K(M)=N:ONN GOSUB30,110,160,240,320:GOTO560
510 E=RND(5):IFE=N THEN510
520 IFM=1THEN550
530 FORL=1TOM-1:IFE=K(L)THEN510
540 NEXTL
550 K(M)=E:ONE GOSUB30,110,160,240,320
560 NEXTM
570 PRINT@388,"1{7 SPACES}2{7 SPACES}3{6 SPACES}4"
580 SOUND200,1
590 E$=INKEY$:IFE$=""THEN590
600 IFVAL(E$)=D THEN 620
610 SOUND125,1:SOUND89,1:GOTO590
620 PRINT@448,"CORRECT!"
630 SOUND89,1:SOUND125,1:SOUND147,1:SOUND176,2
640 NEXTP
650 PRINT:PRINT"TRY AGAIN? (Y/N)";
660 E$=INKEY$:IFE$="Y"THEN440
670 IFE$<>"N"THEN660
680 CLS:END
```

The Model I version of Matching Shapes works like this:

Line	**Function**
20	Branch past subroutines.
30–390	Draw shapes.
420–540	Draw title screen.
550–570	Wait for ENTER key to be pressed.
580	Perform quiz ten times.
590	Clear screen and PRINT question.
600	Randomly choose shape number N and draw shape to be matched.
610	Randomly choose answer number.

620–710 Randomly draw four shapes for the four possible answers, making sure none of the shapes are the same.
720 Wait for a key to be pressed.
730–740 If answer is incorrect, PRINT message and return for another response.
750–770 If answer is correct, PRINT message, blink arrows under correct answer, and return for next problem.
780–800 PRINT option to try again and branch appropriately.
810 Clear screen and END.

Program 4-13. Matching Shapes

TRS-80 Model I

```
2Ø GOTO 42Ø
3Ø L1=X+4:L2=X+5
4Ø FOR J=Y TO Y+6 STEP 3
5Ø FOR I=L1 TO L2
6Ø SET(I,J):SET(I,J+1):SET(I,J+2)
7Ø NEXT I
8Ø L1=L1-2:L2=L2+2
9Ø NEXT J
1ØØ RETURN
11Ø FOR J=Y TO Y+8
12Ø FOR I=X TO X+15
13Ø SET(I,J)
14Ø NEXTI,J
15Ø RETURN
16Ø L2=X
17Ø FOR J=Y TO Y+8
18Ø FOR I=X TO L2
19Ø SET(I,J)
2ØØ NEXT I
21Ø L2=L2+1
22Ø NEXT J
23Ø RETURN
24Ø FOR I=X+4 TO X+7
25Ø FOR J=Y TO Y+8
26Ø SET(I,J)
27Ø NEXT J,I
28Ø FOR I=X TO X+11
29Ø SET(I,Y+3):SET(I,Y+4):SET(I,Y+5)
3ØØ NEXT I
31Ø RETURN
32Ø FOR J=Y+2 TO Y+5:FOR I=X TO X+13:SET(I,J):NEXT
     I,J
33Ø FOR I=X+2 TO X+11:SET(I,Y+1):SET(I,Y+6):NEXTI
34Ø FOR I=X+4 TO X+9:SET(I,Y):SET(I,Y+7):NEXT I
35Ø RETURN
```

```
360 I=X:II=X+14
370 FOR J=Y TO Y+7:SET(I,J):SET(I+1,J):I=I+2
380 SET(II,J):SET(II+1,J):II=II-2:NEXT J
390 RETURN
420 CLS
430 PRINT@144,"MATCHING SHAPES"
460 X=18:Y=15:GOSUB30
480 X=70:Y=30:GOSUB110
500 X=76:Y=15:GOSUB160
520 X=36:Y=27:GOSUB240
530 X=46:Y=12:GOSUB320
540 X=98:Y=21:GOSUB360
550 PRINT@960,L"PRESS <ENTER> TO START";
560 E$=INKEY$:IFE$=""THEN560
570 IF ASC(E$)<>13 THEN560
580 FOR P=1 TO 10
590 CLS:PRINT"WHICH SHAPE MATCHES?"
600 N=RND(6):X=56:Y=9:ONN GOSUB30,110,160,240,320,
    360
610 D=RND(4)
620 FOR M=1TO4:X=28*M-14:Y=30
630 IF M<>D THEN650
640 K(M)=N:ONN GOSUB30,110,160,240,320,360:GOTO700
650 E=RND(6):IFE=N THEN650
660 IF M=1 THEN690
670 FORL=1TOM-1:IFE=K(L) THEN650
680 NEXTL
690 K(M)=E:ON E GOSUB30,110,160,240,320,360
700 NEXTM
710 PRINT@842,"1{13 SPACES}2{13 SPACES}3{13 SPACES}4"
720 E$=INKEY$:IFE$=""THEN720
730 IFVAL(E$)=D THEN750
740 PRINT@896,E$;" NO, TRY AGAIN.":GOTO720
750 PRINT@896,"CORRECT{9 SPACES}"
760 V=896+14*D-6:FORL=1TO100:PRINT@V,"^^^^^";:PRIN
    T@V,"{5 SPACES}";:NEXTL
770 NEXTP
780 PRINT:PRINT"TRY AGAIN? (Y/N)"
790 E$=INKEY$:IFE$="Y"THEN580
800 IFE$<>"N"THEN790
810 CLS:END
```

The Matching Shapes program works like this on the TI-99/4A:

Line	Function
110	Define R for use in choosing a random color.
120	Branch past subroutines.
130–470	Subroutines to draw shapes.
480–600	Display title while defining graphic shapes.

610 Perform quiz ten times.
620 Clear screen.
630–680 Randomly choose shape number N and draw shape to be matched.
690 Randomly choose answer number D.
700–870 Randomly draw four shapes for the four possible answers. The correct shape is in position D, and none of the shapes may be the same.
880 PRINT question.
890–910 Play a tone, then wait for a key to be pressed.
920–950 If answer is incorrect, play notes and return for another response.
960–1020 If answer is correct, PRINT three asterisks under correct choice, erase question, play musical effect, and return for next problem.
1030–1060 PRINT option to try again and branch appropriately.
1070–1080 Clear screen and END.

Program 4-14. Matching Shapes

TI-99/4A

```
11Ø DEF R=INT(12*RND+5)
12Ø GOTO 48Ø
13Ø CALL VCHAR(X,Y,96,3)
14Ø CALL HCHAR(X+1,Y-1,96,3)
15Ø RETURN
16Ø FOR I=X TO X+2
17Ø CALL HCHAR(I,Y-1,1Ø4,3)
18Ø NEXT I
19Ø RETURN
2ØØ CALL HCHAR(X,Y-1,113)
21Ø CALL VCHAR(X,Y,112,3)
22Ø CALL HCHAR(X,Y+1,114)
23Ø CALL HCHAR(X+1,Y-1,112,3)
24Ø CALL HCHAR(X+2,Y-1,116)
25Ø CALL HCHAR(X+2,Y+1,115)
26Ø RETURN
27Ø CALL HCHAR(X,Y-1,121)
28Ø CALL HCHAR(X+1,Y-1,12Ø)
29Ø CALL HCHAR(X+1,Y,121)
3ØØ CALL HCHAR(X+2,Y-1,12Ø,2)
31Ø CALL HCHAR(X+2,Y+1,121)
32Ø RETURN
33Ø CALL VCHAR(X,Y,128,3)
34Ø CALL HCHAR(X,Y-1,129)
35Ø CALL HCHAR(X,Y+1,13Ø)
```

```
360 CALL HCHAR(X+1,Y+1,131)
370 CALL HCHAR(X+2,Y+1,132)
380 CALL HCHAR(X+2,Y-1,133)
390 CALL HCHAR(X+1,Y-1,134)
400 RETURN
410 CALL HCHAR(X,Y,137)
420 CALL HCHAR(X+1,Y-1,138)
430 CALL VCHAR(X+1,Y,136,2)
440 CALL HCHAR(X+1,Y+1,139)
450 CALL HCHAR(X+2,Y-1,140)
460 CALL HCHAR(X+2,Y+1,141)
470 RETURN
480 CALL CLEAR
490 PRINT TAB(7);"MATCHING SHAPES"::::::::::::
500 FOR C=96 TO 136 STEP 8
510 CALL CHAR(C,"FFFFFFFFFFFFFFFF")
520 NEXT C
530 FOR I=1 TO 16
540 READ C,C$
550 CALL CHAR(C,C$)
560 NEXT I
570 DATA 113,00030F1F3F3F7F7F,114,00C0F0F8FCFCFEFE
    ,115,FEFEFCFCF8F0C,116,7F7F3F3F1F0F03,121,80C0
    E0F0F8FCFEFF
580 DATA 129,0101030307070F0F,130,8080C0C0E0E0F0F,
    131,F8F8FCFCFCFCF8F8,132,F0F0E0E0C0C0808,133
590 DATA 0F0F070703030101,134,1F1F3F3F3F3F1F1F,137
    ,18183C3C7E7EFFFF,138,0101030307070F0F,139,808
    0C0C0E0E0F0F
600 DATA 140,1F1F3F3F7F7FFFFF,141,F8F8FCFCFEFEFFFF
610 FOR P=1 TO 10
620 CALL CLEAR
630 X=11
640 Y=15
650 RANDOMIZE
660 N=IN=(6*RND+1)
670 CALL COLOR(8+N,R,1)
680 ON N GOSUB 130,160,200,270,330,410
690 D=INT(4*RND+1)
700 FOR M=1 TO 4
710 X=21
720 Y=7*M-1
730 IF M<>D THEN 770
740 J(M)=N
750 ON N GOSUB 130,160,200,270,330,410
760 GOTO 860
770 E=INT(6*RND+1)
780 IF E=N THEN 770
790 IF M=1 THEN 830
```

```
8ØØ FOR L=1 TO M-1
81Ø IF E=J(L)THEN 77Ø
82Ø NEXT L
83Ø J(M)=E
84Ø CALL COLOR(8+E,R,1)
85Ø ON E GOSUB 13Ø,16Ø,2ØØ,27Ø,33Ø,41Ø
86Ø NEXT M
87Ø PRINT TAB(4);"1{6 SPACES}2{6 SPACES}3
    {6 SPACES}4"
88Ø PRINT ::"WHICH SHAPE MATCHES?":::
89Ø CALL SOUND(15Ø,1497,2)
9ØØ CALL KEY(Ø,K,S)
91Ø IF S<1 THEN 9ØØ
92Ø IF K=48+D THEN 96Ø
93Ø CALL SOUND(1ØØ,33Ø,2)
94Ø CALL SOUND(1ØØ,262,2)
95Ø GOTO 9ØØ
96Ø CALL HCHAR(19,7*D-2,42,3)
97Ø CALL HCHAR(21,3,32,2Ø)
98Ø CALL SOUND(1ØØ,262,2)
99Ø CALL SOUND(1ØØ,33Ø,2)
1ØØØ CALL SOUND(1ØØ,392,2)
1Ø1Ø CALL SOUND(2ØØ,523,2)
1Ø2Ø NEXT P
1Ø3Ø PRINT ::"TRY AGAIN? (Y/N)";
1Ø4Ø CALL KEY(Ø,K,S)
1Ø5Ø IF K=89 THEN 6ØØ
1Ø6Ø IF K<>78 THEN 1Ø4Ø
1Ø7Ø CALL CLEAR
1Ø8Ø END
```

Counting Shapes

Computers: VIC-20; TRS-80 Color Computer; MC-10; TRS-80 Model I;
TI-99/4A

"Counting Shapes" is a program designed for preschool children or for anyone learning the concept of counting. One to nine shapes are drawn on the screen. You simply press the correct *number* key as an answer. To continue, the answer must be correct. After ten problems, you have the option to try again.

This program, although simpler than "Matching Shapes," does use some of the same shapes. All three versions of Counting Shapes work in the same general way.

The instructions to draw the shapes are in subroutines at the beginning of each program. The coordinates for the nine possible positions are then defined. A random color C is chosen for the objects, a random number N for the number of objects, and a random number S for the shape. A FOR-NEXT loop draws the correct number of objects.

Program 4-15. Counting Shapes

VIC-20

(In this program, make sure to use abbreviations for all BASIC keywords in the lines marked with an asterisk (*). If you don't, the program will not run correctly. Refer to Appendix C, "BASIC Keyword Abbreviations," for the proper abbreviations.)

```
 2 Y=3Ø72Ø:T=36876:POKE36878,15:GOTO28
 4 POKEX,16Ø:POKEX+21,16Ø:POKEX+22,16Ø:POKEX+23,16Ø
   :POKEX+44,16Ø
 6 Z=X+Y:POKEZ,C:POKEZ+21,C:POKEZ+22,C:POKEZ+23,C:P
   OKEZ+44,C:RETURN
*8 FORI=X-1TOX+1:POKEI,16Ø:POKEI+22,16Ø:POKEI+44,16
   Ø:POKEI+Y,C:POKEI+Y+22,C:POKEI+Y+44,C:NEXT:RETUR
   N
 1Ø POKEX,233:POKEX+1,223:POKEX+21,233:POKEX+22,16Ø
    :POKEX+23,16Ø:POKEX+24,223
 12 Z=X+Y:POKEZ,C:POKEZ+1,C:POKEZ+21,C:POKEZ+22,C:P
    OKEZ+23,C:POKEZ+24,C:RETURN
 14 POKEX,223:POKEX+22,16Ø:POKEX+23,223:POKEX+44,16
    Ø:POKEX+45,16Ø:POKEX+46,223
 16 Z=X+Y:POKEZ,C:POKEZ+22,C:POKEZ+23,C:POKEZ+44,C:
    POKEZ+45,C:POKEZ+46,C:RETURN
 18 POKEX-1,233:POKEX,16Ø:POKEX+1,223:POKEX+21,16Ø:
    POKEX+22,16Ø:POKEX+23,16Ø
 2Ø POKEX+43,95:POKEX+44,16Ø:POKEX+45,1Ø5:Z=X+Y
 22 FORI=Z-1TOZ+1:POKEI,C:POKEI+22,C:POKEI+44,C:NEX
    T:RETURN
 24 FORI=XTOX+66STEP22:POKEI,16Ø:POKEI+Y,C:POKEI+1,
    16Ø:POKEI+Y+1,C:NEXT:RETURN
 26 FORD=1TO6Ø:NEXTD:RETURN
 28 PRINT"{CLR}"
 3Ø PRINT"{3 RIGHT}{BLU}COUNTING SHAPES"
 32 C=INT(6*RND(Ø)+2):X=7794:GOSUB4
 34 C=INT(6*RND(Ø)+2):X=7843:GOSUB8
 36 C=INT(6*RND(Ø)+2):X=785Ø:GOSUB1Ø
 38 C=INT(6*RND(Ø)+2):X=7925:GOSUB14
 4Ø C=INT(6*RND(Ø)+2):X=7954:GOSUB18
 42 C=INT(6*RND(Ø)+2):X=7981:GOSUB24
 44 FORI=1TO9:READW(I):NEXT
 46 DATA7771,7842,78Ø3,7875,7924,7973,7956,8ØØ5,8Ø8
    Ø
 48 PRINT"{17 DOWN}PRESS RETURN TO START.";
 5Ø GETE$:IFE$=""THEN5Ø
 52 IFASC(E$)<>13THEN5Ø
 54 FOR K=1 TO 1Ø
 56 PRINT"{CLR}"
 58 C=INT(6*RND(Ø)+2)
 6Ø N=INT(9*RND(Ø)+1)
 62 S=INT(6*RND(Ø)+1)
 64 FORL=1TON:X=W(L)
 66 ON S GOSUB4,8,1Ø,14,18,24
 68 NEXTL
 7Ø PRINT"{HOME}HOW MANY OBJECTS?"
 72 POKET,225:FORD=1TO1ØØ:NEXTD:POKET,Ø
 74 GETE$:IFE$=""THEN74
 76 PRINTE$
 78 IFASC(E$)=48+N THEN86
 8Ø POKET,159:GOSUB26:POKET,135:GOSUB26:POKET,Ø
 82 PRINT"{HOME}{DOWN} {LEFT}";
 84 GOTO72
*86 POKET,195:GOSUB26:POKET,2Ø7:GOSUB26:POKET,215:G
    OSUB26:POKET,225:GOSUB26:GOSUB26:POKET,Ø
 88 NEXT K
 9Ø PRINT"{CLR}"
 92 PRINT"TRY AGAIN? (Y/N)"
 94 GETE$:IFE$="Y"THEN54
 96 IFE$<>"N" THEN 94
 98 PRINT"{CLR}"
 1ØØ END
```

Program 4-16. Counting Shapes

TRS-80 Color Computer; MC-10

```
2Ø GOTO 42Ø
3Ø L1=X:L2=X+1
4Ø FOR J=Y TO Y+4 STEP 2
5Ø FOR I=L1 TO L2
6Ø SET(I,J,C):SET(I,J+1,C)
7Ø NEXT I
8Ø L1=L1-2:L2=L2+2
9Ø NEXT J
1ØØ RETURN
11Ø FOR J=Y TO Y+5
12Ø FOR I=X TO X+9
13Ø SET(I,J,C)
14Ø NEXT I,J
15Ø RETURN
16Ø L2=X+1
17Ø FOR J=Y TO Y+6 STEP 2
18Ø FOR I=X TO L2
19Ø SET(I,J,C):SET(I,J+1,C)
2ØØ NEXT I
21Ø L2=L2+2
22Ø NEXT J
23Ø RETURN
24Ø FOR I=X TO X+1
25Ø FOR J=Y TO Y+5
```

```
26Ø SET(I,J,C)
27Ø NEXT J,I
28Ø FOR I=X-4 TO X+5
29Ø SET(I,Y+2,C):SET(I,Y+3,C)
3ØØ NEXT I
31Ø RETURN
32Ø FOR J=Y TO Y+4 STEP 2
33Ø SET(X,J,C):SET(X,J+1,C)
34Ø SET(X+1,J,C):SET(X+1,J+1,C)
35Ø X=X+2
36Ø NEXT J:X=X-2
37Ø SET(X,Y,C):SET(X,Y+1,C)
38Ø SET(X+1,Y,C):SET(X+1,Y+1,C)
39Ø SET(X-4,Y+4,C):SET(X-3,Y+4,C)
4ØØ SET(X-4,Y+5,C):SET(X-3,Y+5,C)
41Ø RETURN
42Ø CLS
43Ø PRINT @72,"COUNTING SHAPES"
44Ø C=RND(3)+1
45Ø X=1Ø:Y=8:GOSUB 3Ø
46Ø C=RND(3)+1
47Ø X=2Ø:Y=2Ø:GOSUB 11Ø
48Ø C=RND(3)+1
49Ø X=44:Y=16:GOSUB 16Ø
5ØØ C=RND(3)+1
51Ø X=3Ø:Y=1Ø:GOSUB 24Ø
52Ø C=RND(3)+1
53Ø X=52:Y=8:GOSUB 32Ø
54Ø FOR I=1 TO 9:READ X1(I),Y1(I):NEXT I
55Ø DATA 8,4,16,14,28,6,38,14,44,4,52,16,4,2Ø,28,2
    2,48,24
56Ø PRINT @48Ø,"PRESS ENTER TO START.";
57Ø E$=INKEY$:IF E$="" THEN 57Ø
58Ø IF ASC(E$)<>13 THEN 57Ø
59Ø FOR K=1 TO 1Ø
6ØØ CLS
61Ø C=RND(3)+1
62Ø N=RND(9)
63Ø S=RND(5)
64Ø FOR L=1 TO N
65Ø X=X1(L):Y=Y1(L)
66Ø ON S GOSUB 3Ø,11Ø,16Ø,24Ø,32Ø
67Ø NEXT L
68Ø PRINT "HOW MANY OBJECTS?"
69Ø SOUND 227,2
7ØØ E$=INKEY$:IF E$="" THEN 7ØØ
71Ø PRINT @2Ø,E$
72Ø IF ASC(E$)=48+N THEN 76Ø
73Ø SOUND 125,1:SOUND 89,1
```

```
74Ø PRINT @2Ø," "
75Ø GOTO 7ØØ
76Ø SOUND 89,1:SOUND 125,1:SOUND 147,1:SOUND 176,2
77Ø NEXT K
78Ø CLS
79Ø PRINT @258,"TRY AGAIN? (Y/N)"
8ØØ E$=INKEY$:IF E$="Y" THEN 59Ø
81Ø IF E$<>"N" THEN 8ØØ
82Ø CLS
83Ø END
```

Program 4-17. Counting Shapes

TRS-80 Model I

```
2Ø GOTO 42Ø
3Ø L1=X+4:L2=X+5
4Ø FOR J=Y TO Y+6 STEP 3
5Ø FOR I=L1 TO L2
6Ø SET(I,J):SET(I,J+1):SET(I,J+2)
7Ø NEXT I
8Ø L1=L1-2:L2=L2+2
9Ø NEXT J
1ØØ RETURN
11Ø FOR J=Y TO Y+8
12Ø FOR I=X TO X+15
13Ø SET(I,J)
14Ø NEXTI,J
15Ø RETURN
16Ø L2=X
17Ø FOR J=Y TO Y+8
18Ø FOR I=X TO L2
19Ø SET(I,J)
2ØØ NEXT I
21Ø L2=L2+1
22Ø NEXT J
23Ø RETURN
24Ø FOR I=X+4 TO X+7
25Ø FOR J=Y TO Y+8
26Ø SET(I,J)
27Ø NEXT J,I
28Ø FOR I=X TO X+11
29Ø SET(I,Y+3):SET(I,Y+4):SET(I,Y+5)
3ØØ NEXT I
31Ø RETURN
32Ø FOR J=Y+2 TO Y+5:FOR I=X TO X+13:SET(I,J
    ):NEXT I,J
33Ø FOR I=X+2 TO X+11:SET(I,Y+1):SET(I,Y+6):
    NEXTI
34Ø FOR I=X+4 TO X+9:SET(I,Y):SET(I,Y+7):NEX
    T I
```

```
350 RETURN
360 I=X:II=X+14
370 FOR J=Y TO Y+7:SET(I,J):SET(I+1,J):I=I+2
380 SET(II,J):SET(II+1,J):II=II-2:NEXT J
390 RETURN
420 CLS
430 PRINT@144,"COUNTING SHAPES"
460 X=18:Y=15:GOSUB30
480 X=70:Y=30:GOSUB110
500 X=76:Y=15:GOSUB160
520 X=36:Y=27:GOSUB240
530 X=46:Y=12:GOSUB320
540 X=98:Y=21:GOSUB360
550 FORI=1TO9:READX1(I),Y1(I):NEXTI
560 DATA46,12,70,30,98,13,36,27,20,17,98,27,
    76,15,2,9,8,33
570 PRINT@960,"PRESS <ENTER> TO START.";
580 E$=INKEY$:IFE$=""THEN580
590 IF ASC(E$)<>13THEN580
600 FOR K=1 TO 10
610 CLS
630 N=RND(9)
640 S=RND(6)
650 FOR L=1TON
660 X=X1(L):Y=Y1(L)
670 ON S GOSUB30,110,160,240,320,360
680 NEXTL
690 PRINT"HOW MANY OBJECTS?"
710 E$=INKEY$:IFE$=""THEN710
720 PRINT@20,E$
730 IF ASC(E$)=48+N THEN770
740 PRINT@84,"SORRY, TRY AGAIN.":FOR II=1TO5
    00:NEXTII
750 PRINT@20," ":PRINT@84,"{17 SPACES}"
760 GOTO710
770 FOR II=1TO10:PRINT@84,"{7 SPACES}":PRINT
    @84,"CORRECT":NEXTII
775 FOR II=1TO200:E$=INKEY$:IFE$<>""THEN780
776 NEXTII
780 NEXTK
790 CLS
795 PRINT@212,"GOOD WORK!"
800 PRINT@516,"TRY AGAIN? (Y/N)"
810 E$=INKEY$:IFE$="Y" THEN600
820 IFE$<>"N"THEN810
830 CLS
840 END
```

Program 4-18. Counting Shapes

TI-99/4A

```
110 GOTO 470
120 CALL VCHAR(X,Y,96,3)
130 CALL HCHAR(X+1,Y-1,96,3)
140 RETURN
150 FOR I=X TO X+2
160 CALL HCHAR(I,Y-1,104,3)
170 NEXT I
180 RETURN
190 CALL HCHAR(X,Y-1,113)
200 CALL VCHAR(X,Y,112,3)
210 CALL HCHAR(X,Y+1,114)
220 CALL HCHAR(X+1,Y-1,112,3)
230 CALL HCHAR(X+2,Y-1,116)
240 CALL HCHAR(X+2,Y+1,115)
250 RETURN
260 CALL HCHAR(X,Y-1,121)
270 CALL HCHAR(X+1,Y-1,120)
280 CALL HCHAR(X+1,Y,121)
290 CALL HCHAR(X+2,Y-1,120,2)
300 CALL HCHAR(X+2,Y+1,121)
310 RETURN
320 CALL VCHAR(X,Y,128,3)
330 CALL HCHAR(X,Y-1,129)
340 CALL HCHAR(X,Y+1,130)
350 CALL HCHAR(X+1,Y+1,131)
360 CALL HCHAR(X+2,Y+1,132)
370 CALL HCHAR(X+2,Y-1,133)
380 CALL HCHAR(X+1,Y-1,134)
390 RETURN
400 CALL HCHAR(X,Y,137)
410 CALL HCHAR(X+1,Y-1,138)
420 CALL VCHAR(X+1,Y,136,2)
430 CALL HCHAR(X+1,Y+1,139)
440 CALL HCHAR(X+2,Y-1,140)
450 CALL HCHAR(X+2,Y+1,141)
460 RETURN
470 CALL CLEAR
480 PRINT TAB(7);"COUNTING SHAPES"::::::::::::
490 FOR C=96 TO 136 STEP 8
500 CALL CHAR(C,"FFFFFFFFFFFFFFFF")
510 NEXT C
520 FOR I=1 TO 16
530 READ C,C$
540 CALL CHAR(C,C$)
550 NEXT I
```

```
560 DATA 113,ØØØ3ØF1F3F3F7F7F,114,ØØCØFØF8FCFCFEFE
    ,115,FEFEFCFCF8FØC,116,7F7F3F3F1FØFØ3,121,8ØCØ
    EØFØF8FCFEFF
570 DATA 129,Ø1Ø1Ø3Ø3Ø7Ø7ØFØF,13Ø,8Ø8ØCØCØEØEØFØF,
    131,F8F8FCFCFCFCF8F8,132,FØFØEØEØCØCØ8Ø8,133
58Ø DATA ØFØFØ7Ø7Ø3Ø3Ø1Ø1,134,1F1F3F3F3F3F1F1F,137
    ,18183C3C7E7EFFFF,138,Ø1Ø1Ø3Ø3Ø7Ø7ØFØF,139,8Ø8
    ØCØCØEØEØFØF
59Ø DATA 14Ø,1F1F3F3F7F7FFFFF,141,F8F8FCFCFEFEFFFF
6ØØ FOR I=1 TO 9
61Ø READ X1(I),Y1(I)
62Ø NEXT I
63Ø DATA 8,5,9,11,7,17,1Ø,23,14,7,13,16,14,28,19,1
    1,18,22
64Ø FOR K=1 TO 1Ø
65Ø CALL CLEAR
66Ø C=INT(12*RND+5)
67Ø N=INT(9*RND+1)
68Ø S=INT(6*RND+1)
69Ø CALL COLOR(S+8,C,1)
7ØØ FOR L=1 TO N
71Ø X=X1(L)
72Ø Y=Y1(L)
73Ø ON S GOSUB 12Ø,15Ø,19Ø,26Ø,32Ø,4ØØ
74Ø NEXT L
75Ø PRINT "HOW MANY OBJECTS?"
76Ø CALL SOUND(1ØØ,1497,2)
77Ø CALL KEY(Ø,KE,ST)
78Ø IF ST<1 THEN 77Ø
79Ø CALL HCHAR(23,22,KE)
8ØØ IF KE=48+N THEN 85Ø
81Ø CALL SOUND(1ØØ,33Ø,2)
82Ø CALL SOUND(1ØØ,262,2)
83Ø CALL HCHAR(23,22,32)
84Ø GOTO 76Ø
85Ø CALL SOUND(1ØØ,262,2)
86Ø CALL SOUND(1ØØ,33Ø,2)
87Ø CALL SOUND(1ØØ,392,2)
88Ø CALL SOUND(2ØØ,523,2)
89Ø NEXT K
9ØØ CALL CLEAR
91Ø PRINT "TRY AGAIN? (Y/N)"
92Ø CALL KEY(Ø,KE,ST)
93Ø IF KE=89 THEN 64Ø
94Ø IF KE<>78 THEN 92Ø
95Ø CALL CLEAR
96Ø END
```

Locating Points

Computers: VIC-20; TRS-80 Color Computer; MC-10; TI-99/4A

A computer can easily use a tutorial program because it can respond differently to each different answer. Color and graphics make learning even more fun. "Locating Points" teaches you how to locate points on a rectangular coordinate grid. This concept is often taught at the third-grade level, and reviewed in algebra classes.

A random example point is first displayed, with the coordinates labeled. You can view as many examples as you want. Next, a randomly chosen point is shown, and you enter your estimate of its coordinates. You can then move to the last part of the program, where you're given a point's coordinates. Using the proper movement keys (each program tells you what they are), you move the point until it is at the desired coordinates. The computer will tell you if you've placed the point correctly.

This is how the VIC version works:

Line	Function
2	PRINT title screen.
4	Define volume and sound.
6–7	Define string variables for grid. Delay loop.
8–26	Draw grid and show example point.
30–51	Present problem to give coordinates for given point.
52–54	PRINT instructions.
55–74	Present problem to locate point with given coordinates.
75–79	PRINT option to have another problem, start over, or end program. The program then shifts appropriately.
80–83	Label point and draw yellow lines from point to axis.
84	Calculate graphics memory location.
86	Play music for incorrect answer.
88	Subroutine for correct answer.
89	Delay loop for music.
90–92	PRINT grid.
94	Clear keyboard buffer.
96–99	PRINT "PRESS RETURN" and wait for response.
100	Clear screen and END.

Program 4-19. Locating Points

VIC-20

(In this program, make sure to use abbreviations for all BASIC keywords in the lines marked with an asterisk (*). If you don't, the program will not run correctly. Refer to Appendix C, "BASIC Keyword Abbreviations," for the proper abbreviations.)

```
2 PRINT"{CLR}{5 DOWN}{3 SPACES}{BLK}LOCATING POINT
  S"
```

```
 4 POKE36878,15:S=36876
 6 A$="{GRN}{2 SPACES}-{2 SPACES}-{2 SPACES}-
   {2 SPACES}-{2 SPACES}-{2 SPACES}-":B$="{GRN}+
   **+**+**+**+**+***"
 7 FORI=1TO3000:NEXT
 8 GOSUB90
 10 PRINT"A POINT IS LOCATED BY AN X-COORDINATE AND
     A Y-COORDINATE (X,Y)."
 12 X=INT(4*RND(0))+1:Y=INT(3*RND(0))+1
 14 GOSUB84:POKEC,81:POKEC1,2
 18 GOSUB80
 22 PRINT:PRINT"{GRN}ANOTHER EXAMPLE? (Y/N){BLU}";:
    GOSUB94
 24 GETR$:IFR$="Y"THEN8
 26 IFR$<>"N"THEN24
 30 GOSUB90:PRINT"WHERE IS THE POINT?":PRINT:PRINT"
    {BLK}({RED}?{BLK},{RED}?{BLK})";
 31 X=INT(6*RND(0)):Y=INT(5*RND(0)):GOSUB84:POKEC,8
    1:POKEC1,2
 32 GOSUB94
 33 GETR$:IFR$=""THEN33
 34 IFASC(R$)<48ORASC(R$)>57THEN33
 35 POKE8099,ASC(R$):POKE38819,2:X$=R$
 36 GETR$:IFR$=""THEN36
 37 IFASC(R$)<48ORASC(R$)>57THEN36
 38 POKE8101,ASC(R$):POKE38821,2:Y$=R$
 39 IFVAL(X$)<>XTHEN42
 40 IFVAL(Y$)=YTHEN48
*42 GOSUB86:PRINT"{2 SPACES}{BLK}(";RIGHT$(STR$(X),
    1);",";RIGHT$(STR$(Y),1);")":GOSUB80:GOSUB96:GO
    TO30
 48 GOSUB88:PRINT"{GRN}PRESS":PRINT"1 SAME TYPE PRO
    BLEM{3 SPACES}2 CONTINUE PROGRAM";:GOSUB94
 50 GETR$:IFR$="1"THEN30
 51 IFR$<>"2"THEN50
 52 PRINT"{CLR}{BLU}YOU WILL BE GIVEN THE COORDINAT
    ES (X,Y).":PRINT:PRINT"PRESS F1 TO MOVE UP"
*53 PRINT:PRINTTAB(6);"F3 TO MOVE LEFT":PRINT:PRINT
    TAB(6);"F5 TO MOVE RIGHT":PRINTTAB(6);"F7 TO MO
    VE DOWN"
*54 PRINT:PRINT"MOVE THE POINT TO THE CORRECT LOCAT
    ION,{5 SPACES}THEN PRESS <RETURN>.{3 DOWN}":GOS
    UB96
 55 GOSUB90:X=INT(6*RND(0)):Y=INT(5*RND(0)):A=0:B=0
 56 PRINT"LOCATE (";RIGHT$(STR$(X),1);",";RIGHT$(ST
    R$(Y),1);")";:GOSUB94
 57 POKES,231:GOSUB89:P=7990-B*66+3*A:POKEP,81:POKE
    P+30720,3
 58 GETR$:IFR$=""THEN58
 59 IFASC(R$)=13THEN73
 61 IFASC(R$)<>133THEN64
 62 B=B+1:IFB>4THENB=4
 63 GOTO72
 64 IFASC(R$)<>134THEN67
 65 A=A-1:IFA<ØTHENA=Ø
 66 GOTO72
 67 IFASC(R$)<>135THEN7Ø
 68 A=A+1:IFA>5THENA=5
 69 GOTO72
 7Ø IFASC(R$)<>136THEN58
 71 B=B-1:IFB<ØTHENB=Ø
 72 POKEP,91:POKEP+3Ø72Ø,5:GOTO57
 73 GOSUB84:IFP=CTHEN75
 74 GOSUB86:POKEC,81:POKEC1,2:GOSUB96:GOTO52
*75 GOSUB88:PRINT"{GRN}PRESS":PRINT" 1 SAME TYPE PR
    OBLEM{3 SPACES}2 START PROGRAM OVER{2 SPACES}3
    {SPACE}END PROGRAM";:GOSUB94
 76 GETR$:IFR$="1"THEN52
 77 IFR$="2"THEN8
 78 IFR$="3"THEN1ØØ
 79 GOTO76
 8Ø FORI=C1+1TOC1+5:POKEI,2:NEXT
 81 POKEC+1,4Ø:POKEC+2,X+48:POKEC+3,44:POKEC+4,Y+48
    :POKEC+5,41
 82 FORI=C1+22TO3871Ø+X*3STEP22:POKEI,7:NEXT
 83 FORI=C1-1TOC1-X*3STEP-1:POKEI,7:NEXT:RETURN
 84 C=799Ø-Y*66+3*X:C1=C+3Ø72Ø:RETURN
 86 POKES,159:GOSUB89:POKES,135:GOSUB89:RETURN
*88 PRINT"{RED} CORRECT!":POKES,195:GOSUB89:POKES,2
    Ø7:GOSUB89:POKES,215:GOSUB89:POKES,225:GOSUB89
 89 FORI=1TO15Ø:NEXT:POKES,Ø:RETURN
*9Ø PRINT"{CLR}{BLK}{2 SPACES}Y":PRINTA$:PRINT"
    {BLK}4";B$:PRINTA$:PRINTA$:PRINT" {BLK}3";B$:PR
    INTA$:PRINTA$:PRINT" {BLK}2";B$:PRINTA$:PRINTA$
    :PRINT" {BLK}1";B$:PRINTA$:PRINTA$
 92 PRINT" {BLK}Ø";B$;"{BLK}X{2 SPACES}Ø{2 SPACES}1
    {2 SPACES}2{2 SPACES}3{2 SPACES}4{2 SPACES}5
    {BLU}":PRINT:RETURN
 94 FORI=1TO1Ø:GETR$:NEXT:RETURN
 96 PRINT:PRINT:PRINT"{GRN}PRESS RETURN";:GOSUB94
 97 GETR$:IFR$=""THEN97
 98 IFASC(R$)<>13THEN97
 99 RETURN
 1ØØ PRINT"{CLR}{BLU}":END
```

Here's how the program works on the Color Computer:

Line	Function
1	Branch past subroutines.
2–3	PRINT grid.
4–7	PRINT "PRESS ENTER" and wait for response.
8–9	Choose point and calculate graphics print position.
10	Calculate coordinates to SET point.
11	Play music for incorrect response.
12	Play music for correct response.
13–14	PRINT title screen and define string variables for grid graphics.
15–18	Draw grid with example point.
19–30	Present problem to find coordinates for given point.
31–32	PRINT instructions.
33–52	Present problem to locate point with given coordinates.
53–57	PRINT option to have another problem, start over, or end program; program branches appropriately. Clear screen and END.

Program 4-20. Locating Points

TRS-80 Color Computer; MC-10

```
1 GOTO13
2 CLS:PRINT" Y";A$:FORX=3TO1STEP-1:PRINT"  ";RIGHT$
  (STR$(X),1);B$:PRINT"   ";A$:PRINT"   ";A$:NEXT
3 PRINT@321,"Ø";B$:PRINT@354,"Ø{4 SPACES}1
  {4 SPACES}2{4 SPACES}3{4 SPACES}4{4 SPACES}5  X"
  :PRINT:RETURN
4 PRINT@496,"PRESS <ENTER>";
5 R$=INKEY$:IFR$=""THEN5
6 IFASC(R$)<>13THEN5
7 RETURN
8 X=RND(5)
9 Y=RND(3):A=322-96*Y+X*5:PRINT@A,CHR$(159);CHR$(1
  59);:RETURN
1Ø B=4+X*1Ø:C=2Ø-6*Y:FORI=C+2 TO2Ø:SET(B,I,4):SET(
   B+2,I,4):NEXT:FORI=B-2 TO4STEP-1:SET(I,C,4):NEX
   T:RETURN
11 SOUND125,2:SOUND89,2:RETURN
12 SOUND89,2:SOUND125,2:SOUND147,2:SOUND176,4:RETU
   RN
13 CLS:PRINT@197,"** LOCATING POINTS **":C$=CHR$(1
   75):D$=C$+C$+C$+C$+C$:E$="{3 SPACES}"+C$+C$:A$=
   E$:B$=D$
14 FORI=1TO4:A$=A$+E$:B$=B$+D$:NEXT:A$=C$+C$+A$:B$
   =C$+C$+B$+C$+C$:FORD=1TO2ØØØ:NEXT
```

```
15 GOSUB2:PRINT"A POINT HAS AN X-COORDINATE
   {5 SPACES}AND A Y-COORDINATE (X,Y).":X=RND(4):G
   OSUB9:X$=RIGHT$(STR$(X),1)
16 Y$=RIGHT$(STR$(Y),1):PRINT@A+2,"("+X$+","+Y$+")
   ";:GOSUB1Ø:PRINT@48Ø,"ANOTHER EXAMPLE? (Y/N)";
17 R$=INKEY$:IFR$="Y"THEN15
18 IFR$<>"N"THEN17
19 GOSUB2:GOSUB8:PRINT@416,"WHAT ARE THE COORDINAT
   ES?":PRINT"{5 SPACES}(?,?)"
2Ø U$=INKEY$:IFU$=""THEN2Ø
21 PRINT@454,U$;
22 V$=INKEY$:IFV$=""THEN22
23 PRINT@456,V$;
24 IFX<>VAL(U$) THEN26
25 IFY=VAL(V$) THEN27
26 GOSUB11:GOSUB1Ø:PRINT@46Ø,"LOCATION IS (";RIGHT
   $(STR$(X),1);",";RIGHT$(STR$(Y),1);")";:GOSUB4:
   GOTO19
27 GOSUB12:PRINT@46Ø,"CORRECT!":GOSUB4:PRINT@496,L
   "{13 SPACES}";
28 PRINT@416,"PRESS 1 FOR SAME TYPE PROBLEM
   {9 SPACES}2 TO CONTINUE PROGRAM"
29 R$=INKEY$:IFR$="1"THEN19
3Ø IFR$<>"2"THEN29
31 CLS:PRINT@66,"YOU WILL BE GIVEN THE":PRINT"  CO
   ORDINATES.":PRINT@162,"USE THE ARROW KEYS TO MO
   VE"
32 PRINT"  THE POINT TO THE CORRECT":PRINT"  LOCAT
   ION, THEN PRESS <ENTER>.":GOSUB4
33 CLS:GOSUB2:X=RND(5):Y=RND(3):PRINT@416,"LOCATE
   (";RIGHT$(STR$(X),1);",";RIGHT$(STR$(Y),1);")":
   D=4:E=2Ø:B=4:C=2Ø
34 SET(B,C,2):SET(B+2,C,2):SOUND227,2
35 R$=INKEY$:IFR$=""THEN35
36 IFASC(R$)=13THEN49
37 IFASC(R$)<>9THEN4Ø
38 D=B+1Ø:IFD>54THEND=54
39 GOTO48
4Ø IFASC(R$)<>8THEN43
41 D=B-1Ø:IFD<4THEND=4
42 GOTO48
43 IFASC(R$)<>94THEN46
44 E=C-6:IFE<2THENE=2
45 GOTO48
46 IFASC(R$)<>1ØTHEN35
47 E=C+6:IFE>2ØTHENE=2Ø
48 SET(B,C,3):SET(B+2,C,3):B=D:C=E:GOTO34
49 D=4+X*1Ø:E=2Ø-6*Y:IFB<>D THEN51
5Ø IFC=E THEN52
```

```
51 GOSUB11:SET(D,E,4):SET(D+2,E,4):GOSUB1Ø:GOSUB4:
   GOTO33
52 GOSUB12:PRINT@448,"CORRECT!":GOSUB4
53 PRINT@496,"{13 SPACES}";:PRINT@416,"PRESS 1 FOR
   SAME TYPE PROBLEM":PRINT"{6 SPACES}2 TO START P
   ROGRAM OVER":PRINT@486,"3 TO END PROGRAM";
54 R$=INKEY$:IFR$="1"THEN33
55 IFR$="2"THEN15
56 IFR$<>"3"THEN54
57 CLS:END
```

The TI-99/4A version is slightly diffrent, although the display is much the same as the other computers' programs.

Line	Function
100	Define random function.
110–400	PRINT title screen and define graphics characters.
410–430	Play music for incorrect answer.
440–480	Play music for correct answer.
490–550	PRINT grid.
560–600	PRINT "PRESS ENTER" and wait for response.
610–770	Present random example of a point with coordinates.
790–1150	Present problem to give coordinates for given point.
1160–1200	PRINT instructions.
1210–1710	Present problem to locate point with given coordinates.
1720–1770	Give option to repeat problem or end, and then branch appropriately.
1780–1830	Choose random point.
1840–1870	Draw vertical red line from point to X-axis.
1880–1910	Draw horizontal red line from point to Y-axis.
1920	END.

Program 4-21. Locating Points

TI-99/4A

```
100 DEF R(N)=INT(N*RND+1)
110 CALL CLEAR
120 PRINT "{4 SPACES}*******************":"
    {4 SPACES}*";TAB(23);"*":"{4 SPACES}* LOCATING
     POINTS *"
130 PRINT "{4 SPACES}*";TAB(23);"*":"{4 SPACES}***
    ****************":::::::
140 A$="1818181818181818"
150 B$="181818FFFF181818"
160 C$="ØØØØØØFFFF"
```

```
170 FOR C=96 TO 112 STEP 8
180 CALL CHAR(C,A$)
190 CALL CHAR(C+1,B$)
200 CALL CHAR(C+2,C$)
210 NEXT C
220 CALL CHAR(120,"183C7EFFFF7E3C18")
230 CALL CHAR(128,"183C7EFFFF7E3C18")
240 CALL CHAR(129,"ØØØØØØØØØ30C30C")
250 CALL CHAR(130,"Ø30C30C")
260 CALL CHAR(94,"ØØ1Ø2828444482FE")
270 CALL COLOR(1Ø,5,1)
280 CALL COLOR(11,1Ø,1)
290 CALL COLOR(12,11,1)
300 CALL COLOR(13,7,1)
310 CALL CHAR(14Ø,"1Ø1Ø1Ø1Ø1Ø1Ø1")
320 CALL CHAR(141,"ØØØØØØFF")
330 CALL CHAR(142,"1Ø1Ø1ØF")
340 CALL COLOR(14,13,1)
350 A$="`  h  h  h  h  h  h  h"
360 B$="ajjijjijjijjijjijji"
370 C$="abbabbabbabbabbabbabb"
380 CALL CLEAR
390 CALL COLOR(2,2,1)
400 GOTO 61Ø
410 CALL SOUND(1ØØ,33Ø,2)
420 CALL SOUND(1ØØ,262,2)
430 RETURN
440 CALL SOUND(1ØØ,262,2)
450 CALL SOUND(1ØØ,33Ø,2)
460 CALL SOUND(1ØØ,392,2)
470 CALL SOUND(2ØØ,523,2)
480 RETURN
490 CALL CLEAR
500 PRINT "{4 SPACES}Y":"{4 SPACES}";A$:"
    {4 SPACES}";A$:"{3 SPACES}4";B$:"{4 SPACES}";A
    $:"{4 SPACES}";A$:"{3 SPACES}3";B$
510 PRINT "{4 SPACES}";A$:"{4 SPACES}";A$:"
    {3 SPACES}2";B$
520 PRINT "{4 SPACES}";A$:"{4 SPACES}";A$:"
    {3 SPACES}1";B$:"{4 SPACES}";A$:"{4 SPACES}";A
    $:"{3 SPACES}Ø";C$
530 PRINT "{4 SPACES}Ø  1  2  3  4  5  6  7":::
540 CALL HCHAR(2Ø,31,88)
550 RETURN
560 PRINT TAB(16);"PRESS <ENTER>";
570 CALL KEY(Ø,K,S)
580 IF K<>13 THEN 57Ø
590 CALL HCHAR(24,18,32,13)
600 RETURN
```

```
610 GOSUB 490
620 PRINT "THE LOCATION OF A POINT IS":"GIVEN BY I
    TS X-COORDINATE"
630 PRINT "AND Y-COORDINATE (X,Y)"
640 RANDOMIZE
650 X=R(5)
660 GOSUB 1790
670 GOSUB 1840
680 CALL HCHAR(Y1,X1+2,40)
690 CALL HCHAR(Y1,X1+3,48+X)
700 CALL HCHAR(Y1,X1+4,44)
710 GOSUB 1880
720 CALL HCHAR(Y1,X1+5,48+Y)
730 CALL HCHAR(Y1,X1+6,41)
740 PRINT :"WANT ANOTHER EXAMPLE? (Y/N)";
750 CALL KEY(0,K,S)
760 IF K=89 THEN 610
770 IF K<>78 THEN 750
780 CALL CLEAR
790 PRINT "YOU WILL BE SHOWN A POINT.")::"PRESS THE
     NUMBER OF THE"
800 PRINT :"X-COORDINATE THEN THE"
810 PRINT :"NUMBER OF THE Y-COORDINATE.":::::::
820 GOSUB 560
830 CALL CLEAR
840 GOSUB 490
850 PRINT :::
860 RANDOMIZE
870 GOSUB 1780
880 CALL HCHAR(21,7,40)
890 CALL HCHAR(21,9,44)
900 CALL HCHAR(21,11,41)
910 CALL KEY(0,K,S)
920 CALL HCHAR(21,8,63)
930 CALL HCHAR(21,8,32)
940 IF S<1 THEN 910
950 CALL HCHAR(21,8,K)
960 X2=K
970 CALL KEY(0,K,S)
980 CALL HCHAR(21,10,63)
990 CALL HCHAR(21,10,32)
1000 IF S<1 THEN 970
1010 CALL HCHAR(21,10,K)
1020 Y2=K
1030 IF X2<>X+48 THEN 1100
1040 IF Y2<>Y+48 THEN 1100
1050 GOSUB 440
1060 PRINT "PRESS":"1 FOR SAME TYPE PROBLEM":"2 TO
      CONTINUE PROGRAM";
```

```
1Ø7Ø CALL KEY(Ø,K,S)
1Ø8Ø IF K=49 THEN 83Ø
1Ø9Ø IF K=5Ø THEN 116Ø ELSE 1Ø7Ø
11ØØ GOSUB 41Ø
111Ø GOSUB 184Ø
112Ø GOSUB 188Ø
113Ø PRINT "THE CORRECT ANSWER IS (";STR$(X);",";S
     TR$(Y);")"
114Ø GOSUB 56Ø
115Ø GOTO 83Ø
116Ø CALL CLEAR
117Ø PRINT "NOW YOU WILL BE GIVEN THE"::"COORDINAT
     ES."
118Ø PRINT :"USE THE ARROW KEYS TO MOVE"::"THE POI
     NT TO THE CORRECT"
119Ø PRINT :"PLACE, THEN PRESS <ENTER>.":::::
12ØØ GOSUB 56Ø
121Ø CALL CLEAR
122Ø GOSUB 49Ø
123Ø RANDOMIZE
124Ø X=R(7)
125Ø Y=R(4)
126Ø X1=7+3*X
127Ø Y1=17-3*Y
128Ø PRINT :"PLOT (";STR$(X);",";STR$(Y);")"::
129Ø C1=97
13ØØ A=17
131Ø A1=A
132Ø B=7
133Ø B1=B
134Ø CALL HCHAR(A,B,12Ø)
135Ø CALL KEY(Ø,K,S)
136Ø IF S<1 THEN 135Ø
137Ø IF K=13 THEN 162Ø
138Ø IF K<>69 THEN 143Ø
139Ø IF A=5 THEN 135Ø
14ØØ CALL GCHAR(A-3,B,C)
141Ø A=A-3
142Ø GOTO 157Ø
143Ø IF K<>88 THEN 148Ø
144Ø IF A=17 THEN 135Ø
145Ø CALL GCHAR(A+3,B,C)
146Ø A=A+3
147Ø GOTO 157Ø
148Ø IF K<>83 THEN 153Ø
149Ø IF B=7 THEN 135Ø
15ØØ CALL GCHAR(A,B-3,C)
151Ø B=B-3
152Ø GOTO 157Ø
```

```
1530 IF K<>68 THEN 1350
1540 IF B=28 THEN 1350
1550 CALL GCHAR(A,B+3,C)
1560 B=B+3
1570 CALL HCHAR(A1,B1,C1)
1580 A1=A
1590 B1=B
1600 C1=C
1610 GOTO 1340
1620 CALL SOUND(150,1397,2)
1630 CALL GCHAR(Y1,X1,C)
1640 IF C=120 THEN 1710
1650 GOSUB 410
1660 CALL HCHAR(Y1,X1,128)
1670 GOSUB 1840
1680 GOSUB 1880
1690 GOSUB 560
1700 GOTO 1210
1710 GOSUB 440
1720 PRINT "PRESS":"1 FOR SAME TYPE PROBLEM":"2 TO
      END PROGRAM";
1730 CALL KEY(0,K,S)
1740 IF K=49 THEN 1210
1750 IF K<>50 THEN 1730
1760 CALL CLEAR
1770 STOP
1780 X=R(7)
1790 Y=R(4)
1800 X1=7+3*X
1810 Y1=17-3*Y
1820 CALL HCHAR(Y1,X1,128)
1830 RETURN
1840 FOR I=Y1+1 TO 17
1850 CALL HCHAR(I,X1,112)
1860 NEXT I
1870 RETURN
1880 FOR I=X1-1 TO 7 STEP -1
1890 CALL HCHAR(Y1,I,114)
1900 NEXT I
1910 RETURN
1920 END
```

Chapter 5
Sound

Chapter 5
Sound

Few, if any, computers produce sound effects the same way. The VIC-20, for example, uses different commands to create sound than does the Commodore 64, even though both machines are Commodore computers. Translating BASIC to duplicate sound effects is difficult because of this. You have to know how both computers produce the sound in order to translate a program from one computer's version of BASIC to another.

The programs in this chapter, "Musical Keyboard," "Treble Clef," and "Bass Clef," are all examples of computer-generated sound. Since it is so hard to translate sound commands, all three programs have been listed for each of three computers: the VIC-20, the TRS-80 Color Computer, and the TI-99/4A. By comparing the programs, you can see how the creation of sound was translated from one machine's version of BASIC to another.

Musical Keyboard

Computers: VIC-20; TRS-80 Color Computer; MC-10; TI-99/4A

This program won't teach you how to play the piano, but it will help you remember the keys. The musical keyboard displayed on the screen consists of white and black keys. The program shows how the white keys are named — with the first seven letters of the alphabet. After the letter names are shown, there is a short quiz. A key is randomly chosen, and you must press the key's letter name. If the letter pressed is correct, the correct note is played. You must answer correctly to continue. The quiz consists of ten keys.

"Musical Keyboard" on the VIC works like this:

Line	**Function**
2	Branch past subroutines.
3–16	Define letter name of note AA$, tone S to be played, and coordinate CC of key.
20–25	Draw the keyboards.
26–29	Wait for RETURN to be pressed.
30–31	PRINT title screen, define SS to play music, and turn on volume.
32–33	Play scale for title screen.
34–42	PRINT instructions.
50–60	Present quiz for ten notes.
51	Clear screen and randomly choose and draw one of two keyboards.
52–55	Randomly choose key and blink an asterisk on the key.

56–57 PRINT message, then wait for key to be pressed.
58–59 If name is incorrect, sound a noise and return for another response.
60 If name is correct, PRINT the letter name on the key, play the tone, and then go to next key.
61–63 PRINT option to try again and branch appropriately.
64 Clear screen and END.

Program 5-1. Musical Keyboard

VIC-20

(In this program, make sure to use abbreviations for all BASIC keywords in the lines marked with an asterisk (*). If you don't, the program will not run correctly. Refer to Appendix C, "BASIC Keyword Abbreviations," for the proper abbreviations.)

```
2 GOTO3Ø
3 AA$="C":S=131:CC=Ø:RETURN
4 AA$="D":S=145:CC=3:RETURN
5 AA$="E":S=158:CC=6:RETURN
6 AA$="F":S=161:IFK=1THENCC=8:RETURN
7 CC=Ø:RETURN
8 AA$="G":S=173:IFK=1THENCC=11:RETURN
9 CC=3:RETURN
1Ø AA$="A":S=181:IFK=1THENCC=14:RETURN
11 CC=6:RETURN
12 AA$="B":S=189:IFK=1THENCC=17:RETURN
13 CC=9:RETURN
14 AA$="C":S=192:CC=11:RETURN
15 AA$="D":S=2ØØ:CC=14:RETURN
16 AA$="E":S=2Ø6:CC=17:RETURN
2Ø FORKI=1TO1Ø:PRINT" {BLK}{RVS}{2 SPACES}{OFF}
   {RVS}{2 SPACES}{OFF} B̲ {RVS}{2 SPACES}{OFF}
   {RVS}{2 SPACES}{OFF} {RVS}{2 SPACES}{OFF} B̲":NE
   XT
21 FORKI=1TO3:PRINT"{2 SPACES}[G]{2 SPACES}[G]
    B̲{2 SPACES}[G]{2 SPACES}[G]{2 SPACES}[G]
    B̲":NEXT:RETURN
22 FORKI=1TO1Ø:PRINT" {BLK}{RVS}{2 SPACES}{OFF}
   {RVS}{2 SPACES}{OFF} {RVS}{2 SPACES}{OFF} B̲
   {RVS}{2 SPACES}{OFF} {RVS}{2 SPACES}{OFF} B̲":NE
   XT
23 FORKI=1TO3:PRINT"{2 SPACES}[G]{2 SPACES}[G]
   {2 SPACES}[G] B̲{2 SPACES}[G]{2 SPACES}[G]
    B":NEXT:RETURN
24 FORI=1TO7:PRINT"{2 SPACES}{RVS} {OFF} {RVS}
   {OFF} {RVS} {OFF} B̲ {RVS} {OFF} {RVS} {OFF} B̲
   {RVS} {OFF} {RVS} {OFF} {RVS} ":NEXT:RETURN
25 FORI=1TO3:PRINT"{2 SPACES}B̲ B̲ B̲ B̲ B̲ B̲ B̲ B̲ B̲ B̲":
   NEXT:PRINT"[22 Y] ":RETURN
26 PRINT"{2 DOWN}{GRN}PRESS RETURN";
```

```
 27 GETA$:IFA$=""THEN27
 28 IFASC(A$)<>13THEN27
 29 PRINT"{CLR}{BLU}":RETURN
 3Ø PRINT"{CLR}{BLU}{3 DOWN}{2 SPACES}LET'S LEARN N
    OTES":PRINT"{2 DOWN}{4 SPACES}THE KEYBOARD
    {7 DOWN}"
 31 SS=36876:POKE36878,15
 32 DATA131,145,158,161,173,181,189,192
 33 FORI=1TO8:READS:POKESS,S:FORD=1TO2ØØ:NEXTD,I:PO
    KESS,Ø
 34 PRINT"{CLR}{DOWN}{BLU}A PIANO OR ORGAN":PRINT"K
    EYBOARD HAS GROUPS{3 SPACES}OF TWO BLACK KEYS"
 35 PRINT"AND THREE BLACK KEYS. {2 DOWN}{BLK}":GOSU
    B24:GOSUB25:GOSUB26
*36 PRINT"{DOWN}LOOK AT A SET OF":PRINT"TWO {BLK}BL
    ACK{BLU} KEYS:":PRINT"{DOWN}THE NAMES OF THE KE
    YS ARE C, D, AND E."
 37 PRINT"{BLK}{DOWN}":FORI=1TO7:PRINT"{3 SPACES}
    ⁅H⁆{3 SPACES}{RVS}{2 SPACES}{OFF}{3 SPACES}
    {RVS}{2 SPACES}{OFF}{4 SPACES}⁅H⁆":NEXT
 38 PRINT"{3 SPACES}⁅H⁆{4 SPACES}⁅H⁆{4 SPACES}
    ⁅H⁆{4 SPACES}⁅H⁆{6 SPACES}⁅H⁆ {RED}C{BLK}
    {2 SPACES}⁅H⁆ {RED}D{BLK}{2 SPACES}⁅H⁆
    {RED}E{BLK}{2 SPACES}⁅H⁆":PRINT"{3 SPACES}
    ⁅H⁆{4 SPACES}⁅H⁆{4 SPACES}⁅H⁆{4 SPACES}
    ⁅H⁆"
 39 PRINT"⁅22 Y⁆":GOSUB26
*4Ø PRINT"THE LETTER NAMES OF{3 SPACES}THE KEYS ARE
     THE{6 SPACES}ALPHABET LETTERS{6 SPACES}UP TO G
    .{3 DOWN}{BLK}"
 41 GOSUB24:PRINT"{2 SPACES}B{SHIFT-SPACE}B
    {SHIFT-SPACE}B{SHIFT-SPACE}B{SHIFT-SPACE}B
    {SHIFT-SPACE}B{SHIFT-SPACE}B{SHIFT-SPACE}B
    {SHIFT-SPACE}B{SHIFT-SPACE}B{2 SPACES}{RED}F
    {BLK}B{RED}G{BLK}B{RED}A{BLK}B{RED}B{BLK}B{PUR}
    C{BLK}B{PUR}D{BLK}B{PUR}E{BLK}B{RED}F{BLK}B
    {RED}G{BLK}B{RED}A{BLK}B{RED}B{BLK}";
 42 PRINT"{2 SPACES}B{SHIFT-SPACE}B{SHIFT-SPACE}B
    {SHIFT-SPACE}B{SHIFT-SPACE}B{SHIFT-SPACE}B
    {SHIFT-SPACE}B{SHIFT-SPACE}B{SHIFT-SPACE}B
    {SHIFT-SPACE}B{SHIFT-SPACE}⁅22 Y⁆":GOSUB26
 5Ø FORI=1TO1Ø
 51 PRINT"{CLR}{BLK}":K=INT(RND(1)*2)+1:ONKGOSUB2Ø,
    22:PRINT"⁅22 T⁆"
 52 N=INT(RND(1)*7)+1:ONKGOTO53,54
 53 ONNGOSUB3,4,5,6,8,1Ø,12:GOTO55
 54 ONNGOSUB6,8,1Ø,12,14,15,16
 55 POKE7944+CC,42:CS=38664+CC:FORD=1TO2Ø:POKECS,7:
    POKECS,2:NEXT
```

```
56 PRINT"{BLU}{3 DOWN}NAME THE NOTE"
57 GETA$:IFA$=""THEN57
58 IFA$=AA$THEN6Ø
59 POKE36877,128:FORD=1TO4ØØ:NEXT:POKE36877,Ø:GOTO
   57
6Ø POKE7944+CC,ASC(A$)-64:POKESS,S:FORD=1TO4ØØ:NEX
   TD:POKESS,Ø:NEXTI
61 PRINT"{3 DOWN}{GRN}TRY AGAIN (Y/N)"
62 GETA$:IFA$="Y"THEN5Ø
63 IFA$<>"N"THEN62
64 PRINT"{CLR}{BLU}":END
```

The Color Computer version of the program works somewhat differently:

Line	Function
0	Clear string space and branch past subroutines.
1–4	Wait for user to press ENTER.
5–15	Define sound S and letter name H$.
16	PRINT title.
17	Define string variables to PRINT keyboard.
18–33	Present instructions to learn the letter names of the keys.
34	Title for quiz.
35–36	Draw keyboard.
37–45	Present quiz. Choose a key randomly and blink a character under the key. If the answer is correct, the correct note plays. The key must be named correctly to continue.
46–48	PRINT option to try again and branch appropriately.
49	Clear screen and END.

Program 5-2. Musical Keyboard

TRS-80 Color Computer; MC-10

```
Ø CLEAR3ØØ:GOTO16
1 PRINT@496,"PRESS <ENTER>";
2 G$=INKEY$:IFG$=""THEN2
3 IFASC(G$)<>13THEN2
4 CLS:RETURN
5 S=5:H$="F":RETURN
6 S=32:H$="G":RETURN
7 S=58:H$="A":RETURN
8 S=78:H$="B":RETURN
9 S=89:H$="C":RETURN
1Ø S=1Ø8:H$="D":RETURN
11 S=125:H$="E":RETURN
12 S=133:H$="F":RETURN
```

```
13 S=147:H$="G":RETURN
14 S=159:H$="A":RETURN
15 S=17Ø:H$="B":RETURN
16 CLS:PRINT "  ** THE MUSICAL KEYBOARD **"
17 A$=CHR$(159)+CHR$(154)+CHR$(144):B$=CHR$(159)+C
   HR$(159)+CHR$(149):L$=CHR$(159)+A$+A$+A$+B$+A$+
   A$+B$+A$+A$+A$+CHR$(159)
18 PRINT@129,"EACH KEY ON THE KEYBOARD":PRINT" HAS
    A SPECIFIC LETTER NAME.":PRINT@225,"THE LETTER
   S WE USE ARE":PRINT" A B C D E F G"19 PRINT " T
   HEN THE LETTERS START OVER--":PRINT " ABCDEFGAB
   CDEFG ...":GOSUB1
2Ø FORI=ØTO192STEP32:PRINT@I,L$:NEXT:C$=CHR$(159)+
   CHR$(159)+CHR$(149):M$=""
21 FORI=1TO1Ø:M$=M$+C$:NEXT:M$=CHR$(159)+M$+CHR$(1
   59)
22 PRINT@224,M$:PRINT@256,M$:PRINT@289,"F  G  A  B
     C  D  E  F  G  A  B":PRINT@353,"THE KEYBOARD
   HAS GROUPS OF":PRINT" 3 BLACK KEYS ALTERNATING"
23 PRINT" WITH GROUPS OF 2 BLACK KEYS.":GOSUB 1:PR
   INT" ONE OF THE EASIEST KEYS":PRINT" TO FIND IS
    'C'."
24 PRINT@97,"FIND A GROUP OF TWO BLACK KEYS."
25 FORI=17ØTO266STEP32:PRINT@I,A$+A$+CHR$(159)+CHR
   $(159):NEXT
26 FORI=298TO33ØSTEP32:PRINT@I,B$+B$+CHR$(159)+CHR
   $(159):NEXT
27 PRINT@385,"JUST TO THE LEFT OF THE":PRINT@417,"
   TWO BLACK KEYS IS 'C'.":PRINT@33Ø,"C";:GOSUB1
28 PRINT @33,"'D' IS BETWEEN THE":PRINT " TWO BLAC
   K KEYS.":PRINT " 'E' IS ON THE RIGHT."
29 FORI=2Ø2TO298STEP32:PRINT@I,A$+A$+CHR$(159)+CHR
   $(159):NEXT
3Ø FORI=33ØTO362STEP32:PRINT@I,B$+B$+CHR$(159)+CHR
   $(159):NEXT
31 PRINT@394,"C  D  E":GOSUB1:PRINT@33,"THE KEYS B
   Y THE GROUP OF":PRINT" 3 BLACK KEYS ARE:"
32 FORI=2Ø2TO298STEP32:PRINT@I,A$+A$+A$+CHR$(159)+
   CHR$(159):NEXT
33 FORI=33ØTO362STEP32:PRINT@I,B$+B$+B$+CHR$(159)+
   CHR$(159):NEXT:PRINT@394,"F  G  A  B":GOSUB1
34 PRINT @228,"NOW FOR A QUIZ . . .":GOSUB1
35 CLS:FORI=ØTO192STEP32:PRINT@I,L$:NEXT
36 PRINT@224,M$:PRINT@256,M$:PRINT@394,"NAME THE K
   EY"
37 FOR I=1 TO 1Ø
38 N=RND(11):IF N=N1 THEN 38
39 P=286+3*N:N1=N:PRINT @P,CHR$(198)
4Ø G$=INKEY$:IF G$="" THEN 4Ø
```

```
41 ON N GOSUB 5,6,7,8,9,1Ø,11,12,13,14,15
42 IFG$=H$THEN44
43 SOUND125,1:SOUND89,1:GOTO4Ø
44 PRINT@P,G$:SOUNDS,15:PRINT@P," "
45 NEXT
46 PRINT@394," GOOD !!{4 SPACES}":PRINT@448,"TRY A
   GAIN? (Y/N)"
47 E$=INKEY$:IF E$="Y" THEN 35
48 IF E$<>"N" THEN 47
49 CLS:END
```

Here's how the TI version of Musical Keyboard works:

Line	Function
110–140	PRINT title screen.
150–230	Define characters, strings, and colors for graphics.
240–270	READ note name ASCII code and frequency for each key.
280–420	Present instructions.
430–700	Present quiz. Choose a key randomly and blink a question mark on the key. If the answer key pressed is correct, the correct note plays. R chooses one of two keyboards displayed.
710–740	PRINT option to try again and branch appropriately.
750–860	Draw musical keyboard.
870–900	Wait for ENTER key to be pressed.
910–920	Clear screen and END.

Program 5-3. Musical Keyboard

TI-99/4A

```
11Ø DIM N(13,2),L$(2),M$(2)
12Ø CALL CLEAR
13Ø CALL SCREEN(8)
14Ø PRINT "  ** MUSICAL KEYBOARD **"::::::::::::
15Ø CALL CHAR(97,"Ø")
16Ø CALL CHAR(98,"FFFFFFFFFFFFFFFF")
17Ø CALL CHAR(99,"Ø1Ø1Ø1Ø1Ø1Ø1Ø1Ø1")
18Ø CALL CHAR(1ØØ,"8Ø8Ø8Ø8Ø8Ø8Ø8Ø8Ø")
19Ø L$(1)="dabbabbacdabbabbabbacdabbabb"
2ØØ L$(2)="bbabbabbacdabbabbacdabbabbab"
21Ø M$(1)="dacdacdacdacdacdacdacdacdacd"
22Ø M$(2)="cdacdacdacdacdacdacdacdacdac"
23Ø CALL COLOR(9,2,16)
24Ø FOR C=1 TO 13
25Ø READ N(C,1),N(C,2)
26Ø NEXT C
```

```
270 DATA 67,262,68,294,69,330,70,349,71,392,65,440
    ,66,494,67,523,68,587,69,659,70,698,71,784,65,
    880
280 R=1
290 GOSUB 750
300 PRINT ::"THE LETTER NAMES OF THE KEYS"::"ARE T
    HE FIRST 7 LETTERS"::"OF THE ALPHABET."
310 DATA 67,68,69,70,71,65,66,67,68
320 FOR C=4 TO 28 STEP 3
330 READ G
340 CALL HCHAR(15,C,G)
350 NEXT C
360 PRINT :"THE KEYBOARD HAS GROUPS OF"::"TWO AND
    THREE BLACK KEYS."
370 GOSUB 870
380 GOSUB 750
390 PRINT :"YOU MAY REMEMBER THAT JUST"::"LEFT OF
    THE TWO BLACK KEYS"::"IS THE KEY CALLED 'C'.":
    ::
400 CALL HCHAR(14,4,67)
410 CALL HCHAR(14,25,67)
420 GOSUB 870
430 SC=0
440 FOR T=1 TO 10
450 F=0
460 RANDOMIZE
470 R=INT(2*RND+1)
480 X=INT(9*RND+1)
490 J=3*X+R
500 GOSUB 750
510 PRINT :::TAB(8);"NAME THE KEY"::::
520 IF R=1 THEN 540
530 X=X+4
540 CALL HCHAR(14,J,63)
550 CALL KEY(0,K,S)
560 CALL COLOR(4,16,16)
570 CALL COLOR(4,7,16)
580 IF S<1 THEN 550
590 IF K=N(X,1)THEN 630
600 CALL SOUND(400,-8,2)
610 F=1
620 GOTO 550
630 CALL HCHAR(14,J,K)
640 CALL SOUND(600,N(X,2),2)
650 CALL SOUND(1,N(X,2),30)
660 IF F=1 THEN 680
670 SC=SC+1
680 NEXT T
690 CALL COLOR(4,2,1)
```

```
700 PRINT "SCORE";SC;"OUT OF 10"
710 PRINT ::"TRY AGAIN? (Y/N)"
720 CALL KEY(0,K,S)
730 IF K=89 THEN 430
740 IF K=78 THEN 910 ELSE 720
750 CALL CLEAR
760 FOR C=1 TO 12
770 PRINT L$(R)
780 IF R=2 THEN 800
790 CALL HCHAR(23,31,97)
800 NEXT C
810 FOR C=1 TO 3
820 PRINT M$(R)
830 IF R=2 THEN 850
840 CALL HCHAR(23,31,97)
850 NEXT C
860 RETURN
870 PRINT :"PRESS <ENTER>.";
880 CALL KEY(0,K,S)
890 IF K<>13 THEN 880
900 RETURN
910 CALL CLEAR
920 END
```

Treble Clef Notes

Computers: VIC-20; TRS-80 Color Computer; MC-10; TI-99/4A

"Treble Clef Notes" can help you learn to read the notes of the treble clef. You can more easily remember the notes by thinking of the word F-A-C-E for the letter names of the notes, starting with the bottom space. You can remember the phrase Every Good Boy Does Fine, which has been a standard phrase in teaching the treble clef line notes. The first letter of each word in the phrase represents the letter name of the note, starting with the bottom line of the treble staff.

After the names of the letters are introduced, there is a quiz of ten notes. A note is chosen randomly, and you press the key with the letter name of the note. If the answer is correct, the right note will play. You must be correct to continue the game. After ten problems, you have the option to try again or end the program.

Treble Clef works like this on the VIC:

Line	Function
2	Dimension N$ for the name of the note and the tone, turn volume on, and define S for sound.
3	READ values for N$ from data.

8–12	PRINT title screen and play scale.
14–24	PRINT menu screen of options and then branch accordingly.
30–35	PRINT instructions to learn names of the space notes.
37	Quiz for naming ten notes chosen randomly.
40–46	Instructions to learn names of the line notes.
48	Quiz for naming ten notes chosen randomly.
50–63	Draw treble staff with note A chosen randomly.
64–66	Clear the note.
70–86	Perform the quiz.
90	DATA for note names and tones.
95–98	Wait for RETURN to be pressed.
100	Perform quiz including all notes.
200	Clear screen and END.

Program 5-4. Treble Clef

VIC-20

(In this program, make sure to use abbreviations for all BASIC keywords in the lines marked with an asterisk (*). If you don't, the program will not run correctly. Refer to Appendix C, "BASIC Keyword Abbreviations," for the proper abbreviations.)

```
 2 DIMN$(8,1):POKE36878,15:S=36876
 3 FORI=ØTO8:READN$(I,Ø),N$(I,1):NEXT
 8 PRINT"{CLR}{BLU}{2 DOWN}{2 SPACES}LET'S LEARN NO
   TES{2 DOWN}":PRINTTAB(5);"TREBLE CLEF"
 12 FORI=ØTO8:POKES,VAL(N$(I,1)):FORJ=1TO2ØØ:NEXTJ,
    I:POKES,Ø
*14 PRINT"{CLR}{BLU}{2 DOWN}{5 SPACES}TREBLE CLEF
    {2 DOWN}":PRINT"CHOOSE:":PRINT"{DOWN} 1 NOTES O
    N SPACES":PRINT"{DOWN} 2 NOTES ON LINES"
 16 PRINT"{DOWN} 3 ALL TREBLE NOTES":PRINT"{DOWN} 4
     END PROGRAM":GETA$
 18 GETA$:IFA$=""THEN18
 2Ø IFASC(A$)<49ORASC(A$)>52THEN18
 22 ONVAL(A$)GOSUB3Ø,4Ø,1ØØ,2ØØ
 24 GOTO14
 3Ø PRINT"{CLR}{DOWN}THERE ARE FOUR SPACES ON THE S
    TAFF."
 31 PRINT"THE NAMES OF THE NOTESON THE SPACES ARE:"
    :PRINT"{2 DOWN}{BLK}[21 Y]"
*32 PRINTTAB(11);"{RED}E{BLK}{DOWN}":PRINT"[21 Y]
    ":PRINTTAB(9);"{RED}C{BLK}{DOWN}":PRINT"
    [21 Y]"
*33 PRINTTAB(7);"{RED}A{BLK}{DOWN}":PRINT"[21 Y]"
    :PRINTTAB(5);"{RED}F{BLK}{DOWN}":PRINT"[21 Y]
    "
 35 PRINT"{BLU}REMEMBER THE WORD {RED}FACE":GOSUB95
    :Z=1
 37 FORI=1TO10:A=(INT(RND(1)*4)+1)*2-1:GOSUB70:NEXT
    :Z=0:RETURN
 40 PRINT"{CLR}THERE ARE FIVE LINES{2 SPACES}ON THE
     TREBLE STAFF.{DOWN}"
 41 PRINT"THE NAMES OF THE NOTESON THE LINES ARE:
    {BLK}":PRINT"{2 DOWN}CCCCCCCCCCCCCCCCCCC{RED}F
    {BLK}CC"
 42 PRINT"{DOWN}CCCCCCCCCCCCCCC{RED}D{BLK}CCCCCC
    {2 DOWN}CCCCCCCCCCC{RED}B{BLK}CCCCCCCCCC
    {2 DOWN}CCCCCCC{RED}G{BLK}CCCCCCCCCCCCCC"
*43 PRINT"{DOWN}CCC{RED}E{BLK}CCCCCCCCCCCCCCCCCC":G
    OSUB95:PRINT"TO HELP YOU REMEMBER{2 SPACES}THE
    {SPACE}NOTES, THINK OF"
 44 PRINT"THIS PHRASE:":PRINT"{3 DOWN}{BLK}CCCCCCCC
    CCCCCCCCC{RED}F{BLU}INE{BLK}C{2 DOWN}CCCCCCCCCC
    CCC{RED}D{BLU}OES{BLK}CCCCC"
*45 PRINT"{DOWN}CCCCCCCCCC{RED}B{BLU}OY{BLK}CCCCCCC
    CC{2 DOWN}CCCCCC{RED}G{BLU}OOD{BLK}CCCCCCCCCCCC
    {2 DOWN}C{RED}E{BLU}VERY{BLK}CCCCCCCCCCCCCCCCC"
 46 GOSUB95
 48 FORI=1TO10:A=(INT(RND(1)*5))*2:GOSUB70:NEXT:RET
    URN
 50 PRINT"{CLR}{BLK}{2 DOWN}{3 SPACES}UI{DOWN}
    {2 LEFT}GG":PRINT"⁅3 Y⁆TT⁅17 Y⁆{2 SPACES}
    ⁅M⁆N"
 52 PRINT"⁅2 Y⁆NO⁅18 Y⁆ ⁅M⁆ ⁅G⁆"
 55 PRINT"⁅Y⁆PUOM⁅17 Y⁆ ⁅M⁆ G@{2 SPACES}":PRI
    NT"⁅Y⁆P⁅Y⁆GP⁅17 Y⁆{2 SPACES}MBN"
 60 PRINT"⁅3 Y⁆B⁅18 Y⁆{3 SPACES}B{DOWN}{2 LEFT}
    JK":PRINT"{4 DOWN}{BLU}NAME THE NOTE";
 61 IFZ=1THENPRINT"{YEL}{2 SPACES}B{UP}{LEFT}B{UP}
    {LEFT}UCCI{DOWN}{3 LEFT}WWH{DOWN}{LEFT}H{DOWN}
    {4 LEFT}BJKH{DOWN}{4 LEFT}JFFK"
*62 B=22*A:POKE7935-B,85:POKE7936-B,67:POKE7937-B,7
    3:POKE7957-B,74:POKE7958-B,70:POKE7959-B,75
 63 FORII=38655-BTO38657-B:POKEII,2:POKEII+22,2:NEX
    T:RETURN
*64 POKE7935-B,32:POKE7936-B,32:POKE7937-B,32:POKE7
    957-B,119:POKE7958-B,119:POKE7959-B,119:RETURN
*66 POKE7935-B,119:POKE7936-B,119:POKE7937-B,119:PO
    KE7957-B,32:POKE7958-B,32:POKE7959-B,32:RETURN
 70 GOSUB50
 72 GETA$:IFA$=""THEN72
 74 IFA$=N$(A,0)THEN78
 76 POKE36877,128:FORD=1TO60:NEXT:POKE36877,0:GOTO7
    2
 78 POKES,VAL(N$(A,1)):POKE7962-B,ASC(N$(A,0))-64:P
    OKE38682-B,0
 80 FORD=1TO300:NEXT:ONA+1GOSUB64,66,64,66,64,66,64
    ,66,64
```

```
82 IFA/2=INT(A/2)THENPOKE7962-B,119:GOTO86
84 POKE7962-B,32
86 POKES,Ø:RETURN
9Ø DATAE,158,F,161,G,173,A,181,B,189,C,192,D,2ØØ,E
   ,2Ø6,F,2Ø8
95 PRINT"{GRN}PRESS RETURN";
96 GETA$:IFA$=""THEN96
97 IFASC(A$)<>13THEN96
98 PRINT"{CLR}{BLU}":RETURN
1ØØ FORI=1TO1Ø:A=INT(RND(1)*9):GOSUB7Ø:NEXT:RETURN
2ØØ PRINT"{CLR}{BLU}":END
```

On the Color Computer, Treble Clef works in this way:

Line	Function
1	GOTO line 22, where the PRINT statements for the title screen and instructions begin.
2–5	Routine for pressing the ENTER key to continue reading the instruction screens.
6–15	Set tones and assign letters to H$ for the ten possible notes.
16–21	Clear the screen and draw the treble staff on the screen.
22–28	PRINT the title screen and instructions. At the end of each screen of instructions, the player must press the ENTER key to continue.
29–30	After drawing the treble staff, place notes in the correct positions, as well as PRINT the mnemonics such as F-A-C-E and Every Good Boy Does Fine.
31–35	Perform the quiz, randomly selecting one of the ten possible notes. As a key is pressed, the note plays and its letter is displayed. If the answer is correct, the reward message shows and the player is asked if another note is wanted.
36	Wrong answer sound and return to line 33 for another key to be pressed.
37–38	Reward sound and message. Player is asked if another round is desired.
39–40	Begin the game again if Y was pressed or end the program if N was pressed.

Program 5–5. Treble Clef

TRS-80 Color Computer, MC-10

```
1 GOTO22
2 PRINT@464,"PRESS <ENTER>";
3 E$=INKEY$:IFE$=""THEN3
```

```
4 IFASC(E$)<>13THEN3
5 CLS:RETURN
6 S=1Ø8:H$="D":RETURN
7 S=125:H$="E":RETURN
8 S=133:H$="F":RETURN
9 S=147:H$="G":RETURN
1Ø S=159:H$="A":RETURN
11 S=17Ø:H$="B":RETURN
12 S=176:H$="C":RETURN
13 S=185:H$="D":RETURN
14 S=193:H$="E":RETURN
15 S=197:H$="F":RETURN
16 CLS:PRINTI$+D$+CHR$(156)+CHR$(156)+F$+B$:PRINT@
   32,I$+CHR$(154)+F$+CHR$(149)+D$+B$:PRINT@64,K$+
   K$+CHR$(146)+J$+CHR$(145)+C$+A$
17 PRINT@96,I$+CHR$(154)+CHR$(158)+CHR$(153)+F$+B$
   :PRINT@128,K$+J$+CHR$(144)+C$+CHR$(145)+K$+A$:P
   RINT@16Ø,G$+CHR$(158)+CHR$(153)+F$+CHR$(149)+G$
   +B$
18 PRINT@192,K$+CHR$(145)+C$+CHR$(145)+C$+CHR$(145
   )+C$+CHR$(146)+C$+A$:PRINT@224,F$+CHR$(154)+F$+
   CHR$(151)+D$+CHR$(149)+F$+CHR$(149)+B$
19 PRINT@256,K$+CHR$(145)+K$+CHR$(146)+J$+CHR$(145
   )+A$:PRINT@288,G$+CHR$(155)+CHR$(15Ø)+CHR$(157)
   +D$+CHR$(154)+CHR$(158)+CHR$(156)+CHR$(151)+B$
2Ø PRINT@32Ø,K$+J$+J$+CHR$(146)+K$+A$:PRINT@352,F$
   +F$+CHR$(158)+F$+CHR$(154)+G$+B$
21 PRINT@384,G$+F$+CHR$(15Ø)+CHR$(156)+CHR$(149)+G
   $+B$:PRINT@416,I$+G$+F$+B$:RETURN
22 CLS:PRINT"{4 SPACES}** TREBLE CLEF NOTES **":A$
   ="":B$="":C$=CHR$(147):D$=CHR$(159):FORI=1TO21:
   A$=A$+C$:B$=B$+D$:NEXT
23 F$=D$+D$:G$=F$+D$:I$=G$+G$:J$=C$+C$:K$=J$+C$
24 PRINT@97,"EACH NOTE DRAWN ON THE":PRINT" TREBLE
    STAFF REPRESENTS A":PRINT" CERTAIN NOTE ON THE
    KEYBOARD."
25 PRINT@225,"IF YOU LEARN THE LETTER NAMES
   {3 SPACES}OF THE NOTES YOU CAN FIND THE
   {3 SPACES}CORRECT NOTE ON THE KEYBOARD.":GOSUB2
26 PRINT@33,"THERE ARE TWO PHRASES TO":PRINT" HELP
    LEARN THE TREBLE NOTES.":PRINT@129,"THE WORD '
   F A C E' REPRESENTS"
27 PRINT" THE NOTES ON THE SPACES.":PRINT@257,"THE
    LINE NOTES ARE THE FIRST":PRINT" LETTERS OF EA
   CH WORD IN"
28 PRINT" THIS PHRASE:":PRINT@353,"EVERY GOOD BOY
   DOES FINE.":PRINT" -{5 SPACES}-{4 SPACES}-
   {3 SPACES}-{4 SPACES}-":GOSUB2
```

```
29 GOSUB16:PRINT@400,"F   A   C   E";:PRINT@304,"F";:
   PRINT@243,"A";:PRINT@182,"C";:PRINT@121,"E";:GO
   SUB2
30 GOSUB16:PRINT@331,"EVERY";:PRINT@272,"GOOD";:PR
   INT@212,"BOY";:PRINT@151,"DOES";:PRINT@90,"FINE
   ";:GOSUB2
31 FORI=1TO10:GOSUB16:PRINT@458,"NAME THE NOTE":N=
   RND(10):Z=N*2
32 FORX=41TO46:FORY=Z TOZ+3:SET(X,Y,4):NEXTY,X
33 E$=INKEY$:IFE$=""THEN33
34 ONN GOSUB15,14,13,12,11,10,9,8,7,6
35 IFE$=H$THEN37
36 SOUND125,1:SOUND89,1:GOTO33
37 PRINT@N*32+54,E$;:SOUNDS,15:NEXT
38 CLS:PRINT@234,"GOOD WORK!":PRINT@320,"TRY AGAIN
   ? (Y/N)"
39 E$=INKEY$:IFE$="Y"THEN31
40 IFE$<>"N"THEN39
41 CLS:END
```

The TI version of Treble Clef operates like this:

Line	Function
110–130	PRINT title screen.
140–170	READ note name and frequency for each of the notes on the clef.
180–280	Define characters for graphics. L$ will print a line.
290–440	Present instructions to learn the names of the notes.
450–750	Perform the quiz of ten notes chosen randomly. Note is played when correct key is pressed. The answer must be correct to continue.
760–810	PRINT option to try again and branch appropriately.
820–860	Draw the treble staff.
870–910	Wait for ENTER key to be pressed.
920–930	Clear screen and END.

Program 5-6. Treble Clef

TI-99/4A

```
110 DIM N(9,2)
120 CALL CLEAR
130 PRINT TAB(6);"** TREBLE CLEF **":::::::::::
140 FOR C=1 TO 9
150 READ N(C,1),N(C,2)
160 NEXT C
170 DATA 70,698,69,659,68,587,67,523,66,494,65,440
    ,71,392,70,349,69,330
180 FOR C=96 TO 131
```

```
19Ø READ C$
2ØØ CALL CHAR(C,C$)
21Ø NEXT C
22Ø DATA ØØØØØØFF,ØØØ384482828282,828282828282828
    2,848488FF889Ø9ØA,ØØØØØØØØØØØ1Ø1Ø2,AØAØCØCØCØ4
    Ø4Ø4
23Ø DATA Ø4Ø81ØFF2Ø4Ø8Ø8,4Ø4Ø4ØFF2Ø2Ø2Ø2,Ø1Ø2Ø4Ø4Ø
    81Ø1Ø1,2Ø2Ø2Ø2Ø2Ø2Ø2Ø2,2Ø2Ø4ØFF4Ø4Ø4Ø4,ØØØØØØF
    FØ3Ø4Ø81
24Ø DATA 1Ø1Ø1ØFF1Ø1Ø1Ø1,ØØØØØØFF18Ø4Ø2Ø1,8Ø8Ø8Ø8Ø
    8Ø8Ø8Ø8,1Ø2Ø2Ø2Ø2Ø4Ø4Ø4,Ø8Ø8Ø8Ø8Ø8Ø8Ø8Ø8,CØ2Ø2
    Ø1Ø1ØØ8Ø8Ø8
25Ø DATA 8Ø8Ø8ØFF4Ø4Ø4Ø2,4Ø4Ø4ØFF2Ø2Ø1ØØC,Ø4Ø4Ø4FF
    Ø4Ø4Ø4Ø4,2Ø2Ø1Ø1ØØ8Ø4Ø4Ø3,Ø2Ø2Ø2Ø2Ø2Ø1Ø1Ø1,Ø8Ø
    81Ø1Ø2Ø4Ø8
26Ø DATA 8Ø4Ø38FF,Ø1Ø1Ø1FFØ1Ø1Ø1Ø1,Ø3ØC3ØFF,Ø1Ø1Ø1
    Ø1Ø1Ø1Ø1Ø1,Ø1Ø1Ø1Ø111111ØE,ØØØØØØØØØØØØF3ØC,ØØØØ
    ØØØØØØFØØCØ3
27Ø DATA Ø1Ø2Ø2Ø4Ø4Ø4Ø2Ø1,7Ø88Ø83Ø4Ø4ØØØ4,8Ø4Ø4Ø2Ø
    2Ø2Ø4Ø8,8Ø6Ø1ØØF,Ø1Ø6Ø8F
28Ø L$="````````````````````````````"
29Ø GOSUB 82Ø
3ØØ PRINT :"THINK OF THE WORD 'FACE' FOR"::"THE NO
    TE NAMES ON SPACES."::
31Ø CALL HCHAR(14,14,7Ø)
32Ø CALL HCHAR(12,17,65)
33Ø CALL HCHAR(1Ø,2Ø,67)
34Ø CALL HCHAR(8,23,69)
35Ø GOSUB 87Ø
36Ø GOSUB 82Ø
37Ø PRINT :"MEMORIZE THIS PHRASE TO HELP"::"LEARN
    LINE NOTES E G B D F."::
38Ø DATA 15,69,15,86,15,69,15,82,15,89,13,71,13,79
    ,13,79,13,68,11,66,11,79,11,89,9,68,9,79
39Ø DATA 9,69,9,83,7,7Ø,7,73,7,78,7,69
4ØØ FOR I=9 TO 28
41Ø READ C,G
42Ø CALL HCHAR(C,I,G)
43Ø NEXT I
44Ø GOSUB 87Ø
45Ø GOSUB 82Ø
46Ø PRINT TAB(8);"NAME THE NOTE"::::
47Ø FOR T=1 TO 1Ø
48Ø RANDOMIZE
49Ø X=INT(9*RND+1)
5ØØ J=5+X
51Ø CALL HCHAR(J,2Ø,125)
52Ø CALL HCHAR(J,21,126)
53Ø CALL HCHAR(J+1,19,127)
```

```
540 CALL HCHAR(J+1,22,129)
550 CALL HCHAR(J+2,20,130)
560 CALL HCHAR(J+2,21,131)
570 CALL HCHAR(J+1,21,128)
580 CALL SOUND(150,1397,4)
590 CALL KEY(0,K,S)
600 IF S<1 THEN 590
610 IF K=N(X,1)THEN 640
620 CALL SOUND(200,-5,4)
630 GOTO 590
640 CALL HCHAR(J+1,21,N(X,1))
650 CALL SOUND(500,N(X,2),2)
660 CALL SOUND(1,N(X,2),30)
670 IF X/2=INT(X/2)THEN 720
680 CALL HCHAR(J,20,32,2)
690 CALL HCHAR(J+1,19,96,4)
700 CALL HCHAR(J+2,20,32,2)
710 GOTO 750
720 CALL HCHAR(J,20,96,2)
730 CALL HCHAR(J+1,19,32,4)
740 CALL HCHAR(J+2,20,96,2)
750 NEXT T
760 PRINT "GOOD!  TRY AGAIN? (Y/N)";
770 CALL KEY(0,K,S)
780 IF K=78 THEN 920
790 IF K<>89 THEN 770
800 CALL HCHAR(24,13,32,15)
810 GOTO 450
820 CALL CLEAR
830 PRINT TAB(9);"TREBLE CLEF":::
840 PRINT "   a":"   b":"```c``"&L$:"   de":"``fg``
    "&L$;" h i":"`jklm`"&L$
850 PRINT " nop q":"`rst`t"&L$:" u v w":"``xyz`"&L
    $:"   {":"   {":"   !"
860 RETURN
870 PRINT :TAB(16);"PRESS <ENTER>";
880 CALL KEY(0,K,S)
890 IF K<>13 THEN 880
900 CALL HCHAR(24,18,32,13)
910 RETURN
920 CALL CLEAR
930 END
```

In this program, you'll see spaces between quotation marks, such as "   ". Type in the quote marks and press the space bar the correct number of times. If it's hard to tell how many spaces to enter, you can use the line above as a reference. For example, in line 850, you can tell that three spaces are needed near the line's end by counting the characters in line 840 directly above the gap.

Bass Clef Notes

Computers: VIC-20; TRS-80 Color Computer; MC-10; TI-99/4A

Similar to Treble Clef, this program introduces you to bass clef musical notes. The letter names can be remembered by two animal phrases — All Cows Eat Grass for the space notes, and Great Big Dogs Fight Animals for the line notes. The first letter of each word in the phrase is the letter name of the musical note. Both letter strings start with the notes at the bottom and move upward.

A note is randomly drawn on the bass staff, and you must press the key with the letter name of that note. If the answer is correct, the correct note or a short musical tune will play. After a quiz of ten notes, you can try again or end the program.

The program explanation for each computer's version is almost identical to those for Treble Clef Notes, except that this program uses bass clef notes. Refer to the program descriptions for Treble Clef if you want to see how each program works.

Program 5-7. Bass Clef

VIC-20

(In this program, make sure to use abbreviations for all BASIC keywords in the lines marked with an asterisk (*). If you don't, the program will not run correctly. Refer to Appendix C, "BASIC Keyword Abbreviations," for the proper abbreviations.)

```
 2 DIMN$(8,1):POKE36878,15:S=36874:FORI=ØTO8:READN$
   (I,Ø),N$(I,1):NEXT
 3 Y$="{20 Y}"
 6 PRINT"{CLR}{BLU}{2 DOWN}{2 SPACES}LET'S LEARN NO
   TES{2 DOWN}":PRINTTAB(6);"BASS CLEF"
 8 FORI=ØTO8:POKES,VAL(N$(I,1)):FORJ=1TO2ØØ:NEXTJ,I
   :POKES,Ø
*1Ø PRINT"{CLR}{BLU}{2 DOWN}{6 SPACES}BASS CLEF
    {2 DOWN}":PRINT" CHOOSE:":PRINT"{DOWN} 1 NOTES
    {SPACE}ON SPACES":PRINT"{DOWN}{SHIFT-SPACE}2 NO
    TES ON LINES"
 12 PRINT"{DOWN} 3 ALL BASS NOTES":PRINT"{DOWN} 4 E
    ND PROGRAM":GETA$
 14 GETA$:IFA$=""THEN14
 15 IFASC(A$)<49ORASC(A$)>52THEN14
 16 ONVAL(A$)GOSUB3Ø,7Ø,8Ø,99
 18 GOTO1Ø
*3Ø PRINT"{CLR}{DOWN}THERE ARE FOUR SPACES ON THE S
    TAFF.{DOWN}":PRINT"THE NAMES OF THE NOTESON THE
     SPACES ARE:"
*32 PRINT"{2 DOWN}{BLK}";Y$:PRINTTAB(11);"{RED}G
    {BLK}{DOWN}":PRINTY$:PRINTTAB(9);"{RED}E{BLK}
    {DOWN}":PRINTY$:PRINTTAB(7);"{RED}C{BLK}{DOWN}"
    :PRINTY$:PRINTTAB(5);"{RED}A{BLK}{DOWN}":PRINTY
    $
 34 PRINT"{BLU}REMEMBER THE PHRASE{3 SPACES}{RED}A
    {BLK}LL {RED}C{BLK}OWS {RED}E{BLK}AT {RED}G
    {BLK}RASS":GOSUB95
*35 PRINT"{CLR}{BLK}{4 DOWN}":PRINTY$:PRINTTAB(14);
    "{RED}G{BLU}RASS{BLK}":PRINTY$:PRINTTAB(10);"
    {RED}E{BLU}AT{BLK}":PRINTY$:PRINTTAB(5);"{RED}C
    {BLU}OWS{BLK}":PRINTY$:PRINT" {RED}A{BLU}LL
    {BLK}"
 36 PRINTY$:PRINT"{5 DOWN}":GOSUB95
 38 FORI=1TO10:A=(INT(RND(1)*4)+1)*2-1:GOSUB40:NEXT
    :RETURN
 40 PRINT"{CLR}{BLK}{4 DOWN}⁅2 Y⁆N⁅Y⁆M⁅17 Y⁆
    {SPACE}⁅N⁆{3 SPACES}M Q":PRINT"⁅2 Y⁆K
    ⁅2 Y⁆P⁅Y⁆Q⁅14 Y⁆{5 SPACES}⁅M⁆"
 42 PRINT"⁅5 Y⁆N⁅16 Y⁆{4 SPACES}N":PRINT"
    ⁅3 Y⁆N⁅18 Y⁆":PRINTY$;"⁅2 Y⁆"
 44 PRINT"{4 DOWN}{BLU}NAME THE NOTE"
*46 B=22*A:POKE7935-B,85:POKE7936-B,67:POKE7937-B,7
    3:POKE7957-B,74:POKE7958-B,70:POKE7959-B,75
 48 FORII=38655-BTO38657-B:POKEII,2:POKEII+22,2 :NE
    XT
 50 GETA$:IFA$=""THEN50
 51 IFA$=N$(A,0)THEN54
 52 POKE36877,128:FORD=1TO60:NEXT:POKE36877,0:GOTO5
    0
 54 POKES,VAL(N$(A,1)):POKE7962-B,ASC(N$(A,0))-64:P
    OKE38682-B,0
 56 FORD=1TO300:NEXT:ONA+1GOSUB66,68,66,68,66,68,66
    ,68,66
 58 IFA/2=INT(A/2)THENPOKE7962-B,119:GOTO60
 59 POKE7962-B,32
 60 POKES,0:RETURN
*66 POKE7935-B,32:POKE7936-B,32:POKE7937-B,32:POKE7
    957-B,119:POKE7958-B,119:POKE7959-B,119:RETURN
*68 POKE7935-B,119:POKE7936-B,119:POKE7937-B,119:PO
    KE7957-B,32:POKE7958-B,32:POKE7959-B,32:RETURN
 70 PRINT"{CLR}{BLU}THERE ARE FIVE LINES{2 SPACES}O
    N THE BASS STAFF.{DOWN}"
 71 PRINT"THE NAMES OF THE NOTESON THE LINES ARE:":
    PRINT"{2 DOWN}{BLK}*******************{RED}A
    {BLK}**{DOWN}"
 72 PRINT"***************{RED}F{BLK}******{2 DOWN}
    ***********{RED}D{BLK}**********{2 DOWN}
    *******{RED}B{BLK}**************"
*73 PRINT"{DOWN}***{RED}G{BLK}******************{DO
    WN}":GOSUB95:PRINT"TO HELP YOU REMEMBER{2 SPACE
    S}THE NOTES, THINK OF"
 74 PRINT"THIS PHRASE:":PRINT"{3 DOWN}{BLK}********
    ******{RED}A{BLU}NIMALS {BLK}*{2 DOWN}**********
    {RED}F{BLU}IGHT{BLK}*******"
```

```
*75 PRINT"{DOWN}*******{RED}D{BLU}OGS{BLK}*********
    **{2 DOWN}*****{RED}B {BLU}IG{BLK}**************
    {2 DOWN}*{RED}G{BLU}REAT{BLK}****************"
 76 PRINT"{DOWN}":GOSUB95
 77 FORI=1TO1Ø:A=(INT(RND(1)*5))*2:GOSUB4Ø:NEXT:RET
    URN
 8Ø FORI=1TO1Ø:A=INT(RND(1)*9):GOSUB4Ø:NEXT:RETURN
 9Ø DATAG,175,A,183,B,191,C,195,D,2Ø1,E,2Ø7,F,2Ø9,G
    ,215,A,219
 95 PRINT"{GRN}PRESS RETURN";
 96 GETA$:IFA$=""THEN96
 97 IFASC(A$)<>13THEN96
 98 PRINT"{CLR}{BLU}":RETURN
 99 PRINT"{CLR}{BLU}":END
```

Program 5-8. Bass Clef

TRS-80 Color Computer; MC-10

```
1 GOTO18
2 PRINT@464,"PRESS <ENTER>";
3 G$=INKEY$:IF G$=""THEN 3
4 IFASC(G$)<>13THEN3
5 CLS:RETURN
6 H$="G":RETURN
7 H$="A":RETURN
8 H$="B":RETURN
9 H$="C":RETURN
1Ø H$="D":RETURN
11 H$="E":RETURN
12 H$="F":RETURN
13 CLS:PRINTB11$+B$:PRINT@32,B11$+B$:PRINT@64,B11$
   +B$:PRINT@96,A3$+A1$+CHR$(145)+A3$+CHR$(146)+CH
   R$(145)+A1$+A$
14 PRINT@128,B3$+CHR$(144)+B3$+B2$+CHR$(155)+CHR$(
   149)+BL$:PRINT@16Ø,A3$+A3$+A3$+A1$+CHR$(145)+A1
   $+A$
15 PRINT@192,B6$+B2$+B2$+CHR$(149)+BL$:PRINT@224,A
   3$+A3$+A3$+CHR$(146)+A3$+A$:PRINT@256,B6$+B2$+C
   HR$(158)+CHR$(151)+B1$+B$
16 PRINT@288,A3$+A3$+A1$+CHR$(146)+A3$+A$:PRINT@32
   Ø,B6$+CHR$(155)+CHR$(151)+B3$+B$
17 PRINT@352,A3$+A3$+A3$+A2$+A$:PRINT@384,B11$+B$:
   PRINT@416,B11$+B$:RETURN
18 CLS:PRINT"{5 SPACES}** BASS CLEF NOTES **":A$="
   ":B$="":A1$=CHR$(147):B1$=CHR$(159)
19 FOR I=1 TO 21:A$=A$+A1$:B$=B$+B1$:NEXT I
2Ø B2$=B1$+B1$:B3$=B2$+B1$:B6$=B3$+B3$:A2$=A1$+A1$
   :A3$=A2$+A1$
21 B11$=B6$+B3$+B2$:BL$=B2$+CHR$(151)+B6$+B6$+B6$
22 PRINT@97,"THE PHRASES TO HELP LEARN THE{3 SPACE
   S}BASS CLEF NOTES BOTH":PRINT" CONCERN ANIMALS."
```

```
23 PRINT@225,"REMEMBER, THE FIRST LETTER OF
   {3 SPACES}EACH WORD IS THE NAME OF NOTE."
24 PRINT@321,"FOR THE SPACES, THINK--":PRINT" ALL
   COWS EAT GRASS.":PRINT" -{3 SPACES}-{4 SPACES}-
   {3 SPACES}-":GOSUB 2
25 GOSUB 13
26 PRINT@392,"ALL COWS EAT GRASS";:PRINT@335,"ALL"
   ;:PRINT@274,"COWS";:PRINT@214,"EAT";:PRINT@153,
   "GRASS";:GOSUB2
27 PRINT @33,"THE PHRASE TO LEARN FOR THE
   {5 SPACES}NOTES ON THE LINES OF THE{7 SPACES}BA
   SS CLEF IS:"
28 PRINT @193,"GREAT BIG DOGS FIGHT ANIMALS.
   {3 SPACES}-{5 SPACES}-{3 SPACES}-{4 SPACES}-
   {5 SPACES}-"
29 PRINT @353,"REMEMBER TO START THE PHRASE
   {4 SPACES}ON THE BOTTOM LINE OF THE CLEF.":GOSU
   B 2
3Ø GOSUB13:PRINT@418,"GREAT BIG DOGS FIGHT ANIMALS
   ";
31 PRINT @366,"GREAT";:PRINT @3Ø5,"BIG";:PRINT @24
   3,"DOGS";:PRINT @182,"FIGHT";:PRINT @121,"ANIMA
   LS";:GOSUB 2
32 FOR I=1 TO 1Ø
33 GOSUB13:PRINT@458,"NAME THE NOTE"
34 N=RND(9):IFN=N1 THEN34
35 Y1=N*2+2:N1=N:FORX=41TO46:FORY=Y1 TOY1+3:SET(X,
   Y,4):NEXT Y,X
36 G$=INKEY$:IF G$="" THEN 36
37 ON N GOSUB 7,6,12,11,1Ø,9,8,7,6
38 IFG$=H$THEN4Ø
39 SOUND125,1:SOUND89,1:GOTO36
4Ø PRINT@(N+1)*32+54,G$;:SOUND88,2:SOUND125,2:SOUN
   D147,2:SOUND176,4
41 NEXT I
42 CLS:PRINT@17Ø,"GOOD WORK!":PRINT@32Ø,"TRY AGAIN
   ? (Y/N)"
43 E$=INKEY$:IF E$="Y" THEN 32
44 IF E$<>"N" THEN 43
45 CLS:END
```

Program 5-9. Bass Clef

TI-99/4A

```
11Ø DIM N(9,2)
12Ø CALL CLEAR
13Ø PRINT TAB(7);"** BASS CLEF **"::::::::::
14Ø FOR I=1 TO 9
15Ø READ N(I,1),N(I,2)
16Ø NEXT I
```

```
170 DATA 65,22Ø,71,196,7Ø,175,69,165,68,147,67,131
    ,66,123,65,11Ø,71,11Ø
18Ø FOR I=96 TO 114
19Ø READ C$
2ØØ CALL CHAR(I,C$)
21Ø NEXT I
22Ø DATA ØØØØØØFF,ØØØØØØFFØF1Ø6Ø8,ØØØØØØFFFF,ØØØØØ
    ØFF8Ø7ØØCØ3,Ø1Ø2Ø4181Ø2Ø2Ø4,8Ø4Ø2Ø2Ø1ØØ8Ø8Ø4,4
    Ø583CFF3C18
23Ø DATA Ø4Ø2Ø2FFØ1Ø1Ø1Ø1,Ø1Ø1Ø2Ø2Ø4Ø4Ø4Ø8,Ø8Ø81ØF
    F2Ø4Ø8Ø8,Ø1Ø1Ø2Ø4Ø8,ØØØE1F1F1FØE
24Ø DATA ØØØØØØØØØØØØF3ØC,ØØØØØØØØØØFØØCØ3,Ø1Ø2Ø2Ø4
    Ø4Ø4Ø2Ø1,7Ø88Ø83Ø4Ø4ØØØ4,8Ø4Ø4Ø2Ø2Ø2Ø4Ø8,8Ø6Ø1
    ØØF,Ø1Ø6Ø8F
25Ø L$="``````````````````````"
26Ø GOSUB 76Ø
27Ø PRINT :"LEARN THIS PHRASE FOR THE"::"NOTES ON
    SPACES, A C E G."::
28Ø GOSUB 8ØØ
29Ø DATA 13,3Ø,15,65,15,76,15,76,15,32,13,67,13,79
    ,13,87,13,83,13,32,11,69,11,65,11,84,11,32
3ØØ DATA 9,71,9,82,9,65,9,83,9,83
31Ø GOSUB 86Ø
32Ø GOSUB 76Ø
33Ø PRINT :"THIS PHRASE HELPS YOU KNOW"::"THE LINE
     NOTES, G B D F A."::
34Ø GOSUB 8ØØ
35Ø DATA 8,32,16,71,16,82,16,69,16,65,16,84,14,66,
    14,73,14,71,12,68,12,79,12,71,12,83,1Ø,7Ø,1Ø,73
36Ø DATA 1Ø,71,1Ø,72,1Ø,84,8,65,8,78,8,73,8,77,8,6
    5,8,76,8,83,8,32
37Ø GOSUB 86Ø
38Ø GOSUB 76Ø
39Ø PRINT TAB(8);"NAME THE NOTE"::::
4ØØ FOR T=1 TO 1Ø
41Ø RANDOMIZE
42Ø X=INT(9*RND+1)
43Ø J=6+X
44Ø CALL HCHAR(J,2Ø,1Ø8)
45Ø CALL HCHAR(J,21,1Ø9)
46Ø CALL HCHAR(J+1,19,11Ø)
47Ø CALL HCHAR(J+1,22,112)
48Ø CALL HCHAR(J+2,2Ø,113)
49Ø CALL HCHAR(J+2,21,114)
5ØØ CALL HCHAR(J+1,21,111)
51Ø CALL SOUND(15Ø,1397,4)
52Ø CALL KEY(Ø,K,S)
53Ø IF S<1 THEN 52Ø
54Ø IF K=N(X,1)THEN 57Ø
```

~~~
550 CALL SOUND(200,-5,4)
560 GOTO 520
570 CALL HCHAR(J+1,21,N(X,1))
580 IF X<9 THEN 610
590 CALL SOUND(500,1475,30,1475,30,1475,30,-4,1)
600 GOTO 620
610 CALL SOUND(500,N(X,2),2)
620 CALL SOUND(1,N(X,2),30)
630 IF X/2=INT(X/2)THEN 680
640 CALL HCHAR(J,20,32,2)
650 CALL HCHAR(J+1,19,96,4)
660 CALL HCHAR(J+2,20,32,2)
670 GOTO 710
680 CALL HCHAR(J,20,96,2)
690 CALL HCHAR(J+1,19,32,4)
700 CALL HCHAR(J+2,20,96,2)
710 NEXT T
720 PRINT :"GOOD!  TRY AGAIN? (Y/N)"
730 CALL KEY(0,K,S)
740 IF K=89 THEN 380
750 IF K=78 THEN 900 ELSE 730
760 CALL CLEAR
770 PRINT TAB(10);"BASS CLEF":::
780 PRINT ::"``abc`"&L$:" d{3 SPACES}e k":"`f```g"
    &L$:"{5 SPACES}h k":"`````i"&L$:"{4 SPACES}h":
    "````i`"&L$:"{3 SPACES}j":"``````"&L$:::
790 RETURN
800 READ C,J
810 FOR I=C TO J
820 READ K,G
830 CALL HCHAR(K,I,G)
840 NEXT I
850 RETURN
860 PRINT TAB(16);"PRESS <ENTER>";
870 CALL KEY(0,K,S)
880 IF K<>13 THEN 870
890 RETURN
900 CALL CLEAR
910 END
~~~

Chapter 6
Math Competency

Chapter 6
Math Competency

The six programs in this chapter generate story or word problems for mathematics, much like those found on standardized math tests given in high school. Each program concentrates on one type of word problem, presenting a problem and waiting for a response from you. "Buying Items," for example, displays a list of five items, along with their prices. Questions are then presented, such as asking you to total the items' costs.

Illustrating how word or story problems are formed, these programs could be used by anyone interested in mathematics, from those in the third grade to high school students.

If you enter an incorrect answer while using these programs, you'll see a reminder of how to get to the right answer, and the problem will repeat. If you answer correctly, you'll always have a choice of continuing with another problem or quitting the problem.

Buying Items

Computers: VIC-20; TRS-80 Color Computer; MC-10; TRS-80 Model I; TI-99/4

"Buying Items" gives a list of five items along with their prices. The first question asks you to total all five items. The second question asks which two items could be purchased, given a certain amount of money. This question is in multiple-choice form.

There are three different categories for price lists. The number A is randomly chosen as 1, 2, or 3. School supplies is number 1, a toy store is number 2, and a grocery store is number 3. The items I$ are read in as data in an array I$(A,C), where I$(2,4) would mean the name of an item in category 2 (toys) and the fourth item listed.

The data for each item includes a minimum price I(A,C,1) and a maximum price I(A,C,2). For the actual price list for the problem, the price P is a random number from the minimum to the maximum. A subroutine is used to convert the price calculated as a number of cents to a dollar value for printing in the problem. The price P is a whole number of cents.

A random number F is chosen to be 1 or 2 so that the wording of the problem will vary. If F is 1 the question is, "How much will it cost to buy all the items on the list?" If F is 2, the question has a person's name. Six names are read in as data, chosen at random. If the name number N is greater than 3, the person is a boy; otherwise, a girl; and the printed question uses the appropriate pronouns.

M is a random amount of money based on a range from the *lowest* possible price of the two least expensive items in the list to a *maximum* price that would still be less than any other two items. The correct answer for the multiple-choice question is the phrase H$(A), and it is placed in a random position R, one of the four choices. The incorrect choices use one item from the first three in the list, and one item from the last two in the list. The computer makes sure the four phrases are different.

Here's how the program works on the VIC:

Line	Function
1–2	PRINT title screen, DIMension variables, READ six names from data, and turn on volume for sound.
3	READ from DATA items and minimum and maximum prices, initialize S for sound statements, and branch past subroutines.
4–7	Wait for user to press RETURN.
9	Delay loop for sound.
10–13	Convert cost in cents to dollars.
16	Play music for correct answer.
25	DIMension H and S for multiple-choice answers and define H$ answers.
28	Wait for user to press RETURN.
30	Randomly choose category and initialize total TP = 0.
32–36	PRINT problem, randomly choosing price for each item.
38–41	PRINT one of two forms of the question.
44	Receive answer.
46–48	If answer is incorrect, give the right answer with an explanation.
60	If answer is correct, play sound.
62	Clear lower part of screen.
64–81	PRINT second question.
82–87	PRINT multiple-choice answers.
88	Receive answer.
89–90	If answer is correct, play sound.
92–96	PRINT option to try again and branch appropriately.
100–102	If answer is incorrect, PRINT explanation.
120–122	DATA statements.
200	Clear screen and END.

Program 6-1. Buying Items

VIC-20

(In this program, make sure to use abbreviations for all BASIC keywords in the lines marked with an asterisk (*). If you don't, the program will not run correctly. Refer to Appendix C, "BASIC Keyword Abbreviations," for the proper abbreviations.)

```
 1 PRINT"{CLR}{2 DOWN}{BLU}{3 SPACES}MATH COMPETENC
   Y":PRINT"{2 DOWN}{4 SPACES}BUYING ITEMS{8 DOWN}"
 2 DIMI$(3,5),I(3,5,2),N$(6),J(5):FORC=1TO6:READN$(
   C):NEXT:POKE36878,15
 3 FORA=1TO3:FORC=1TO5:READI$(A,C),I(A,C,1),I(A,C,2
   ):NEXTC,A:S=36876:GOTO25
 4 PRINT"{GRN}PRESS RETURN";
 5 GETA$:IFA$=""THEN5
 6 IFASC(A$)<>13THEN5
 7 RETURN
 9 FORDE=1TO1ØØ:NEXT:RETURN
 1Ø P$=STR$(P):IFLEN(P$)=2THENP$="Ø"+RIGHT$(P$,1)
 11 PR$=RIGHT$(P$,2):PL$=LEFT$(P$,LEN(P$)-2):IFLEN(
    PL$)<2THENPL$="{SHIFT-SPACE} "
 13 P$="$"+PL$+"."+PR$:RETURN
 16 POKES,195:GOSUB9:POKES,2Ø7:GOSUB9:POKES,215:GOS
    UB9
 18 POKES,225:GOSUB9:GOSUB9:POKES,Ø:RETURN
 2Ø DIMH$(3),S$(4):H$(1)="PENCIL AND ERASER"
 25 H$(2)="BALL AND TRUCK":H$(3)="CANDY AND FRUIT"
 28 GOSUB4
 3Ø A=INT(RND(Ø)*3+1):TP=Ø
 32 PRINT"{CLR}{BLU}GIVEN THIS PRICE LIST:{BLK}"
 34 FORC=1TO5:D=I(A,C,2)-I(A,C,1):P=I(A,C,1)+INT(RN
    D(Ø)*D+1):GOSUB1Ø:TP=TP+P
 36 PRINTTAB(4);I$(A,C);TAB(13);P$:NEXT
 38 F=INT(RND(Ø)*2+1)
 39 IFF=1THENPRINT"{DOWN}{BLU}HOW MUCH WILL IT COST
     TO BUY ALL THE ITEMS{2 SPACES}ON THE LIST?":GO
    TO44
 4Ø N=INT(RND(Ø)*6+1):PRINT"{DOWN}{BLU}";N$(N);" WA
    NTS TO BUY":PRINT"EVERYTHING ON THE"
 41 PRINT"LIST.{2 SPACES}WHAT WOULD THE TOTAL COST
    {SPACE}BE?"
 44 PRINT"{DOWN}{RED}$";:INPUT X
 46 IFABS(X-TP/1ØØ)<.ØØ1THEN6Ø
 47 POKES,159:GOSUB9:POKES,135:GOSUB9:POKES,Ø
*48 PRINT"{PUR}{DOWN}ADD ALL FIVE NUMBERS. THE TOTA
    L IS ";:P=TP:GOSUB1Ø:PRINTP$:PRINT"{DOWN}":GOSU
    B4:GOTO3Ø
 6Ø GOSUB16
 62 FORD=7834TO8185:POKED,32:NEXT
 64 PRINT"{6 UP}{BLU}":IFF=1THENPRINT"IF YOU COULD
    {SPACE}ONLY":GOTO7Ø
```

```
65 PRINT"{UP}IF ";N$(N);" COULD ONLY"
7Ø IFA=1THENM=INT(RND(1)*5)+25:GOTO76
72 IFA=2THENM=INT(RND(1)*36)+239:GOTO76
74 M=INT(RND(1)*18)+1ØØ
76 P=M:GOSUB1Ø:IFF=1THENPRINT"{2 UP}SPEND ";P$:GOT
   O78
77 PRINT"SPEND ";P$
78 PRINT"WHICH OF THESE PAIRS{2 SPACES}OF ITEMS ON
    THE LIST"
79 IFF=1THENPRINT"COULD YOU BUY?":GOTO82
8Ø IFN<4THENPRINT"COULD SHE BUY?":GOTO82
81 PRINT"COULD HE BUY?"
82 R=INT(RND(1)*4)+1:PRINT"{DOWN}":FORV=1TO4:IFV=R
   THENS$(V)=H$(A):GOTO87
83 X=INT(RND(1)*2)+4:S$(V)=I$(A,X):X=INT(RND(1)*3)
   +1:S$(V)=S$(V)+" AND "+I$(A,X)
84 IFV=1THEN87
85 FORV1=1TOV-1:IFS$(V1)=S$(V)THEN83
86 NEXTV1
87 PRINTCHR$(64+V);" "+S$(V):NEXTV:PRINT"{RED}";
88 GETA$:IFA$=""THEN88
89 PRINTA$:IFASC(A$)<>64+RTHEN1ØØ
9Ø GOSUB16
92 PRINT"{PUR}{DOWN}TRY AGAIN?(Y/N)";
93 GETA$:IFA$=""THEN93
94 IFA$="Y"THEN3Ø
95 IFA$="N"THEN2ØØ
96 GOTO93
1ØØ POKES,159:GOSUB9:POKES,135:GOSUB9:POKES,Ø
1Ø2 PRINT"{PUR}THE TOTAL OF THE TWO{2 SPACES}ITEMS
     MUST BE LESS{4 SPACES}THAN ";P$:GOTO92
12Ø DATANANCY,CINDY,CHERY,DAVID,RANDY,LANCE,PENCIL
    ,8,15,ERASER,2,1Ø
121 DATA NOTEBOOK,35,99,RULER,29
122 DATA49,PAPER,59,9Ø,DOLL,249,599,BALL,49,89,TRU
    CK,1ØØ,15Ø,GAME,27Ø,5ØØ,MODEL,3ØØ,7ØØ
123 DATACANDY,2Ø,5Ø,MEAT,123,425,FRUIT,24,5Ø,CHIPS
    ,1ØØ,257,BREAD,1ØØ,179
2ØØ PRINT"{CLR}{BLU}":END
```

The program looks and works differently on the Color Computer and MC-10:

Line	Function
1	DIMension variables, READ from DATA the items and minimum and maximum prices.
2	Define H$ for multiple-choice answers and branch past subroutines.

3–6 Wait for user to press ENTER.
7–10 Convert cost in cents to dollars.
11–12 Clear screen, initialize total price TP, randomly choose category, and PRINT list of items with random prices.
13 Ask question, receive answer and compare input to correct answer.
14 If input answer was incorrect, show how to get correct answer and return for another problem.
15 If answer is correct, PRINT message and clear lower part of screen.
16–18 Randomly choose cost.
19–24 PRINT multiple-choice question.
25 Wait for user to press a key for answer.
26–27 If answer is incorrect, give correct answer.
28 If answer is correct, PRINT message.
29–31 PRINT option to try again and branch appropriately.
32–33 DATA statements.
34 Clear screen and END.

Program 6-2. Buying Items

TRS-80 Color Computer; MC-10; TRS-80 Model I (with changes)

```
1 DIMI$(3,5),I(3,5,2),J(5),H$(3),S$(4):FORA=1TO3:F
  ORC=1TO5:READI$(A,C),I(A,C,1),I(A,C,2):NEXTC,A
2 H$(1)="PENCIL AND ERASER":H$(2)="BALL AND TRUCK"
  :H$(3)="CANDY AND FRUIT":GOTO11
3 PRINT@495,"PRESS <ENTER>";
4 A$=INKEY$:IFA$=""THEN4
5 IF ASC(A$)<>13 THEN 4
6 RETURN
7 P$=STR$(P)
8 IFLEN(P$)=2THENP$=" Ø"+RIGHT$(P$,1)
9 PR$=RIGHT$(P$,2):PL$=LEFT$(P$,LEN(P$)-2):IFLEN(P
  L$)<2THENPL$=" "+PL$
1Ø P$="$"+PL$+"."+PR$:RETURN
11 CLS:TP=Ø:A=RND(3):PRINT"GIVEN THIS PRICE LIST:"
12 FORC=1TO5:D=I(A,C,2)-I(A,C,1):P=I(A,C,1)+RND(D)
   :GOSUB7:TP=TP+P:PRINTTAB(4);I$(A,C);TAB(13);P$:
   NEXT:PRINT
13 PRINT"HOW MUCH WILL IT COST TO BUY{4 SPACES}ALL
    THE ITEMS ON THE LIST?":PRINT"$";:INPUTX:IFABS
   (X-TP/1ØØ)<.ØØ1THEN15
14 PRINT:PRINT"ADD ALL FIVE NUMBERS.":PRINT"THE TO
   TAL IS ";:P=TP:GOSUB7:PRINTP$:GOSUB3:GOTO11
```

```
15 PRINT:PRINT"CORRECT!":FORC=1TO1ØØ:NEXT:FORC=1TO
   8:PRINT@192+32*C," ":NEXT
16 IFA=1THENM=RND(5)+25:GOTO19
17 IFA=2THENM=RND(36)+239:GOTO19
18 M=RND(18)+1ØØ
19 P=M:GOSUB7:PRINT@224,"IF YOU COULD ONLY SPEND "
   ;P$:PRINT"WHICH OF THESE PAIRS OF ITEMS
   {3 SPACES}ON THE LIST COULD YOU BUY?"
2Ø R=RND(4):FORV=1TO4:IFV=R THENS$(V)=H$(A):GOTO24
21 X=RND(2)+3:S$(V)=I$(A,X):X=RND(3):S$(V)=S$(V)+"
    AND "+I$(A,X):IFV=1THEN24
22 FORV1=1TOV-1:IFS$(V1)=S$(V)THEN21
23 NEXTV1
24 PRINTTAB(3);CHR$(64+V);" "+S$(V):NEXTV
25 A$=INKEY$:IFA$=""THEN25
26 PRINTA$:IFASC(A$)=64+R THEN28
27 PRINT"THE TOTAL OF THE TWO ITEMS MUST BE LESS T
   HAN ";P$;"--";CHR$(64+R):GOTO29
28 PRINT"CORRECT!"
29 PRINT@48Ø,"TRY AGAIN? (Y/N)";
3Ø A$=INKEY$:IFA$="Y"THEN11
31 IFA$<>"N"THEN3Ø
32 DATAPENCIL,8,15,ERASER,2,1Ø,NOTEBOOK,35,99,RULE
   R,29,49,PAPER,59,9Ø,DOLL,249,599
33 DATA BALL,49,89,TRUCK,1ØØ,15Ø,GAME,27Ø,5ØØ,MODE
   L,3ØØ,7ØØ,CANDY,2Ø,5Ø,MEAT,123,425,FRUIT,24,5Ø,
   CHIPS,1ØØ,257,BREAD,1ØØ,179
34 CLS:END
```

For the TRS-80 Model I computer, make the following changes in the above listing:

Insert CLEAR 100 at the beginning of line 1.
Change line 3 to PRINT: PRINT "PRESS <ENTER>";
In line 15 change @192+32*C to @320+64*C
In line 19 change @224 to @448
In line 29 delete @480.

Buying Items on the TI-99/4A works this way:

Line	Function
100–150	Clear screen; PRINT title screen; define characters and colors.
160	DIMension variables.
170–290	READ from DATA six names N$ and items I$ with minimum and maximum prices I.
300–320	Define H$ for multiple-choice answers.
330	Branch past subroutines.

340–370	Wait for ENTER key to be pressed.
380–400	Play music for incorrect answer.
410–450	Play music for correct answer.
460–540	Convert cost in cents to dollars.
550–570	Randomly choose one of three categories and initialize total cost TP=0.
580–660	PRINT problem, randomly choosing price for each item.
670–720	Draw a colored box around list.
730–820	PRINT one of two forms of the question.
830	Receive answer.
840–910	If answer is incorrect, give the right answer with an explanation and return for another problem.
920–930	If answer is correct, play sound then clear lower part of screen.
940–1150	PRINT second question.
1160–1300	PRINT multiple-choice answers.
1310–1340	Receive answer.
1350–1360	If answer is correct, play sound.
1370–1400	PRINT option to try again and branch appropriately.
1410–1440	If answer is incorrect, play different sound effect and show correct answer.
1450–1460	Clear screen and END.

Program 6-3. Buying Items

TI-99/4A

```
1ØØ CALL CLEAR
11Ø PRINT TAB(6);"MATH COMPETENCY"
12Ø CALL CHAR(136,"Ø8Ø4Ø2FFØ2Ø4Ø8")
13Ø PRINT :::TAB(7);"BUYING ITEMS"
14Ø CALL COLOR(14,9,16)
15Ø PRINT ::::::::::
16Ø DIM I$(3,5),I(3,5,2),N$(6),J(5),H$(3),S$(4)
17Ø FOR C=1 TO 6
18Ø READ N$(C)
19Ø NEXT C
2ØØ FOR A=1 TO 3
21Ø FOR C=1 TO 5
22Ø READ I$(A,C),I(A,C,1),I(A,C,2)
23Ø NEXT C
24Ø NEXT A
25Ø DATA ANGIE,CINDY,CHERY,RICKY,BOBBY,RANDY,PENCI
    L,8,15
26Ø DATA ERASER,2,1Ø,NOTEBOOK,35,99,RULER,29,49
27Ø DATA PAPER,59,9Ø,DOLL,249,599,BALL,49,89,TRUCK
    ,1ØØ,15Ø
```

```
280 DATA GAME,27Ø,5ØØ,MODEL,3ØØ,7ØØ,CANDY,2Ø,5Ø
290 DATA MEAT,123,425,FRUIT,24,5Ø,CHIPS,1ØØ,257,BR
    EAD,1ØØ,179
3ØØ H$(1)="PENCIL AND ERASER"
31Ø H$(2)="BALL AND TRUCK"
32Ø H$(3)="CANDY AND FRUIT"
33Ø GOTO 55Ø
34Ø PRINT TAB(15);"PRESS <ENTER>";
35Ø CALL KEY(Ø,K,S)
36Ø IF K<>13 THEN 35Ø
37Ø RETURN
38Ø CALL SOUND(1ØØ,33Ø,2)
39Ø CALL SOUND(15Ø,262,2)
4ØØ RETURN
41Ø CALL SOUND(1ØØ,262,2)
42Ø CALL SOUND(1ØØ,33Ø,2)
43Ø CALL SOUND(1ØØ,392,2)
44Ø CALL SOUND(2ØØ,523,2)
45Ø RETURN
46Ø P$=STR$(P)
47Ø IF LEN(P$)>1 THEN 49Ø
48Ø P$="Ø"&P$
49Ø IF LEN(P$)>2 THEN 51Ø
5ØØ P$=" "&P$
51Ø PR$=SEG$(P$,LEN(P$)-1,2)
52Ø PL$=SEG$(P$,1,LEN(P$)-2)
53Ø P$="$"&PL$&"."&PR$
54Ø RETURN
55Ø RANDOMIZE
56Ø A=INT(RND*3+1)
57Ø TP=Ø
58Ø CALL CLEAR
59Ø PRINT "GIVEN THIS PRICE LIST:"::
6ØØ FOR C=1 TO 5
61Ø D=I(A,C,2)-I(A,C,1)
62Ø P=I(A,C,1)+INT(RND*D+1)
63Ø GOSUB 46Ø
64Ø TP=TP+P
65Ø PRINT TAB(6);I$(A,C);TAB(15);P$
66Ø NEXT C
67Ø R=INT(RND*13+4)
68Ø CALL COLOR(13,R,R)
69Ø CALL HCHAR(18,6,128,18)
7ØØ CALL VCHAR(19,6,128,5)
71Ø CALL VCHAR(19,23,128,5)
72Ø CALL HCHAR(24,6,128,18)
73Ø F=INT(RND*2+1)
74Ø IF F=2 THEN 79Ø
75Ø PRINT ::"HOW MUCH WILL IT COST"
```

```
76Ø PRINT "TO BUY ALL THE ITEMS"
77Ø PRINT "ON THE LIST?"
78Ø GOTO 83Ø
79Ø N=INT(RND*6+1)
8ØØ PRINT ::N$(N);" WANTS TO BUY"
81Ø PRINT "EVERYTHING ON THE LIST."
82Ø PRINT "WHAT WILL THE TOTAL COST BE?"
83Ø INPUT "$":X
84Ø IF ABS(X-TP/1ØØ)<.ØØ1 THEN 92Ø
85Ø GOSUB 38Ø
86Ø PRINT :"ADD ALL FIVE NUMBERS."
87Ø P=TP
88Ø GOSUB 46Ø
89Ø PRINT "THE TOTAL IS ";P$:::
9ØØ GOSUB 34Ø
91Ø GOTO 55Ø
92Ø GOSUB 41Ø
93Ø CALL HCHAR(2Ø,1,32,128)
94Ø IF F=2 THEN 97Ø
95Ø PRINT "IF YOU COULD ONLY SPEND"
96Ø GOTO 98Ø
97Ø PRINT "IF ";N$(N);" COULD ONLY SPEND"
98Ø IF A<>1 THEN 1Ø1Ø
99Ø M=INT(RND*5+25)
1ØØØ GOTO 1Ø5Ø
1Ø1Ø IF A<>2 THEN 1Ø4Ø
1Ø2Ø M=INT(RND*36+239)
1Ø3Ø GOTO 1Ø5Ø
1Ø4Ø M=INT(RND*18+1ØØ)
1Ø5Ø P=M
1Ø6Ø GOSUB 46Ø
1Ø7Ø PRINT P$;", WHICH OF THESE PAIRS"
1Ø8Ø PRINT "OF ITEMS COULD ";
1Ø9Ø IF F<>1 THEN 112Ø
11ØØ PRINT "YOU BUY?"::
111Ø GOTO 116Ø
112Ø IF N>3 THEN 115Ø
113Ø PRINT "SHE BUY?"::
114Ø GOTO 116Ø
115Ø PRINT "HE BUY?"::
116Ø R=INT(RND*4+1)
117Ø FOR V=1 TO 4
118Ø IF V=R THEN 128Ø
119Ø X=INT(RND*2+4)
12ØØ S$(V)=I$(A,X)
121Ø X=INT(RND*3+1)
122Ø S$(V)=S$(V)&" AND "&I$(A,X)
123Ø IF V=1 THEN 129Ø
124Ø FOR V1=1 TO V-1
```

```
125Ø IF S$(V1)=S$(V)THEN 119Ø
126Ø NEXT V1
127Ø GOTO 129Ø
128Ø S$(V)=H$(A)
129Ø PRINT TAB(3);CHR$(64+V);"  "&S$(V)
13ØØ NEXT V
131Ø CALL SOUND(15Ø,1397,2)
132Ø CALL KEY(Ø,K,S)
133Ø IF (K<65)+(K>68)THEN 132Ø
134Ø CALL HCHAR(K-45,4,42)
135Ø IF K<>64+R THEN 141Ø
136Ø GOSUB 41Ø
137Ø PRINT :"TRY AGAIN? (Y/N)";
138Ø CALL KEY(Ø,K,S)
139Ø IF K=89 THEN 55Ø
14ØØ IF K=78 THEN 145Ø ELSE 138Ø
141Ø GOSUB 38Ø
142Ø CALL HCHAR(19+R,3,136)
143Ø PRINT :"THE TOTAL OF THE TWO ITEMS  MUST BE L
     ESS THAN ";P$
144Ø GOTO 137Ø
145Ø CALL CLEAR
146Ø END
```

Sales Tax

Computers: VIC-20; TRS-80 Color Computer; MC-10; TRS-80 Model I; TI-99/4A

This word problem program shows you how to calculate sales tax on purchased items. One of three categories is chosen randomly, and then four items are listed with random prices. Sales tax is selected randomly, from 2 to 5 percent. You enter the total cost of the items, including sales tax as your answer.

Here's how the three versions of the program work:

Line	Function
110–200	READ from DATA the title of store T$, item P$(I,J), and base price P(I,J).
210	Clear screen.
220	Initialize total cost A.
230–280	PRINT problem with random tax T.
290–310	PRINT random title or kind of store.
320–370	PRINT four items with random prices.
380–410	Draw separator line and ask for total cost.
420–430	Calculate total, plus tax. If input value is correct, shift to appropriate line.

440–570	If input value is incorrect, show how to get right answer and return for another problem.
580–650	If answer is correct, present option to try again and branch appropriately.
660–690	Convert number to dollars and cents for printing.
700–710	Clear screen and END.

Program 6-4. Sales Tax

VIC-20

```
110 FORI=1TO4
120 READT$(I)
130 FORJ=1TO4
140 READP$(I,J),P(I,J)
150 NEXTJ
160 NEXTI
170 DATAHARDWARE,HAMMER,1500,PLIERS,300,SAW,600,NA
    ILS,100
180 DATACLOTHES,BELT,400,TIE,500,SHIRT,600,PANTS,2
    000
190 DATATOYS,BALL,100,CAR,200,GAME,500,DOLL,600
200 DATASUPPLIES,PAPER,400,CLIPS,100,PENCILS,100,E
    NVELOPES,200
210 PRINT"{CLR}"
220 A=0
230 PRINT"SALES TAX ON THE"
240 T=INT(4*RND(0)+2)
250 PRINT"FOLLOWING ITEMS IS":PRINTT;"PER CENT, OR
     ";"$.0"+RIGHT$(STR$(T),1)
260 PRINT"FOR EACH DOLLAR SPENT."
270 PRINT"WHAT IS THE TOTAL":PRINT"COST?"
280 PRINT
290 I=INT(4*RND(0)+1)
300 PRINTTAB(5);T$(I)
310 PRINT
320 FORJ=1TO4
330 C=P(I,J)+25*(INT(4*RND(0)))
340 GOSUB660
350 PRINTP$(I,J),C$
360 A=A+C
370 NEXTJ
380 PRINT"####################"
390 PRINT
400 INPUT "TOTAL COST = $";B
410 PRINT
420 TX=1+T/100
430 TA=INT(((A/100)*TX+.005)*100)/100
440 IF ABS(B-TA)<.01 THEN 580
450 C=A
```

```
46Ø GOSUB66Ø
47Ø PRINT"ADD COSTS FOR TOTAL."
48Ø PRINT"PRICE OF ITEMS = ":PRINTC$
49Ø PRINT"MULTIPLY BY ";T/1ØØ;" FOR"
5ØØ PRINT"TAX, THEN ADD."
51Ø PRINT
52Ø C=TA*1ØØ
53Ø GOSUB66Ø
54Ø PRINT"TOTAL COST ={SHIFT-SPACE}";C$
55Ø PRINT
56Ø INPUT "PRESS <RETURN>";E$
57Ø GOTO21Ø
58Ø PRINT"CORRECT!"
59Ø PRINT
6ØØ PRINT"CHOOSE: 1 TRY AGAIN"
61Ø PRINTTAB(8);"2 END"
62Ø GET E$
63Ø IF E$="1" THEN 21Ø
64Ø IF E$="2" THEN 7ØØ
65Ø GOTO62Ø
66Ø C$=STR$(C)
67Ø L=LEN(C$)
68Ø C$="$"+LEFT$(C$,L-2)+"."+RIGHT$(C$,2)
69Ø RETURN
7ØØ PRINT"{CLR}"
71Ø END
```

Program 6-5. Sales Tax

TRS-80 Color Computer; MC-10; TRS-80 Model I

```
1Ø5 DIMT$(4),P$(4,4),P(4,4)
11Ø FOR I=1 TO 4
12Ø READ T$(I)
13Ø FOR J=1 TO 4
14Ø READ P$(I,J),P(I,J)
15Ø NEXT J
16Ø NEXT I
17Ø DATA HARDWARE,HAMMER,15ØØ,PLIERS,3ØØ,SAW
    ,6ØØ,SCREWDRIVER,1ØØ
18Ø DATA CLOTHES,BELT,4ØØ,TIE,5ØØ,SHIRT,6ØØ,
    PANTS,2ØØØ
19Ø DATA TOYS,BALL,1ØØ,CAR,2ØØ,GAME,5ØØ,DOLL
    ,6ØØ
2ØØ DATA SUPPLIES,PAPER,4ØØ,CLIPS,1ØØ,PENCIL
    S,1ØØ,ENVELOPES,2ØØ
21Ø CLS
22Ø A=Ø
23Ø PRINT "SALES TAX ON THE FOLLOWING"
24Ø T=RND(4)+2
25Ø PRINT "ITEMS IS";T;"PER CENT, OR"
```

```
260 PRINT "$.Ø"+RIGHT$(STR$(T),1)+" FOR EACH
     DOLLAR SPENT."
27Ø PRINT "WHAT IS THE TOTAL COST?"
28Ø PRINT
29Ø I=RND(4)
3ØØ PRINT TAB(8);T$(I)
31Ø PRINT
32Ø FOR J=1 TO 4
33Ø C=P(I,J)+25*(RND(4)-1)
34Ø GOSUB 66Ø
35Ø PRINT P$(I,J),C$
36Ø A=A+C
37Ø NEXT J
38Ø PRINT "#########################"
39Ø PRINT
4ØØ INPUT "TOTAL COST = $ ";B
41Ø PRINT
42Ø TX=1+T/1ØØ
43Ø TA=INT(((A/1ØØ)*TX+.ØØ5)*1ØØ)/1ØØ
44Ø IF ABS(B-TA)<.Ø1 THEN 58Ø
45Ø C=A
46Ø GOSUB 66Ø
47Ø PRINT "ADD COSTS FOR TOTAL."
48Ø PRINT "PRICE OF ITEMS = ";C$
49Ø PRINT "MULTIPLY BY ";T/1ØØ;" FOR TAX,"
5ØØ PRINT "THEN ADD."
51Ø PRINT
52Ø C=TA*1ØØ
53Ø GOSUB 66Ø
54Ø PRINT "TOTAL COST = ";C$
55Ø PRINT
56Ø INPUT "PRESS <ENTER>";E$
57Ø GOTO 21Ø
58Ø PRINT "CORRECT!"
59Ø PRINT
6ØØ PRINT "CHOOSE: 1 TRY AGAIN"
61Ø PRINT TAB(8);"2 END"
62Ø E$=INKEY$
63Ø IF E$="1" THEN 21Ø
64Ø IF E$="2" THEN 7ØØ
65Ø GOTO 62Ø
66Ø C$=STR$(C)
67Ø L=LEN(C$)
68Ø C$="$"+LEFT$(C$,L-2)+"."+RIGHT$(C$,2)
69Ø RETURN
7ØØ CLS
71Ø END
```

Program 6-6. Sales Tax

TI-99/4A

```
110 FOR I=1 TO 4
120 READ T$(I)
130 FOR J=1 TO 4
140 READ P$(I,J),P(I,J)
150 NEXT J
160 NEXT I
170 DATA HARDWARE,HAMMER,1500,PLIERS,300,SAW,600,N
    AILS,100
180 DATA CLOTHES,BELT,400,TIE,500,SHIRT,600,PANTS,
    2000
190 DATA TOYS,BALL,100,CAR,200,GAME,500,DOLL,600
200 DATA SUPPLIES,PAPER,400,CLIPS,100,PENCILS,100,
    ENVELOPES,200
210 CALL CLEAR
220 A=0
230 PRINT "SALES TAX ON THE FOLLOWING"
235 RANDOMIZE
240 T=INT(4*RND+2)
250 PRINT "ITEMS IS";T;"PER CENT, OR"
260 PRINT "$.0";STR$(T);" FOR EACH DOLLAR SPENT."
270 PRINT "WHAT IS THE TOTAL COST?"
280 PRINT
290 I=INT(4*RND+1)
300 PRINT TAB(8);T$(I)
310 PRINT
320 FOR J=1 TO 4
330 C=P(I,J)+25*(INT(4*RND))
340 GOSUB 660
350 PRINT P$(I,J),C$
360 A=A+C
370 NEXT J
380 PRINT "########################"
390 PRINT
400 INPUT "TOTAL COST = $ ":B
410 PRINT
420 TX=1+T/100
430 TA=INT(((A/100)*TX+.005)*100)/100
440 IF ABS(B-TA)<.01 THEN 580
450 C=A
460 GOSUB 660
470 PRINT "ADD COSTS FOR TOTAL."
480 PRINT "PRICE OF ITEMS = ";C$
490 PRINT "MULTIPLY BY ";T/100;" FOR TAX,"
500 PRINT "THEN ADD."
510 PRINT
520 C=TA*100
530 GOSUB 660
```

```
540 PRINT "TOTAL COST = ";C$
550 PRINT
560 INPUT "PRESS <ENTER>":E$
570 GOTO 210
580 PRINT "CORRECT!"
590 PRINT
600 PRINT "CHOOSE: 1 TRY AGAIN"
610 PRINT TAB(9);"2 END"
620 CALL KEY(0,K,S)
630 IF K=49 THEN 210
640 IF K=50 THEN 700
650 GOTO 620
660 C$=STR$(C)
670 L=LEN(C$)
680 C$="$"&SEG$(C$,1,L-2)&"."&SEG$(C$,L-1,2)
690 RETURN
700 CALL CLEAR
710 END
```

Earning Money

Computers: VIC-20; TRS-80 Color Computer; MC-10; TRS-80 Model I;
TI-99/4A

"Earning Money," another math competency program, provides you with word problems that use addition and multiplication. The program presents two kinds of problems for you to look at. The first are relatively simple, asking you to calculate how much money a person makes in a week, given an hourly wage and the number of hours worked, for example. Once you answer correctly, you'll have the option of continuing with the simpler problems, or moving on to more difficult ones.

This second group of word problems is more complicated, asking you to calculate the money earned in several weeks, for instance, giving you the amount earned each hour, day, or week. As with the first group of problems, you'll use multiplication and addition to arrive at your answer. If you enter an incorrect answer, the program will show you how the total should have been computed, and then ask if you want to continue. You'll have the choice of going on or quitting the program entirely.

The names of the people in the problems are read in as N$(I) and T$(I), where I is a subscript from 0 to 5. The ways of earning money are read in as phrases J$(I).

The wage earned is a random number from $1 to $3.25 in amounts divisible by 25¢.

A name is chosen with the random number N, and the number of hours in the first problem is a random number H. For the second type of problem, the number of weeks is a random number W from 2 to 20 weeks. The third type of problem chooses a random name, a random job, and a random number of weeks W from 2 to 9 weeks.

The VIC version works like this:

Line	Function
2–4	PRINT title screen and DIMension variables name N$, job J$, and name T$.
6	READ from DATA the names and jobs, turn on volume for sound, initialize S for sound, and branch.
10–13	Wait for RETURN key to be pressed.
14–15	Convert price to dollar and cent value.
18	Delay loop for sounds.
20	Play sounds for incorrect answer.
22	Play sounds for correct answer.
28	Wait for RETURN key to be pressed.
30	Randomly choose name, hours, and wage.
32–40	PRINT problem and check input answer.
42–47	Procedure for correct answer. PRINT option to try again or continue and then branch appropriately.
49–52	If answer is incorrect, show how to get correct answer and return for another problem.
54–64	Randomly choose variables and PRINT next problem. Compare input answer.
66–69	Procedure for correct answer. PRINT option to try again or continue and then branch appropriately.
70–72	If answer is incorrect, show how to get correct answer and return for another problem of same type.
80–88	Randomly choose variables and PRINT next problem.
90	Input answer and compare with correct answer.
92–95	Procedure for correct answer. PRINT option to try again and branch appropriately.
97–98	If answer is incorrect, show how to get correct answer, and then return for another problem of same type.
100–102	DATA statement.
110	Clear screen and END.

Program 6-7. Earning Money

VIC-20

(In this program, make sure to use abbreviations for all BASIC keywords in the lines marked with an asterisk (*). If you don't, the program will not run correctly. Refer to Appendix C, "BASIC Keyword Abbreviations," for the proper abbreviations.)

```
 2 PRINT"{CLR}{BLU}{2 DOWN}{3 SPACES}MATH COMPETENC
   Y":PRINT"{3 DOWN}{4 SPACES}EARNING MONEY{9 DOWN}
   "
 4 DIMN$(5),J$(5),T$(5)
 6 FORI=ØTO5:READN$(I),J$(I),T$(I):NEXT:POKE36878,1
   5:S=36876:GOTO28
 1Ø PRINT"{DOWN}{GRN}PRESS RETURN";
 11 GETA$:IFA$=""THEN11
 12 IFASC(A$)<>13THEN11
 13 PRINT"{CLR}{BLU}":RETURN
 14 P=1ØØ+25*INT(RND(Ø)*1Ø)
 15 P$=STR$(P):PL$=LEFT$(P$,LEN(P$)-2):PR$=RIGHT$(P
    $,2):P$="$"+PL$+"."+PR$:RETURN
 18 FORV=1TO1ØØ:NEXT:RETURN
 2Ø POKES,159:GOSUB18:POKES,135:GOSUB18:POKES,Ø:RET
    URN
*22 POKES,195:GOSUB18:POKES,2Ø7:GOSUB18:POKES,215:G
    OSUB18:POKES,225:GOSUB18:GOSUB18:POKES,Ø:RETURN
 28 GOSUB1Ø
 3Ø N=INT(RND(Ø)*6):H=8+INT(RND(Ø)*11):GOSUB14
 32 PRINT"{CLR}{BLU}";N$(N);" WORKS";H;"HOURS":IFN<
    3THENPRINT"PER WEEK.{2 SPACES}HE EARNS":GOTO36
 34 PRINT"PER WEEK.{2 SPACES}SHE EARNS"
 36 PRINTP$;" PER HOUR.":IFN<3THENPRINT"HOW MUCH DO
    ES HE EARN":GOTO4Ø
 38 PRINT"HOW MUCH DOES SHE EARN";
 4Ø PRINT"IN A WEEK?{DOWN}{BLK}":INPUT"$";D:D1=P*H/
    1ØØ:IFABS(D-D1)>.ØØ1THEN49
 42 GOSUB22:PRINT"{4 DOWN}"
 44 PRINT"{GRN}PRESS 1 TO TRY AGAIN{8 SPACES}2 TO C
    ONTINUE"
 45 GETA$:IFA$="1"THEN3Ø
 46 IFA$="2"THEN54
 47 GOTO45
 49 GOSUB2Ø
*5Ø PRINT"{DOWN}{RED}MULTIPLY";H;"HOURS BY":PRINTP$
    ;" PER HOUR.":P=H*P:GOSUB15:PRINT"{DOWN}THE ANS
    WER IS ";P$
 52 PRINT"{3 DOWN}":GOTO28
 54 PRINT"{CLR}{BLU}";:N=INT(RND(Ø)*5):H=INT(RND(Ø)
    *11)+8:GOSUB14:PRINTN$(N);" EARNS ";P$;" PER"
 56 IFN<3THENPRINT"HOUR.{2 SPACES}HE WORKS";:GOTO6Ø
 58 PRINT"HOUR.{2 SPACES}SHE WORKS";
 6Ø PRINTH:PRINT"HOURS PER WEEK.{2 SPACES}HOW":IFN<
    3THENPRINT"MUCH WILL HE EARN IN":GOTO64
 62 PRINT"MUCH WILL SHE EARN IN"
 64 W=INT(RND(Ø)*19+2):PRINTW;"WEEKS?{DOWN}{BLK}":I
    NPUT"$";D:D1=P*H*W/1ØØ:IFABS(D-D1)>.ØØ1THEN7Ø
 66 GOSUB22:PRINT"{4 DOWN}{GRN}PRESS 1 TO TRY AGAIN
    {8 SPACES}2 TO CONTINUE"
 67 GETA$:IFA$="1"THEN54
 68 IFA$="2"THEN8Ø
 69 GOTO67
*7Ø GOSUB2Ø:PRINT"{DOWN}{RED}MULTIPLY";H;"HOURS BY"
    :PRINTP$;" PER HOUR. THEN":PRINT"MULTIPLY BY";W
    ;"WEEKS."
 72 PRINT"{DOWN}THE ANSWER IS ";:P=H*P*W:GOSUB15:PR
    INTP$:PRINT"{3 DOWN}":GOSUB1Ø:GOTO54
 8Ø J=INT(RND(Ø)*5):T=INT(RND(Ø)*5):GOSUB14:W=INT(R
    ND(Ø)*8)+2
 82 PRINT"{CLR}{BLU}";T$(T);" EARNED ";P$:PRINT"LAS
    T WEEK":PRINTJ$(J)
 84 IFT<3THENPRINT"{DOWN}IF HE EARNED THIS":GOTO88
 86 PRINT"{DOWN}IF SHE EARNED THIS"
 88 PRINT"AMOUNT EVERY WEEK,":PRINT"WHAT WOULD THE
    {SPACE}TOTAL{2 SPACES}INCOME BE FOR";W;"WEEKS?
    {DOWN}{BLK}"
 9Ø INPUT"$";D:D1=P*W/1ØØ:IFABS(D-D1)>.ØØ1THEN97
 92 GOSUB22:PRINT"{4 DOWN}{GRN}TRY AGAIN? (Y/N)"
 93 GETA$:IFA$="Y"THEN8Ø
 94 IFA$="N"THEN11Ø
 95 GOTO93
 97 GOSUB2Ø:PRINT"{DOWN}{RED}MULTIPLY ";P$;" PER":P
    RINT"WEEK BY";W;"WEEKS.":P=P*W:GOSUB15
 98 PRINT"{DOWN}THE ANSWER IS ";P$:PRINT"{DOWN}":GO
    SUB1Ø:GOTO8Ø
 1ØØ DATASAM,DOING ODD JOBS.,PAUL,JOE,MOWING LAWNS.
     ,JACK,BOB,TENDING CHILDREN,MARK,ANN
 1Ø2 DATARUNNING ERRANDS,JANE,SUE,DOING HOUSEWORK,J
     UDY,KAY,DELIVERING ADS,DAWN
 11Ø PRINT"{CLR}{BLU}":END
```

Earning Money on the Color Computer works this way:

Line	Function
10	Clear screen, PRINT title screen, DIMension variables, READ from DATA the name N$, job J$, and name T$, and branch past subroutines.
20–50	Wait for ENTER key to be pressed.
60–100	Convert price to dollar value.
110–160	Clear screen and PRINT problem, randomly choosing name, hours, and wage.

170 Receive answer, calculate correct answer, and compare answers.
180 Procedure for correct answer.
190–220 PRINT option to try again and branch appropriately.
230 If answer is incorrect, show how to get correct answer and return for another problem.
240–300 PRINT problem, receive student's answer, calculate correct answer, and compare answers.
310 PRINT message for correct answer.
320–340 PRINT option to try again and branch appropriately.
350–360 If answer is incorrect, show how to get correct answer and return for another problem of same type.
370–410 PRINT problem.
420 Receive student's answer, calculate correct answer, and compare answers.
430 PRINT message for correct answer.
440–460 PRINT option to try again and branch appropriately.
470 If answer is incorrect, show how to get correct answer and return for another problem of same type.
480–490 DATA statements.
500 Clear screen and END.

Program 6-8. Earning Money

TRS-80 Color Computer; MC-10; TRS-80 Model I (with changes)

```
10 CLS:PRINT"MATH COMPETENCY":PRINT"EARNING MONEY"
   :DIMN$(5),J$(5),T$(5):FORI=ØTO5:READN$(I),J$(I)
   ,T$(I):NEXT:FORI=1TO5ØØ:NEXT:GOTO11Ø
2Ø PRINT @495,"PRESS <ENTER>";
3Ø A$=INKEY$:IF A$="" THEN 3Ø
4Ø IF ASC(A$)<>13 THEN 3Ø
5Ø RETURN
6Ø P=1ØØ+25*RND(1Ø)
7Ø P$=STR$(P)
8Ø IF LEN(P$)=2 THEN P$=" Ø"+RIGHT$(P$,1)
9Ø PR$=RIGHT$(P$,2):PL$=LEFT$(P$,LEN(P$)-2):IFLEN(
   PL$)<2THENPL$="  "+PL$
1ØØ P$="$"+PL$+"."+PR$:RETURN
11Ø CLS:N=RND(6)-1:H=8+RND(1Ø):GOSUB6Ø:PRINTN$(N);
    " WORKS";H;"HOURS PER WEEK.":IFN<3THENPRINT"HE
     EARNS ";:GOTO13Ø
12Ø PRINT "SHE EARNS ";
13Ø PRINT P$;" PER HOUR.":PRINT
14Ø IF N<3 THEN PRINT "HOW MUCH DOES HE EARN":GOTO
     16Ø
15Ø PRINT "HOW MUCH DOES SHE EARN"
```

```
160 PRINT "IN A WEEK?  $";
170 INPUTD:D1=P*H/100:IF ABS(D-D1)>.001 THEN 230
180 PRINT"CORRECT!":FOR C=1 TO 100:NEXT
190 PRINT @480,"TRY AGAIN? (Y/N)";
200 A$=INKEY$:IF A$="Y" THEN 110
210 IF A$="N" THEN 240
220 GOTO200
230 PRINT:PRINT"MULTIPLY";H;"HOURS BY ";P$:PRINT"P
    ER HOUR.":P=H*P:GOSUB70:PRINT"THE ANSWER IS ";
    P$:GOSUB20:GOTO110
240 CLS:N=RND(6)-1:H=8+RND(10):GOSUB60:PRINTN$(N);
    " EARNS ";P$;" PER HOUR.":PRINT
250 IF N<3 THEN PRINT "HE WORKS";:GOTO 270
260 PRINT "SHE WORKS";
270 PRINT H;"HOURS PER WEEK.":PRINT
280 IF N<3 THEN PRINT "HOW MUCH WILL HE EARN IN":G
    OTO 300
290 PRINT "HOW MUCH WILL SHE EARN IN"
300 W=RND(19)+1:PRINT W;"WEEKS?  $";:INPUTD:D1=P*H
    *W/100:IFABS(D-D1)>.001THEN350
310 PRINT"CORRECT!"
320 PRINT @480,"TRY AGAIN? (Y/N)";
330 A$=INKEY$:IF A$="Y" THEN 240
340 IF A$="N" THEN 370 ELSE 330
350 PRINT:PRINT"MULTIPLY";H;"HOURS BY ";P$:PRINT"P
    ER HOUR.":PRINT:PRINT"THEN MULTIPLY BY";W;"WEE
    KS.":P=H*P*W:GOSUB70
360 PRINT:PRINT"THE ANSWER IS ";P$:GOSUB20:GOTO240
370 CLS:J=RND(6)-1:T=RND(6)-1:GOSUB60:W=RND(8)+1
380 PRINT T$(T);" EARNED ";P$;" LAST WEEK":PRINT J
    $(J);".":PRINT
390 IF T<3 THEN PRINT "IF HE EARNED THIS AMOUNT":G
    OTO 410
400 PRINT "IF SHE EARNED THIS AMOUNT"
410 PRINT "EVERY WEEK, WHAT WOULD THE TOTALINCOME
    BE FOR";W;"WEEKS?"
420 INPUT "$";D:D1=P*W/100:IF ABS(D-D1)>.001 THEN
    470
430 PRINT "CORRECT!"
440 PRINT @480,"TRY AGAIN? (Y/N)";
450 A$=INKEY$:IF A$="Y" THEN 370
460 IF A$="N" THEN 500 ELSE 450
470 PRINT:PRINT"MULTIPLY ";P$;" PER WEEK":PRINT"BY
    ";W;"WEEKS.":P=P*W:GOSUB70:PRINT:PRINT"THE ANS
    WER IS ";P$:GOSUB20:GOTO370
480 DATA SAM,DOING ODD JOBS,JOHN,JOE,MOWING LAWNS,
    ANDY,BOB,TENDING CHILDREN,MARK,ANN
490 DATA RUNNING ERRANDS,LENA,SUE,DOING HOUSEWORK,
    AURA,KAY,DELIVERING ADS,DAWN
500 CLS:END
```

For the TRS-80 Model I version, make the following changes in the above listing:

Change line 20 to PRINT:PRINT "PRESS <ENTER>";

Delete @480, in lines 190, 320, and 440.

Here's the TI-99/4A version:

Line	Function
100–130	Clear screen and PRINT title screen.
140	DIMension variables for name N$, job J$, and name T$.
150–190	READ from DATA for names and jobs.
200	Branch past subroutines.
210–240	Wait for ENTER to be pressed.
250–270	Play sound for incorrect answer.
280–320	Play music for correct answer.
330–360	Convert number to dollars.
370–520	Clear screen, PRINT problem with random name, hours, and wage.
530	Receive answer.
540	Calculate correct answer.
550–560	If answer is correct, play sound effect.
570–600	PRINT option to try again and branch appropriately.
610–670	If answer is incorrect, play sound and show how to get correct answer; return for another problem.
680–840	Clear screen and PRINT problem with random name, wage, and weeks.
850	Receive answer.
860	Calculate correct answer.
870–880	If answer is correct, play appropriate sound.
890–920	PRINT option to try again and branch appropriately.
930–1020	If answer is incorrect, play sound, show how to get correct answer, and then return for another problem.
1030–1160	Clear screen and PRINT problem with random name, wage, and weeks.
1170	Receive answer.
1180	Calculate correct answer.
1190–1200	If answer is correct, play musical effect.
1210–1240	PRINT option to try again and branch appropriately.
1250–1320	If answer is incorrect, play sound and show how to get correct answer, and return for another problem of the same type.
1330–1340	Clear screen and END.

Program 6-9. Earning Money

TI-99/4A

```
100 CALL CLEAR
110 PRINT TAB(6);"MATH COMPETENCY"
120 PRINT :::TAB(7);"EARNING MONEY"
130 PRINT :::::::::
140 DIM N$(5),J$(5),T$(5)
150 FOR I=0 TO 5
160 READ N$(I),J$(I),T$(I)
170 NEXT I
180 DATA SAM,DOING ODD JOBS,JOHN,JOE,MOWING LAWNS,
    ANDY,BOB,TENDING CHILDREN,MARK,ANN
190 DATA RUNNING ERRANDS,LENA,SUE,DOING HOUSEWORK,
    AURA,KAY,DELIVERING ADS,DAWN
200 GOTO 370
210 PRINT :TAB(15);"PRESS <ENTER>";
220 CALL KEY(0,K,S)
230 IF K<>13 THEN 220
240 RETURN
250 CALL SOUND(100,330,2)
260 CALL SOUND(150,262,2)
270 RETURN
280 CALL SOUND(100,262,2)
290 CALL SOUND(100,330,2)
300 CALL SOUND(100,392,2)
310 CALL SOUND(200,523,2)
320 RETURN
330 P=100+25*INT(RND*10)
340 P$=STR$(P)
350 P$="$"&SEG$(P$,1,LEN(P$)-2)&"."&SEG$(P$,LEN(P$
    )-1,2)
360 RETURN
370 CALL CLEAR
380 RANDOMIZE
390 N=INT(RND*6)
400 H=8+INT(RND*11)
410 GOSUB 330
420 PRINT N$(N);" WORKS";H;"HOURS PER WEEK."
430 IF N<3 THEN 460
440 PRINT :"SHE EARNS ";
450 GOTO 470
460 PRINT :"HE EARNS ";
470 PRINT P$;" PER HOUR."
480 IF N<3 THEN 510
490 PRINT :"HOW MUCH DOES SHE EARN"
500 GOTO 520
510 PRINT :"HOW MUCH DOES HE EARN"
520 PRINT :"IN A WEEK?"::
530 INPUT "$":D
540 D1=P*H/100
```

```
550 IF ABS(D-D1)>.ØØ1 THEN 61Ø
56Ø GOSUB 28Ø
57Ø PRINT ::"TRY AGAIN? (Y/N)"
58Ø CALL KEY(Ø,K,S)
59Ø IF K=89 THEN 37Ø
6ØØ IF K=78 THEN 68Ø ELSE 58Ø
61Ø GOSUB 25Ø
62Ø PRINT :"MULTIPLY";H;"HOURS BY ";P$::"PER HOUR.
    "
63Ø P=H*P
64Ø GOSUB 34Ø
65Ø PRINT :"THE ANSWER IS ";P$
66Ø GOSUB 21Ø
67Ø GOTO 37Ø
68Ø CALL CLEAR
69Ø RANDOMIZE
7ØØ N=INT(RND*6)
71Ø H=INT(RND*11)+8
72Ø GOSUB 33Ø
73Ø PRINT N$(N);" EARNS ";P$;" PER HOUR."
74Ø IF N<3 THEN 77Ø
75Ø PRINT :"SHE WORKS";
76Ø GOTO 78Ø
77Ø PRINT :"HE WORKS";
78Ø PRINT H;"HOURS PER WEEK."
79Ø IF N<3 THEN 82Ø
8ØØ PRINT :"HOW MUCH WILL SHE EARN IN"
81Ø GOTO 83Ø
82Ø PRINT :"HOW MUCH WILL HE EARN IN"
83Ø W=INT(RND*19)+2
84Ø PRINT :W;"WEEKS?"::
85Ø INPUT "$":D
86Ø D1=P*H*W/1ØØ
87Ø IF ABS(D-D1)>.ØØ1 THEN 93Ø
88Ø GOSUB 28Ø
89Ø PRINT ::"TRY AGAIN? (Y/N)"
9ØØ CALL KEY(Ø,K,S)
91Ø IF K=89 THEN 68Ø
92Ø IF K=78 THEN 1Ø3Ø ELSE 9ØØ
93Ø GOSUB 25Ø
94Ø PRINT :"MULTIPLY";H;"HOURS BY"
95Ø PRINT :P$;" PER HOUR."
96Ø PRINT :"THEN MULTIPLY BY";W;"WEEKS."
97Ø PRINT :"THE ANSWER IS ";
98Ø P=H*P*W
99Ø GOSUB 34Ø
1ØØØ PRINT P$:::
1Ø1Ø GOSUB 21Ø
1Ø2Ø GOTO 68Ø
1Ø3Ø CALL CLEAR
```

```
1Ø4Ø J=INT(RND*6)
1Ø5Ø T=INT(RND*6)
1Ø6Ø GOSUB 33Ø
1Ø7Ø W=INT(RND*8)+2
1Ø8Ø PRINT T$(T);" EARNED ";P$;" LAST WEEK"
1Ø9Ø PRINT :J$(J);"."
11ØØ IF T<3 THEN 113Ø
111Ø PRINT :"IF SHE EARNED THIS AMOUNT"
112Ø GOTO 114Ø
113Ø PRINT :"IF HE EARNED THIS AMOUNT"
114Ø PRINT :"EVERY WEEK, WHAT WOULD THE"
115Ø PRINT :"TOTAL INCOME BE FOR"
116Ø PRINT :W;"WEEKS?"::
117Ø INPUT "$":D
118Ø D1=P*W/1ØØ
119Ø IF ABS(D-D1)>.ØØ1 THEN 125Ø
12ØØ GOSUB 28Ø
121Ø PRINT ::"TRY AGAIN? (Y/N)";
122Ø CALL KEY(Ø,K,S)
123Ø IF K=89 THEN 1Ø3Ø
124Ø IF K=78 THEN 133Ø ELSE 122Ø
125Ø GOSUB 25Ø
126Ø PRINT :"MULTIPLY ";P$;" PER WEEK"
127Ø PRINT :"BY";W;"WEEKS."
128Ø P=P*W
129Ø GOSUB 34Ø
13ØØ PRINT :"THE ANSWER IS ";P$::
131Ø GOSUB 21Ø
132Ø GOTO 1Ø3Ø
133Ø CALL CLEAR
134Ø END
```

Weekly Expenses

Computers: VIC-20; TRS-80 Color Computer; MC-10; TRS-80 Model I;
TI-99/4A

Multiplication and addition are used in this word problem drill program. One of three summer activities is listed on the screen, along with an itemized list of costs for a week. Summer camp, a sports clinic, or a canyon resort are the week-long activities, and you have to add the total weekly cost of your stay. As soon as you enter the correct answer for the addition section of the problem, the program will ask you to calculate a total, given those weekly expenses, for a random number of weeks. When you respond to this part of the problem, you can either continue with another question or quit the program.

All three computer versions work much the same way:

Line	Function
20	DIMension variables for activity A$, name N$, and phrase P$.
30–140	READ from DATA the names, phrases, and activity title and itemized fees.
150	Clear screen.
160–190	Randomly choose one of three places and PRINT introduction.
200–350	Randomly calculate fees for each item and total cost T.
360–390	Get answer.
400–430	If answer is incorrect, give correct total.
440–470	Randomly choose number of weeks and ask for cost.
480–490	Receive answer.
500–600	If answer is incorrect, show how to get right answer and return for another problem.
610	If answer is correct, PRINT message.
620–670	PRINT option to try again and branch appropriately.
680–720	Convert cost to dollars for printing.
730–740	Clear screen and END.

Program 6-10. Weekly Expenses

VIC-20

```
20 DIMA$(3,5),N$(6),P$(3)
30 FORA=1TO3
40 READN$(A),N$(A+3),P$(A)
50 FORB=1TO5
60 READA$(A,B)
70 NEXTB
80 NEXTA
90 DATALENA,ANDY,IS GOING TO
100 DATACAMP BEAVER,CAMP FEE,HORSE RIDING,TENNIS L
    ESSONS,CRAFT SUPPLIES
110 DATAAURA,BILL,WILL ATTEND
120 DATASPORTS CLINIC,TUITION,UNIFORM FEE,EQUIPMEN
    T FEE,SPECIAL EVENTS
130 DATAMARY,JOHN,WILL STAY AT
140 DATALOGAN CANYON,CAMP FEE,T-SHIRTS,ACTIVITY FE
    E,SUPPLIES
150 PRINT"{CLR}"
160 I=INT(3*RND(0)+1)
170 PRINT"HERE ARE THE EXPENSES"
180 PRINT"FOR ONE WEEK AT":PRINTA$(I,1);"."
190 PRINT
200 C=1000*INT(5*RND(0)+1)+4000
210 GOSUB680
220 PRINTA$(I,2);TAB(15);C$
```

```
230 T=C
240 C=25*INT(12*RND(0)+1)+275
250 GOSUB680
260 PRINTA$(I,3);TAB(15);C$
270 T=T+C
280 C=50*INT(8*RND(0)+1)+150
290 GOSUB680
300 PRINTA$(I,4);TAB(15);C$
310 T=T+C
320 C=50*INT(5*RND(0)+1)+50
330 GOSUB680
340 PRINTA$(I,5);TAB(15);C$
350 T=T+C
360 PRINT
370 PRINT"TOTAL EXPENSES FOR":PRINT"ONE WEEK = ";
380 INPUT"$";A
390 PRINT
400 IFA=T/100THEN440
410 PRINT"ADD THE NUMBERS TO"
420 PRINT"GET TOTAL{SHIFT-SPACE}";T/100
430 PRINT
440 W=INT(7*RND(0)+2)
450 PRINTN$(INT(6*RND(0)+1));" ";P$(INT(3*RND(0)+1
    ))
460 PRINTA$(I,1):PRINT"FOR";W;"WEEKS."
470 PRINT"WHAT WILL IT COST?"
480 INPUT "$";A
490 PRINT
500 IF A=W*T/100 THEN 610
510 PRINT"MULTIPLY TOTAL EXPENSEPER WEEK TIMES";W;
    "WEEKS "
530 C=T*W
540 GOSUB680
550 PRINT"$";T/100;"*";W;"= ";C$
560 PRINT
570 PRINT"PRESS RETURN."
580 GET E$:IFE$=""THEN580
590 IFASC(E$)=13THEN150
600 GOTO580
610 PRINT"CORRECT!"
620 PRINT
630 PRINT"PRESS 1 TRY AGAIN"
640 PRINTTAB(6);"2 END PROGRAM"
650 GETE$:IFE$="1"THEN150
660 IFE$<>"2"THEN650
670 GOTO730
680 M$=STR$(C)
690 IFLEN(M$)=4 THEN M$="{SHIFT-SPACE}"+M$
700 L=LEN(M$)
710 C$="$"+LEFT$(M$,L-2)+"."+RIGHT$(M$,2)
```

```
720 RETURN
730 PRINT"{CLR}"
740 END
```

Program 6-11. Weekly Expenses

TRS-80 Color Computer; MC-10; TRS-80 Model I

```
20 DIM A$(3,5),N$(6),P$(3)
30 FOR A=1 TO 3
40 READ N$(A),N$(A+3),P$(A)
50 FOR B=1 TO 5
60 READ A$(A,B)
70 NEXT B
80 NEXT A
90 DATA LENA,ANDY,IS GOING TO
100 DATA CAMP BEAVER,CAMP FEE,HORSE RIDING,TENNIS
    LESSONS,CRAFT SUPPLIES
110 DATA AURA,BILL,WILL ATTEND
120 DATA SPORTS CLINIC,TUITION,UNIFORM FEE,EQUIPME
    NT FEE,SPECIAL EVENTS
130 DATA MARY,JOHN,WILL STAY AT
140 DATA LOGAN CANYON,CAMP FEE, T-SHIRTS,ACTIVITY
    FEE,SUPPLIES
150 CLS
160 I=RND(3)
170 PRINT "HERE ARE THE EXPENSES FOR"
180 PRINT "ONE WEEK AT ";A$(I,1);"."
190 PRINT
200 C=RND(5)*1000+4000
210 GOSUB 680
220 PRINT A$(I,2),C$
230 T=C
240 C=RND(12)*25+275
250 GOSUB 680
260 PRINT A$(I,3),C$
270 T=T+C
280 C=RND(8)*50+150
290 GOSUB 680
300 PRINT A$(I,4),C$
310 T=T+C
320 C=RND(5)*50+50
330 GOSUB 680
340 PRINT A$(I,5),C$
350 T=T+C
360 PRINT
370 PRINT "TOTAL EXPENSES FOR ONE WEEK ="
380 INPUT "$";A
390 PRINT
400 IF A=T/100 THEN 440
410 PRINT "ADD THE NUMBERS TO GET"
```

```
42Ø PRINT "TOTAL ";T/1ØØ
43Ø PRINT
44Ø W=RND(7)+1
45Ø PRINT N$(RND(6));" ";P$(RND(3))
46Ø PRINT A$(I,1);" FOR";W;"WEEKS."
47Ø PRINT "WHAT WILL IT COST?"
48Ø INPUT "$";A
49Ø PRINT
5ØØ IF A=W*T/1ØØ THEN 61Ø
51Ø PRINT "MULTIPLY TOTAL EXPENSES"
52Ø PRINT "PER WEEK TIMES";W;"WEEKS:"
53Ø C=T*W
54Ø GOSUB 68Ø
55Ø PRINT "$";T/1ØØ;"*";W;"= ";C$
56Ø PRINT
57Ø PRINT "PRESS ENTER."
58Ø E$=INKEY$:IF E$="" THEN 58Ø
59Ø IF ASC(E$)=13 THEN 15Ø
6ØØ GOTO 58Ø
61Ø PRINT "CORRECT!"
62Ø PRINT
63Ø PRINT "PRESS 1 ANOTHER PROBLEM"
64Ø PRINT TAB(6);"2 END PROGRAM"
65Ø E$=INKEY$:IF E$="1" THEN 15Ø
66Ø IF E$<>"2" THEN 65Ø
67Ø GOTO 73Ø
68Ø M$=STR$(C)
69Ø IF LEN(M$)=4 THEN M$=" "+M$
7ØØ L=LEN(M$)
71Ø C$="$"+LEFT$(M$,L-2)+"."+RIGHT$(M$,2)
72Ø RETURN
73Ø CLS
74Ø END
```

Program 6-12. Weekly Expenses

TI-99/4A

```
2Ø DIM A$(3,5),N$(6),P$(3)
3Ø FOR A=1 TO 3
4Ø READ N$(A),N$(A+3),P$(A)
5Ø FOR B=1 TO 5
6Ø READ A$(A,B)
7Ø NEXT B
8Ø NEXT A
9Ø DATA LENA,ANDY,IS GOING TO
1ØØ DATA CAMP BEAVER,CAMP FEE,HORSE RIDING,TENNIS,
    CRAFTS
11Ø DATA AURA,BILL,WILL ATTEND
12Ø DATA SPORTS CLINIC,TUITION,UNIFORM FEE,EQUIPME
    NT FEE,SPECIAL EVENT
```

```
13Ø DATA MARY,JOHN,WILL STAY AT
14Ø DATA LOGAN CANYON,CAMP FEE,T-SHIRTS,ACTIVITY F
    EE,SUPPLIES
15Ø CALL CLEAR
155 RANDOMIZE
16Ø I=INT(3*RND+1)
17Ø PRINT "HERE ARE THE EXPENSES FOR"
18Ø PRINT "ONE WEEK AT ";A$(I,1);"."
19Ø PRINT
2ØØ C=INT(5*RND+1)*1ØØØ+4ØØØ
21Ø GOSUB 68Ø
22Ø PRINT A$(I,2),C$
23Ø T=C
24Ø C=INT(12*RND+1)*25+275
25Ø GOSUB 68Ø
26Ø PRINT A$(I,3),C$
27Ø T=T+C
28Ø C=INT(8*RND+1)*5Ø+15Ø
29Ø GOSUB 68Ø
3ØØ PRINT A$(I,4),C$
31Ø T=T+C
32Ø C=INT(5*RND+1)*5Ø+5Ø
33Ø GOSUB 68Ø
34Ø PRINT A$(I,5),C$
35Ø T=T+C
36Ø PRINT
37Ø PRINT "TOTAL EXPENSES FOR ONE WEEK"
38Ø INPUT "$":A
39Ø PRINT
4ØØ IF A=T/1ØØ THEN 44Ø
41Ø PRINT "ADD THE NUMBERS TO GET"
42Ø PRINT "TOTAL ";T/1ØØ
43Ø PRINT
44Ø W=INT(7*RND+2)
45Ø PRINT N$(INT(6*RND+1));" ";P$(INT(3*RND+1))
46Ø PRINT A$(I,1);" FOR";W;"WEEKS."
47Ø PRINT "WHAT WILL IT COST?"
48Ø INPUT "$":A
49Ø PRINT
5ØØ IF A=W*T/1ØØ THEN 61Ø
51Ø PRINT "MULTIPLY TOTAL EXPENSES"
52Ø PRINT "PER WEEK TIMES";W;"WEEKS:"
53Ø C=T*W
54Ø GOSUB 68Ø
55Ø PRINT "$";T/1ØØ;"*";W;"= ";C$
56Ø PRINT
57Ø PRINT "PRESS ENTER."
58Ø CALL KEY(Ø,K,S)
59Ø IF K=13 THEN 15Ø
```

```
6ØØ GOTO 58Ø
61Ø PRINT "CORRECT!"
62Ø PRINT
63Ø PRINT "PRESS 1 ANOTHER PROBLEM"
64Ø PRINT TAB(7);"2 END PROGRAM"
65Ø CALL KEY(Ø,K,S)
66Ø IF K=49 THEN 15Ø
67Ø IF K=5Ø THEN 73Ø ELSE 65Ø
68Ø M$=STR$(C)
69Ø IF LEN(M$)>3 THEN 7ØØ
695 M$=" "&M$
7ØØ L=LEN(M$)
71Ø C$="$"&SEG$(M$,1,L-2)&"."&SEG$(M$,L-1,2)
72Ø RETURN
73Ø CALL CLEAR
74Ø END
```

Saving Money

Computers: VIC-20; TRS-80 Color Computer; MC-10; TRS-80 Model I;
TI-99/4A

"Saving Money" gives you a chance to brush up on your division skills. The program first chooses one of six names to use in the problem. One of three major items, with its price, is selected. The computer also randomly selects a number to represent the number of weeks you'll have to save for this item. You have to figure out how much money you'd need to save each week to purchase the item.

The price of the item is determined by a base value multiplied by a random number in a range from 1 to F. The base value, multiplier, and value F are all READ from DATA statements in the program.

Here's how the program works:

Line VIC-20	**Line** Color Computer and TI-99/4A	**Function**
30	120	Clear screen.
40–110	130–200	READ from DATA six names N$, the item A$ with a base price B, a multiplier M, and random factor F.
120	210	Clear screen.
130–140	220–230	Randomly PRINT name.
150–160	240–250	Randomly choose one of three items.
170–190	260–280	Calculate and PRINT cost of item.
200–270	290–360	Randomly choose number of weeks and PRINT problem.

280	370	Calculate correct answer.
290	380	Receive answer.
300–370	390–460	If answer is incorrect, show how to get right answer, and return for another problem.
380–400	470–490	PRINT message for correct answer.
410–460	500–550	PRINT option to try again and branch appropriately.
470–510	560–600	Convert cost into dollars.
520–530	610	END.

Program 6-13. Saving Money

VIC-20

For Commodore 64 version, add line 25: 25 POKE 53281,1

```
3Ø PRINT"{CLR}"
4Ø FOR C=1 TO 6
5Ø READ N$(C)
6Ø NEXT C
7Ø FOR C=1 TO 3
8Ø READ A$(C),B(C),M(C),F(C)
9Ø NEXT C
1ØØ DATA SALLY,HEIDI,NANCY,BRENT,DAVID,CHUCK
11Ø DATA BIKE,8Ø,5,7,STEREO,9Ø,5,14,COMPUTER,1ØØ,1
    Ø,1Ø
12Ø PRINT"{CLR}{BLU}"
13Ø R6=INT(6*RND(Ø)+1)
14Ø PRINT N$(R6);" WANTS TO BUY A"
15Ø R3=INT(3*RND(Ø)+1)
16Ø PRINT A$(R3);"."
17Ø T=B(3)+M(R3)*INT(F(R3)*RND(Ø)+1)
18Ø GOSUB 48Ø
19Ø PRINT"IT WILL COST ";C$;"."
2ØØ P$="HE"
21Ø IF R6>=4 THEN 23Ø
22Ø P$="SHE"
23Ø W=1Ø*INT(4*RND(Ø)+1)
24Ø PRINT"IF ";P$;" SAVES FOR"
245 PRINTW;"WEEKS,"
25Ø PRINT"HOW MUCH WILL ";N$(R6)"
26Ø PRINT"NEED TO SAVE EACH"
265 PRINT"WEEK?"
27Ø PRINT"{BLK}"
28Ø S=(T/W)/1ØØ
29Ø INPUT ANS
3ØØ IF ABS(ANS-S)<.Ø1 THEN 38Ø
31Ø PRINT"{RED}TOTAL COST ";C$
32Ø PRINT"DIVIDED BY";W;"WKS ="
```

```
330 GOSUB 47Ø
34Ø PRINTC$
35Ø PRINT"{GRN}"
36Ø INPUT "PRESS <RETURN>";E$
37Ø GOTO 12Ø
38Ø PRINT
39Ø PRINT"{RED}CORRECT!"
4ØØ PRINT
41Ø PRINT"{BLK}ANOTHER PROBLEM?"
42Ø PRINT"(Y/N)"
43Ø GET E$
44Ø IF E$="Y" THEN 12Ø
45Ø IF{2 SPACES}E$="N" THEN 52Ø
46Ø GOTO 43Ø
47Ø T=S+.ØØ5
48Ø T=INT(1ØØ*T)
49Ø T$=STR$(T)
5ØØ C$="$"+LEFT$(T$,LEN(T$)-2)+"."+RIGHT$(T$,2)
51Ø RETURN
52Ø PRINT"{CLR}{BLU}"
53Ø END
```

Program 6-14. Saving Money

TRS-80 Color Computer; MC-10; TRS-80 Model I

```
12Ø CLS
13Ø FOR C=1 TO 6
14Ø READ N$(C)
15Ø NEXT C
16Ø FOR C=1 TO 3
17Ø READ A$(C),B(C),M(C),F(C)
18Ø NEXT C
19Ø DATA SALLY,HEIDI,NANCY,BRENT,DAVID,CHUCK
2ØØ DATA BIKE,8Ø,5,7,STEREO,9Ø,5,14,COMPUTER,1ØØ,1
    Ø,1Ø
21Ø CLS
22Ø R6=RND(6)
23Ø PRINT N$(R6);" WANTS TO BUY A"
24Ø R3=RND(3)
25Ø PRINT A$(R3);"."
26Ø T=B(R3)+M(R3)*RND(F(R3))
27Ø GOSUB 57Ø
28Ø PRINT "IT WILL COST ";C$;"."
29Ø P$="HE"
3ØØ IF R6>=4 THEN 32Ø
31Ø P$="SHE"
32Ø W=1Ø*RND(4)
33Ø PRINT "IF ";P$;" SAVES FOR";W;"WEEKS,"
34Ø PRINT "HOW MUCH WILL ";N$(R6);" NEED"
35Ø PRINT "TO SAVE EACH WEEK?"
```

```
36Ø PRINT
37Ø S=(T/W)/1ØØ
38Ø INPUT ANS
39Ø IF ABS(ANS-S)<.Ø1 THEN 47Ø
4ØØ PRINT "TOTAL COST ";C$
41Ø PRINT "DIVIDED BY";W;"WEEKS ="
42Ø GOSUB 56Ø
43Ø PRINT C$
44Ø PRINT
45Ø INPUT "PRESS <ENTER>";E$
46Ø GOTO 21Ø
47Ø PRINT
48Ø PRINT "CORRECT!"
49Ø PRINT
5ØØ PRINT "ANOTHER PROBLEM?"
51Ø PRINT "(Y/N)"
52Ø E$=INKEY$
53Ø IF E$="Y" THEN 21Ø
54Ø IF E$="N" THEN 61Ø
55Ø GOTO 52Ø
56Ø T=S
57Ø T=INT(1ØØ*T)
58Ø T$=STR$(T)
59Ø C$="$"+LEFT$(T$,LEN(T$)-2)+"."+RIGHT$(T$,2)
6ØØ RETURN
61Ø END
```

Program 6-15. Saving Money

TI-99/4A

```
12Ø CALL CLEAR
13Ø FOR C=1 TO 6
14Ø READ N$(C)
15Ø NEXT C
16Ø FOR C=1 TO 3
17Ø READ A$(C),B(C),M(C),F(C)
18Ø NEXT C
19Ø DATA SALLY,HEIDI,NANCY,BRENT,DAVID,CHUCK
2ØØ DATA BIKE,8Ø,5,7,STEREO,9Ø,5,14,COMPUTER,1ØØ,1
    Ø,1Ø
21Ø CALL CLEAR
215 RANDOMIZE
22Ø R6=INT(6*RND+1)
23Ø PRINT N$(R6);" WANTS TO BUY A"
24Ø R3=INT(3*RND+1)
25Ø PRINT A$(R3);"."
26Ø T=B(R3)+M(R3)*INT(F(R3)*RND+1)
27Ø GOSUB 57Ø
28Ø PRINT "IT WILL COST ";C$;"."
```

```
29Ø P$="HE"
3ØØ IF R6>=4 THEN 32Ø
31Ø P$="SHE"
32Ø W=1Ø*INT(4*RND+1)
33Ø PRINT "IF ";P$;" SAVES FOR";W;"WEEKS,"
34Ø PRINT "HOW MUCH WILL ";N$(R6);" NEED"
35Ø PRINT "TO SAVE EACH WEEK?"
36Ø PRINT
37Ø S=(T/W)/1ØØ
38Ø INPUT ANS
39Ø IF ABS(ANS-S)<.Ø1 THEN 47Ø
4ØØ PRINT "TOTAL COST ";C$
41Ø PRINT "DIVIDED BY";W;"WEEKS ="
42Ø GOSUB 56Ø
43Ø PRINT C$
44Ø PRINT
45Ø INPUT "PRESS <ENTER>":E$
46Ø GOTO 21Ø
47Ø PRINT
48Ø PRINT "CORRECT!"
49Ø PRINT
5ØØ PRINT "ANOTHER PROBLEM?"
51Ø PRINT "(Y/N)"
52Ø CALL KEY(Ø,K,S)
53Ø IF K=89 THEN 21Ø
54Ø IF K=78 THEN 61Ø
55Ø GOTO 52Ø
56Ø T=S
57Ø T=INT(1ØØ*T)
58Ø T$=STR$(T)
59Ø C$="$"&SEG$(T$,1,LEN(T$)-2)&"."&SEG$(T$,LEN(T$
    )-1,2)
6ØØ RETURN
61Ø END
```

Chapter 7
Homework Helpers

Chapter 7
Homework Helpers

Computers work best when they do repetitive tasks. Checking your homework is just that — it's repetitive, time-consuming, and often boring. The programs in this chapter, which have all been adapted for use on one of three computers, are designed to quickly check your homework.

You should do your problems on paper, as usual, showing all the steps in getting an answer, and then use these programs to check your results. These programs, because they are quite simple, show only an answer, not the step-by-step process of solving the problem.

After each problem, you can press 1 for another problem, or 2 to end the program.

Division

Computers: VIC-20; TRS-80 Color Computer; MC-10; TRS-80 Model I; TI-99/4A

Most division problems can be checked and corrected with a calculator. However, if there is a remainder, a calculator converts it to a decimal equivalent. When you first learn division, you learn to divide and specify a remainder if the number does not divide evenly. "Division" keeps the answer in quotient plus remainder form. When you enter the divisor and dividend, the quotient and remainder are printed.

The INTeger function is used in these programs to calculate the whole number part of the quotient. The remainder is then the difference between the original number and the integer portion, multiplied by the divisor.

Program 7-1. Division

VIC-20

```
11Ø PRINT"{CLR}"
12Ø PRINT"DIVISION--REMAINDER"
13Ø PRINT
14Ø PRINT
15Ø PRINTTAB(11);"QUOTIENT"
16Ø PRINTTAB(9);"----------"
17Ø PRINT" DIVISOR ) DIVIDEND"
18Ø PRINT:PRINT
19Ø INPUT "DIVISOR:{2 SPACES}";D
2ØØ IF D<>Ø THEN 23Ø
21Ø PRINT"DIVISOR CANNOT = Ø"
22Ø GOTO 19Ø
23Ø INPUT "DIVIDEND: ";N
24Ø Q=INT(N/D)
```

```
250 R=N-Q*D
260 PRINT
270 PRINT"QUOTIENT = ";Q;" R";R
280 PRINT
290 PRINT"CHOOSE:"
300 PRINT" 1 ANOTHER PROBLEM"
310 PRINT" 2 END PROGRAM"
320 GET A$
330 IF A$="1"THEN 110
340 IF A$<>"2"THEN 320
350 END
```

Program 7-2. Division

TRS-80 Color Computer; MC-10; TRS-80 Model I

```
110 CLS
120 PRINT"DIVISION--REMAINDER"
130 PRINT
140 PRINT
150 PRINTTAB(11);"QUOTIENT"
160 PRINTTAB(9);"----------"
170 PRINT" DIVISOR ) DIVIDEND"
180 PRINT
190 INPUT "DIVISOR:  ";D
200 IF D<>0 THEN 230
210 PRINT"DIVISOR CANNOT = 0"
220 GOTO 190
230 INPUT "DIVIDEND: ";N
240 Q=INT(N/D)
250 R=N-Q*D
260 PRINT
270 PRINT"QUOTIENT = ";Q;" R";R
280 PRINT
290 PRINT"CHOOSE:"
300 PRINT" 1 ANOTHER PROBLEM"
310 PRINT" 2 END PROGRAM"
320 A$=INKEY$
330 IF A$="1"THEN 110
340 IF A$<>"2"THEN 320
350 END
```

Program 7-3. Divison

TI-99/4A

```
110 CALL CHAR(37,"8040202020408")
120 CALL CHAR(38,"00000000000000FF")
130 CALL CLEAR
140 PRINT "DIVISION WITH REMAINDER"::::
150 PRINT TAB(11);"QUOTIENT"
160 PRINT TAB(10);"&&&&&&&&&&"
```

```
170 PRINT " DIVISOR %DIVIDEND":::
180 INPUT "DIVISOR:  ":I
190 IF I<>0 THEN 220
200 PRINT :"SORRY, DIVISOR CANNOT = 0"::
210 GOTO 180
220 INPUT "DIVIDEND: ":D
230 Q=INT(D/I)
240 R=D-Q*I
250 PRINT :"QUOTIENT =";Q;" R";R
260 PRINT :::"CHOOSE:"
270 PRINT " 1 ANOTHER PROBLEM"
280 PRINT " 2 END PROGRAM"
290 CALL KEY(0,K,S)
300 IF K=49 THEN 130
310 IF K<>50 THEN 290
320 CALL CLEAR
330 END
```

All Factors

Computers: VIC-20; TRS-80 Color Computer; MC-10; TRS-80 Model I;
TI-99/4A

This program will find all the factors of a number that you enter. (Factors are all the possible divisors that will yield an integer answer.) For example, the number 12 has the factors 12, 6, 4, 3, 2, and 1. You must enter a number greater than 1. Large numbers will take longer for the computer to factor.

Program 7-4. Find All Factors

VIC-20

```
110 PRINT"{CLR}"
120 PRINT"FINDING ALL FACTORS"
130 PRINT
140 PRINT"ENTER NUMBER TO BE"
150 INPUT "FACTORED: ";N
160 IF N>1 THEN 190
170 PRINT"ENTER NUMBER > 1"
180 GOTO130
190 PRINT"FACTORS OF";N;"ARE"
200 PRINT N;
210 B=INT(N/2+1)
220 FOR C=2 TO B
230 IF N/C<>INT(N/C) THEN 280
240 B=N/C
250 PRINT B;
260 IF B=1 THEN 300
270 IF B=2 THEN 290
280 NEXT C
```

```
290 PRINT " 1"
300 PRINT
310 PRINT"CHOOSE:"
320 PRINT" 1 ANOTHER PROBLEM"
330 PRINT" 2 END PROGRAM"
340 GETA$
350 IF A$="1" THEN 110
360 IF A$<>"2" THEN 340
370 PRINT"{CLR}":END
```

Program 7-5. Find All Factors

TRS-80 Color Computer; MC-10; TRS-80 Model I

```
110 CLS
120 PRINT"FINDING ALL FACTORS"
130 PRINT
140 PRINT"ENTER NUMBER TO BE"
150 INPUT "FACTORED: ";N
160 IF N>1 THEN 190
170 PRINT"ENTER NUMBER > 1"
180 GOTO130
190 PRINT"FACTORS OF";N;"ARE"
200 PRINT N;
210 B=INT(N/2+1)
220 FOR C=2 TO B
230 IF N/C<>INT(N/C) THEN 280
240 B=N/C
250 PRINT B;
260 IF B=1 THEN 300
270 IF B=2 THEN 290
280 NEXT C
290 PRINT " 1"
300 PRINT
310 PRINT "CHOOSE:"
320 PRINT " 1 ANOTHER PROBLEM"
330 PRINT " 2 END PROGRAM"
340 A$=INKEY$
350 IF A$="1" THEN 110
360 IF A$<>"2" THEN 340
370 CLS:END
```

Program 7-6. Find All Factors

TI-99/4A

```
110 CALL CLEAR
120 PRINT "FINDING ALL FACTORS"
130 PRINT
140 PRINT "ENTER NUMBER TO BE"
150 INPUT "FACTORED: ":N
160 IF N>1 THEN 190
```

```
170 PRINT "ENTER NUMBER > 1"
180 GOTO 130
190 PRINT :"FACTORS OF";N;"ARE"
200 PRINT N;
210 B=INT(N/2+1)
220 FOR C=2 TO B
230 IF N/C<>INT(N/C)THEN 280
240 B=N/C
250 PRINT B;
260 IF B=1 THEN 300
270 IF B=2 THEN 290
280 NEXT C
290 PRINT " 1"
300 PRINT
310 PRINT "CHOOSE:"
320 PRINT " 1 ANOTHER PROBLEM"
330 PRINT " 2 END PROGRAM"
340 CALL KEY(0,K,S)
350 IF K=49 THEN 110
360 IF K<>50 THEN 340
370 CALL CLEAR
380 END
```

Prime Factors

Computers: VIC-20; TRS-80 Color Computer; MC-10; TRS-80 Model I; TI-99/4A

"Prime Factors" returns a list of the prime factors for a number you enter. Another term for this process is *complete factorization*. All prime factors multiplied together will yield the original number. For example, the prime factors of the number 12 are 2, 2, and 3. You must enter a number greater than 1.

Program 7-7. Prime Factors

VIC-20

```
110 PRINT"{CLR}"
120 PRINT"PRIME FACTORS OR"
130 PRINT"COMPLETE FACTORIZATION"
140 PRINT
150 PRINT"ENTER NUMBER TO BE"
160 INPUT "FACTORED: ";N
170 IF N>1 THEN 200
180 PRINT"NUMBER MUST BE > 1"
190 GOTO 140
200 PRINT
210 PRINT"THE PRIME FACTORS ARE:"
220 G=INT(N/2)
230 FOR I=2 TO G
240 IF N/I<>INT(N/I) THEN 290
```

```
250 N=N/I
260 G=N
270 PRINT I;
280 GOTO 230
290 NEXT I
300 IF N=1 THEN 320
310 PRINT N
320 PRINT
330 PRINT
340 PRINT"CHOOSE:"
350 PRINT" 1 ANOTHER PROBLEM"
360 PRINT" 2 END PROGRAM"
370 GET A$
380 IF A$="1" THEN 110
390 IF A$<>"2" THEN 370
400 PRINT"{CLR}":END
```

Program 7-8. Prime Factors

TRS-80 Color Computer; MC-10; TRS-80 Model I

```
110 CLS
120 PRINT"PRIME FACTORS OR"
130 PRINT"COMPLETE FACTORIZATION"
140 PRINT
150 PRINT"ENTER NUMBER TO BF"
160 INPUT "FACTORED: ";N
170 IF N>1 THEN 200
180 PRINT"NUMBER MUST BE > 1"
190 GOTO 140
200 PRINT
210 PRINT"THE PRIME FACTORS ARE:"
220 G=INT(N/2)
230 FOR I=2 TO G
240 IF N/I<>INT(N/I) THEN 290
250 N=N/I
260 G=N
270 PRINT I;
280 GOTO 230
290 NEXT I
300 IF N=1 THEN 320
310 PRINT N
320 PRINT
330 PRINT
340 PRINT"CHOOSE:"
350 PRINT" 1 ANOTHER PROBLEM"
360 PRINT" 2 END PROGRAM"
370 A$=INKEY$
380 IF A$="1" THEN 110
390 IF A$<>"2" THEN 370
400 CLS:END
```

Program 7-9. Prime Factors

TI-99/4A

```
110 CALL CLEAR
120 PRINT "PRIME FACTORS OR"
130 PRINT "COMPLETE FACTORIZATION"
140 PRINT
150 PRINT "ENTER NUMBER TO BE"
160 INPUT "FACTORED: ":N
170 IF N>1 THEN 200
180 PRINT "NUMBER MUST BE > 1"
190 GOTO 140
200 PRINT
210 PRINT "THE PRIME FACTORS ARE:"
220 G=INT(N/2)
230 FOR I=2 TO G
240 IF N/I<>INT(N/I)THEN 290
250 N=N/I
260 G=N
270 PRINT I;
280 GOTO 230
290 NEXT I
300 IF N=1 THEN 320
310 PRINT N
320 PRINT
330 PRINT
340 PRINT "CHOOSE:"
350 PRINT " 1 ANOTHER PROBLEM"
360 PRINT " 2 END PROGRAM"
370 CALL KEY(0,K,S)
380 IF K=49 THEN 110
390 IF K<>50 THEN 370
400 CALL CLEAR
410 END
```

Greatest Common Factor

Computers: VIC-20; TRS-80 Color Computer; MC-10; TRS-80 Model I; TI-99/4A

This program will find the greatest common factor (GCF) of two numbers. This mathematical concept is usually introduced before the concept of simplifying fractions. For example, if the two numbers entered are 12 and 18, the greatest common factor is 6. Both numbers can be evenly divided by 6. Although they may also both be divided by 3 or 2, 6 is the largest factor. You enter two numbers, and the computer will show the greatest common factor.

Program 7-10. Greatest Common Factor

VIC-20

```
11Ø PRINT"{CLR}"
12Ø PRINT"GREATEST COMMON FACTOR"
13Ø PRINT"OF TWO NUMBERS"
14Ø PRINT
15Ø INPUT "FIRST NUMBER ";M
16Ø IF M>1 THEN 19Ø
17Ø PRINT"ENTER NUMBER > 1"
18Ø GOTO 14Ø
19Ø IF{2 SPACES}M<1ØØØØ THEN 22Ø
2ØØ PRINT"MUST BE < 1ØØØØ"
21Ø GOTO14Ø
22Ø PRINT
23Ø INPUT "SECOND NUMBER ";N
24Ø IF N>1 THEN 27Ø
25Ø PRINT"ENTER NUMBER > 1"
26Ø GOTO 22Ø
27Ø IF N<1ØØØØ THEN 3ØØ
28Ø PRINT"MUST BE < 1ØØØØ"
29Ø GOTO 22Ø
3ØØ PRINT
31Ø PRINT"GREATEST COMMON FACTOR ={SHIFT-SPACE}";
32Ø IF M=N THEN G=M:GOTO 45Ø
33Ø IF{2 SPACES}M<N THEN 37Ø
34Ø MM=M
35Ø M=N
36Ø N=MM
37Ø FOR I=1 TO M
38Ø IF (M/I)<>INT(M/I)THEN 43Ø
39Ø J=M/I
4ØØ IF (N/J)<>INT(N/J)THEN 43Ø
41Ø G=J
42Ø GOTO 45Ø
43Ø NEXT I
44Ø G=1
45Ø PRINT G
46Ø PRINT
47Ø PRINT"CHOOSE:"
48Ø PRINT" 1 ANOTHER PROBLEM"
49Ø PRINT" 2 END PROGRAM"
5ØØ GET A$
51Ø IF A$="1" THEN 11Ø
52Ø IF A$<>"2" THEN 5ØØ
53Ø PRINT"{CLR}":END
```

Program 7-11. Greatest Common Factor

TRS-80 Color Computer; MC-10; TRS-80 Model I

```
110 CLS
120 PRINT"GREATEST COMMON FACTOR"
130 PRINT"OF TWO NUMBERS"
140 PRINT
150 INPUT "FIRST NUMBER ";M
160 IF M>1 THEN 190
170 PRINT"ENTER NUMBER > 1"
180 GOTO 140
190 IF M<10000 THEN 220
200 PRINT"MUST BE < 10000"
210 GOTO 140
220 PRINT
230 INPUT "SECOND NUMBER ";N
240 IF N>1 THEN 270
250 PRINT"ENTER NUMBER > 1"
260 GOTO 220
270 IF N<10000 THEN 300
280 PRINT"MUST BE < 10000"
290 GOTO 220
300 PRINT
310 PRINT"GREATEST COMMON FACTOR ="
320 IF M=N THEN G=M:GOTO 450
330 IF M<N THEN 370
340 MM=M
350 M=N
360 N=MM
370 FOR I=1 TO M
380 IF (M/I)<>INT(M/I)THEN 430
390 J=M/I
400 IF (N/J)<>INT(N/J)THEN 430
410 G=J
420 GOTO 450
430 NEXT I
440 G=1
450 PRINT G
460 PRINT
470 PRINT"CHOOSE:"
480 PRINT" 1 ANOTHER PROBLEM"
490 PRINT" 2 END PROGRAM"
500 A$=INKEY$
510 IF A$="1" THEN 110
520 IF A$<>"2" THEN 500
530 CLS:END
```

Program 7-12. Greatest Common Factor

TI-99/4A

```
11Ø CALL CLEAR
12Ø PRINT "GREATEST COMMON FACTOR"
13Ø PRINT "OF TWO NUMBERS"
14Ø PRINT
15Ø INPUT "FIRST NUMBER ":M
16Ø IF M>1 THEN 19Ø
17Ø PRINT "ENTER NUMBER > 1"
18Ø GOTO 14Ø
19Ø IF M<1ØØØØ THEN 22Ø
2ØØ PRINT "MUST BE < 1ØØØØ"
21Ø GOTO 14Ø
22Ø PRINT
23Ø INPUT "SECOND NUMBER ":N
24Ø IF N>1 THEN 27Ø
25Ø PRINT "ENTER NUMBER > 1"
26Ø GOTO 22Ø
27Ø IF N<1ØØØØ THEN 3ØØ
28Ø PRINT "MUST BE < 1ØØØØ"
29Ø GOTO 22Ø
3ØØ PRINT
31Ø PRINT "GREATEST COMMON FACTOR ="
32Ø IF M<>N THEN 35Ø
33Ø G=M
34Ø GOTO 47Ø
35Ø IF M<N THEN 39Ø
36Ø MM=M
37Ø M=N
38Ø N=MM
39Ø FOR I=1 TO M
4ØØ IF (M/I)<>INT(M/I)THEN 45Ø
41Ø J=M/I
42Ø IF (N/J)<>INT(N/J)THEN 45Ø
43Ø G=J
44Ø GOTO 47Ø
45Ø NEXT I
46Ø G=1
47Ø PRINT G
48Ø PRINT
49Ø PRINT "CHOOSE:"
5ØØ PRINT " 1 ANOTHER PROBLEM"
51Ø PRINT " 2 END PROGRAM"
52Ø CALL KEY(Ø,K,S)
53Ø IF K=49 THEN 11Ø
54Ø IF K<>5Ø THEN 52Ø
55Ø CALL CLEAR
56Ø END
```

Least Common Multiple

Computers: VIC-20; TRS-80 Color Computer; MC-10; TRS-80 Model I; TI-99/4A

Use this program to find the least common multiple of two or three numbers. First, enter 2 or 3 for the number of given numbers. Next enter the numbers. The program will return the least common multiple.

For example, enter three numbers, such as 12, 18, and 6. The least common multiple is 36. 36 may be divided evenly by 12, 18, and 6, and is the smallest number that can be divided by all three.

This concept is usually introduced to students before they learn about adding and subtracting fractions with unlike denominators. Another name for this concept is *smallest common denominator.*

Program 7-13. Least Common Multiple

VIC-20

```
110 PRINT"{CLR}"
120 PRINT"LEAST COMMON MULTIPLE"
130 PRINT"OF 2 OR 3 NUMBERS"
140 PRINT
150 PRINT"HOW MANY NUMBERS--2,3?"
160 GET A$:IF A$="" THEN 160
170 IF ASC(A$)<50 THEN 160
180 IF ASC(A$)>51 THEN 160
190 PRINT A$;" NUMBERS"
200 FOR C=1 TO VAL(A$)
210 PRINT
220 PRINT"NUMBER";C;
230 INPUT N(C)
240 IF N(C)>1 THEN 270
250 PRINT"NUMBER MUST BE > 1"
260 GOTO 210
270 IF N(C)<1000 THEN 300
280 PRINT"NUMBER MUST BE < 1000"
290 GOTO 210
300 NEXT C
310 IF A$="3" THEN 440
320 IF N(1)<>N(2) THEN 350
330 L=N(1)
340 GOTO 630
350 IF N(1)<N(2) THEN 390
360 NN=N(1)
370 N(1)=N(2)
380 N(2)=NN
390 FOR C=1 TO N(1)
```

```
4ØØ IF C*N(2)/N(1)=INT(C*N(2)/N(1)) THEN L=C*N(2):
    GOTO 63Ø
41Ø NEXT C
42Ø L=N(1)*N(2)
43Ø GOTO 63Ø
44Ø S=Ø
45Ø FOR C=1 TO 2
46Ø IF N(C)<=N(C+1) THEN 51Ø
47Ø NN=N(C)
48Ø N(C)=N(C+1)
49Ø N(C+1)=NN
5ØØ S=1
51Ø NEXT C
52Ø IF S=1 THEN 44Ø
53Ø FOR C=1 TO N(2)
54Ø F=C*N(3)
55Ø IF (F/N(1)=INT(F/N(1))) AND (F/N(2)=INT(F/N(2)
    )) THEN L=F:GOTO 63Ø
56Ø NEXT C
57Ø M=N(2)*N(3)
58Ø FOR C=1 TO N(1)
59Ø F=C*M
6ØØ IF F/N(1)=INT(F/N(1)) THEN L=F:GOTO 63Ø
61Ø NEXT C
62Ø L=M*N(1)
63Ø PRINT
64Ø PRINT"LEAST COMMON MULTIPLE IS ";
65Ø PRINTL
66Ø PRINT
67Ø PRINT"CHOOSE:"
68Ø PRINT" 1 ANOTHER PROBLEM"
69Ø PRINT" 2 END PROGRAM"
7ØØ GET A$
71Ø IF A$="1" THEN 11Ø
72Ø IF A$<>"2" THEN 7ØØ
73Ø PRINT"{CLR}":END
```

Program 7-14. Least Common Multiple

TRS-80 Color Computer; MC-10; TRS-80 Model I

```
11Ø CLS
12Ø PRINT"LEAST COMMON MULTIPLE"
13Ø PRINT"OF 2 OR 3 GIVEN NUMBERS"
14Ø PRINT
15Ø PRINT"HOW MANY NUMBERS--2 OR 3?"
16Ø A$=INKEY$:IF A$="" THEN 16Ø
17Ø IF ASC(A$)<5Ø THEN 16Ø
18Ø IF ASC(A$)>51 THEN 16Ø
19Ø PRINT A$;" NUMBERS"
```

```
200 FOR C=1 TO VAL(A$)
210 PRINT
220 PRINT"NUMBER";C;
230 INPUT N(C)
240 IF N(C)>1 THEN 270
250 PRINT"NUMBER MUST BE > 1"
260 GOTO 210
270 IF N(C)<1000 THEN 300
280 PRINT"NUMBER MUST BE < 1000"
290 GOTO 210
300 NEXT C
310 IF A$="3" THEN 440
320 IF N(1)<>N(2) THEN 350
330 L=N(1)
340 GOTO 630
350 IF N(1)<N(2) THEN 390
360 NN=N(1)
370 N(1)=N(2)
380 N(2)=NN
390 FOR C=1 TO N(1)
400 IF C*N(2)/N(1)=INT(C*N(2)/N(1)) THEN L=C*N(2):
    GOTO 630
410 NEXT C
420 L=N(1)*N(2)
430 GOTO 630
440 S=0
450 FOR C=1 TO 2
460 IF N(C)<=N(C+1) THEN 510
470 NN=N(C)
480 N(C)=N(C+1)
490 N(C+1)=NN
500 S=1
510 NEXT C
520 IF S=1 THEN 440
530 FOR C=1 TO N(2)
540 F=C*N(3)
550 IF (F/N(1)=INT(F/N(1))) AND (F/N(2)=INT(F/N(2)
     )) THEN L=F:GOTO 630
560 NEXT C
570 M=N(2)*N(3)
580 FOR C=1 TO N(1)
590 F=C*M
600 IF F/N(1)=INT(F/N(1)) THEN L=F:GOTO 630
610 NEXT C
620 L=M*N(1)
630 PRINT
640 PRINT"LEAST COMMON MULTIPLE IS"
650 PRINTL
660 PRINT
```

```
67Ø PRINT"CHOOSE:"
68Ø PRINT" 1 ANOTHER PROBLEM"
69Ø PRINT" 2 END PROGRAM"
7ØØ A$=INKEY$
71Ø IF A$="1" THEN 11Ø
72Ø IF A$<>"2" THEN 7ØØ
73Ø CLS:END
```

Program 7-15. Least Common Multiple

TI-99/4A

```
11Ø CALL CLEAR
12Ø PRINT "LEAST COMMON MULTIPLE"
13Ø PRINT "OF 2 OR 3 GIVEN NUMBERS"
14Ø PRINT
15Ø PRINT "HOW MANY NUMBERS --2 OR 3?"
16Ø CALL KEY(Ø,K,S)
17Ø IF K<5Ø THEN 16Ø
18Ø IF K>51 THEN 16Ø
19Ø PRINT K-48;" NUMBERS"
2ØØ FOR C=1 TO K-48
21Ø PRINT
22Ø PRINT "NUMBER";C;
23Ø INPUT N(C)
24Ø IF N(C)>1 THEN 27Ø
25Ø PRINT "NUMBER MUST BE > 1"
26Ø GOTO 21Ø
27Ø IF N(C)<1ØØØ THEN 3ØØ
28Ø PRINT "NUMBER MUST BE < 1ØØØ"
29Ø GOTO 21Ø
3ØØ NEXT C
31Ø IF K=51 THEN 46Ø
32Ø IF N(1)<>N(2)THEN 35Ø
33Ø L=N(1)
34Ø GOTO 69Ø
35Ø IF N(1)<N(2)THEN 39Ø
36Ø NN=N(1)
37Ø N(1)=N(2)
38Ø N(2)=NN
39Ø FOR C=1 TO N(1)
4ØØ IF C*N(2)/N(1)<>INT(C*N(2)/N(1))THEN 43Ø
41Ø L=C*N(2)
42Ø GOTO 69Ø
43Ø NEXT C
44Ø L=N(1)*N(2)
45Ø GOTO 69Ø
46Ø SW=Ø
47Ø FOR C=1 TO 2
48Ø IF N(C)<=N(C+1)THEN 53Ø
49Ø NN=N(C)
```

```
500 N(C)=N(C+1)
510 N(C+1)=NN
520 SW=1
530 NEXT C
540 IF SW=1 THEN 460
550 FOR C=1 TO N(2)
560 F=C*N(3)
570 IF (F/N(1)=INT(F/N(1)))+(F/N(2)=INT(F/N(2)))<>
    -2 THEN 600
580 L=F
590 GOTO 690
600 NEXT C
610 M=N(2)*N(3)
620 FOR C=1 TO N(1)
630 F=C*M
640 IF F/N(1)<>INT(F/N(1))THEN 670
650 L=F
660 GOTO 690
670 NEXT C
680 L=M*N(1)
690 PRINT ::"LEAST COMMON MULTIPLE IS"
700 PRINT L
710 PRINT ::"CHOOSE:"
720 PRINT " 1 ANOTHER PROBLEM"
730 PRINT " 2 END PROGRAM"
740 CALL KEY(0,K,S)
750 IF K=49 THEN 110
760 IF K<>50 THEN 740
770 CALL CLEAR
780 END
```

Equivalent Fractions

Computers: VIC-20; TRS-80 Color Computer; MC-10; TRS-80 Model I;
TI-99/4A

"Equivalent Fractions" can quickly find the unknown in problems such as 1/2=?/8. The fractions are of the form: A/B=C/D. You first press the letter for the unknown, then enter values for the other three numbers as the computer prompts you. The equivalent fractions will be printed.

Program 7-16. Equivalent Fractions

VIC-20

```
110 PRINT"{CLR}"
120 PRINT TAB(7);"A{5 SPACES}C"
130 PRINT TAB(7);"-{2 SPACES}={2 SPACES}-"
140 PRINT TAB(7);"B{5 SPACES}D"
150 PRINT:PRINT
```

```
160 PRINT"WHICH IS THE UNKNOWN?"
170 PRINT"CHOOSE A, B, C, OR D"
180 GETA$:IF A$=""THEN 180
190 K=ASC(A$)
200 IF K<65 THEN 180
210 IF K>68 THEN 180
220 PRINT
230 ON K-64 GOTO 240,350,490,600
240 INPUT "ENTER B =";B
250 IF B<>0 THEN 280
260 PRINT"B CANNOT = 0{DOWN}"
270 GOTO 240
280 INPUT "ENTER C =";C
290 INPUT "ENTER D =";D
300 IF D<>0 THEN 330
310 PRINT"D CANNOT = 0{DOWN}"
320 GOTO 290
330 A=INT(100*(B*C/D+.005))/100
340 GOTO 730
350 INPUT "ENTER A =";A
360 IF A<>0 THEN 390
370 PRINT"SORRY, A<>0{DOWN}"
380 GOTO 350
390 INPUT "ENTER C =";C
400 IF C<>0 THEN 430
410 PRINT"SORRY, C<>0{DOWN}"
420 GOTO 390
430 INPUT "ENTER D =";D
440 IF D<>0 THEN 470
450 PRINT"D CANNOT = 0{DOWN}"
460 GOTO 430
470 B=INT(100*(A*D/C+.005))/100
480 GOTO 730
490 INPUT "ENTER A =";A
500 INPUT "ENTER B =";B
510 IF B<>0 THEN 540
520 PRINT"B CANNOT = 0{DOWN}"
530 GOTO 500
540 INPUT "ENTER D =";D
550 IF D<>0 THEN 580
560 PRINT"D CANNOT = 0{DOWN}"
570 GOTO 540
580 C=INT(100*(A*D/B+.005))/100
590 GOTO 730
600 INPUT "ENTER A =";A
610 IF A<>0 THEN 640
620 PRINT"SORRY, A<>0{DOWN}"
630 GOTO 600
640 INPUT "ENTER B =";B
```

```
650 IF B<>0 THEN 680
660 PRINT"B CANNOT = 0{DOWN}"
670 GOTO 640
680 INPUT "ENTER C =";C
690 IF C<>0 THEN 720
700 PRINT"SORRY, C<>0{DOWN}"
710 GOTO 680
720 D=INT(100*(B*C/A+.005))/100
730 PRINT"{CLR}"
740 PRINT TAB(4);A,C
750 PRINT TAB(4);"----- = -----"
760 PRINT TAB(4);B,D
770 PRINT
780 PRINT"CHOOSE:"
790 PRINT" 1 ANOTHER PROBLEM"
800 PRINT" 2 END PROGRAM"
810 GET A$
820 IF A$="1" THEN 110
830 IF A$<>"2" THEN 810
840 PRINT"{CLR}":END
```

Program 7-17. Equivalent Fractions

TRS-80 Color Computer; MC-10; TRS-80 Model I

```
110 CLS
120 PRINT TAB(10);"A{5 SPACES}C"
130 PRINT TAB(10);"-   =   -"
140 PRINT TAB(10);"B{5 SPACES}D"
150 PRINT:PRINT
160 PRINT"WHICH IS THE UNKNOWN?"
170 PRINT"CHOOSE A, B, C, OR D"
180 A$=INKEY$:IF A$=""THEN 180
190 K=ASC(A$)
200 IF K<65 THEN 180
210 IF K>68 THEN 180
220 PRINT
230 ON K-64 GOTO 240,350,490,600
240 INPUT "ENTER B =";B
250 IF B<>0 THEN 280
260 PRINT"B CANNOT = 0":PRINT
270 GOTO 240
280 INPUT "ENTER C =";C
290 INPUT "ENTER D =";D
300 IF D<>0 THEN 330
310 PRINT"D CANNOT = 0":PRINT
320 GOTO 290
330 A=INT(100*(B*C/D+.005))/100
340 GOTO 730
350 INPUT "ENTER A =";A
360 IF A<>0 THEN 390
```

```
37Ø PRINT"SORRY, A<>Ø":PRINT
38Ø GOTO 35Ø
39Ø INPUT "ENTER C =";C
4ØØ IF C<>Ø THEN 43Ø
41Ø PRINT"SORRY, C<>Ø":PRINT
42Ø GOTO 39Ø
43Ø INPUT "ENTER D =";D
44Ø IF D<>Ø THEN 47Ø
45Ø PRINT"D CANNOT = Ø":PRINT
46Ø GOTO43Ø
47Ø B=INT(1ØØ*(A*D/C+.ØØ5))/1ØØ
48Ø GOTO 73Ø
49Ø INPUT "ENTER A =";A
5ØØ INPUT "ENTER B =";B
51Ø IF B<>Ø THEN 54Ø
52Ø PRINT"B CANNOT = Ø":PRINT
53Ø GOTO 5ØØ
54Ø INPUT "ENTER D =";D
55Ø IF D<>Ø THEN 58Ø
56Ø PRINT"D CANNOT = Ø":PRINT
57Ø GOTO 54Ø
58Ø C=INT(1ØØ*(A*D/B+.ØØ5))/1ØØ
59Ø GOTO 73Ø
6ØØ INPUT "ENTER A =";A
61Ø IF A<>Ø THEN 64Ø
62Ø PRINT"SORRY, A<>Ø":PRINT
63Ø GOTO 6ØØ
64Ø INPUT "ENTER B =";B
65Ø IF B<>Ø THEN 68Ø
66Ø PRINT"B CANNOT = Ø":PRINT
67Ø GOTO 64Ø
68Ø INPUT "ENTER C =";C
69Ø IF C<>Ø THEN 72Ø
7ØØ PRINT"SORRY, C<>Ø":PRINT
71Ø GOTO 68Ø
72Ø D=INT(1ØØ*(B*C/A+.ØØ5))/1ØØ
73Ø CLS
74Ø PRINT TAB(7);A,C
75Ø PRINT TAB(7);"----- = -----"
76Ø PRINTTAB(7);B,D
77Ø PRINT
78Ø PRINT"CHOOSE:"
79Ø PRINT" 1 ANOTHER PROBLEM"
8ØØ PRINT" 2 END PROGRAM"
81Ø A$=INKEY$
82Ø IF A$="1" THEN 11Ø
83Ø IF A$<>"2" THEN 81Ø
84Ø CLS:END
```

Program 7-18. Equivalent Fractions

TI-99/4A

```
11Ø CALL CLEAR
12Ø PRINT TAB(1Ø);"A{5 SPACES}C"
13Ø PRINT TAB(1Ø);"-  =  -"
14Ø PRINT TAB(1Ø);"B{5 SPACES}D":::
15Ø PRINT "WHICH IS THE UNKNOWN?"
16Ø PRINT "CHOOSE A, B, C, OR D.":::
17Ø CALL KEY(Ø,K,S)
18Ø IF (K<65)+(K>68)THEN 17Ø
19Ø ON K-64 GOTO 2ØØ,31Ø,45Ø,56Ø
2ØØ INPUT "ENTER B = ":B
21Ø IF B<>Ø THEN 24Ø
22Ø PRINT "B CANNOT = Ø"::
23Ø GOTO 2ØØ
24Ø INPUT "ENTER C = ":C
25Ø INPUT "ENTER D = ":D
26Ø IF D<>Ø THEN 29Ø
27Ø PRINT "D CANNOT = Ø"::
28Ø GOTO 25Ø
29Ø A=INT(1ØØ*(B*C/D+.ØØ5))/1ØØ
3ØØ GOTO 69Ø
31Ø INPUT "ENTER A = ":A
32Ø IF A<>Ø THEN 35Ø
33Ø PRINT "SORRY, A<>Ø"::
34Ø GOTO 31Ø
35Ø INPUT "ENTER C = ":C
36Ø IF C<>Ø THEN 39Ø
37Ø PRINT "SORRY, C<>Ø"::
38Ø GOTO 35Ø
39Ø INPUT "ENTER D = ":D
4ØØ IF D<>Ø THEN 43Ø
41Ø PRINT "D CANNOT = Ø"::
42Ø GOTO 39Ø
43Ø B=INT(1ØØ*(A*D/C+.ØØ5))/1ØØ
44Ø GOTO 69Ø
45Ø INPUT "ENTER A = ":A
46Ø INPUT "ENTER B = ":B
47Ø IF B<>Ø THEN 5ØØ
48Ø PRINT "B CANNOT = Ø"::
49Ø GOTO 46Ø
5ØØ INPUT "ENTER D = ":D
51Ø IF D<>Ø THEN 54Ø
52Ø PRINT "D CANNOT = Ø"::
53Ø GOTO 5ØØ
54Ø C=INT(1ØØ*(A*D/B+.ØØ5))/1ØØ
55Ø GOTO 69Ø
56Ø INPUT "ENTER A = ":A
57Ø IF A<>Ø THEN 6ØØ
```

```
580 PRINT "SORRY, A<>Ø"::
590 GOTO 56Ø
6ØØ INPUT "ENTER B = ":B
61Ø IF B<>Ø THEN 64Ø
62Ø PRINT "B CANNOT = Ø"::
63Ø GOTO 6ØØ
64Ø INPUT "ENTER C = ":C
65Ø IF C<>Ø THEN 68Ø
66Ø PRINT "SORRY, C<>Ø"::
67Ø GOTO 64Ø
68Ø D=INT(1ØØ*(B*C/A+.ØØ5))/1ØØ
69Ø CALL CLEAR
7ØØ PRINT TAB(7);A,C
71Ø PRINT TAB(7);"----- = -----"
72Ø PRINT TAB(7);B,D
73Ø PRINT :::"CHOOSE:"
74Ø PRINT " 1 ANOTHER PROBLEM"
75Ø PRINT " 2 END PROGRAM"
76Ø CALL KEY(Ø,K,S)
77Ø IF K=49 THEN 11Ø
78Ø IF K<>5Ø THEN 76Ø
79Ø CALL CLEAR
8ØØ END
```

Simplifying Fractions

Computers: VIC-20; TRS-80 Color Computer; MC-10; TRS-80 Model I; TI-99/4A

Enter a numerator, then a denominator. The computer simplifies or reduces the fraction to its lowest terms, or tells you if it cannot be simplified. This algorithm first checks which is larger, the numerator or the denominator. The first factor to be checked is the smaller number. If either the numerator or the denominator is an odd number, even factors will be eliminated by choosing a step size of −2 in the checking loop.

Although you usually reduce fractions starting with the smallest factors, the computer starts with the largest possible factor and decreases for each check.

Program 7-19. Simplifying Fractions

VIC-20

```
11Ø PRINT"{CLR}"
12Ø PRINT"SIMPLIFYING FRACTIONS"
13Ø PRINT
14Ø INPUT "NUMERATOR ={2 SPACES}";N
15Ø INPUT "DENOMINATOR =";D
16Ø IF D<>Ø THEN 19Ø
17Ø PRINT"DENOMINATOR CANNOT = Ø"
```

```
18Ø GOTO 15Ø
19Ø IF N<>Ø THEN 22Ø
2ØØ PRINT:PRINTN;"/";D;" = Ø"
21Ø GOTO 39Ø
22Ø IF D>N THEN 25Ø
23Ø L=D
24Ø GOTO 26Ø
25Ø L=N
26Ø S=-2
27Ø IF D/2<>INT(D/2)THEN 29Ø
28Ø IF N/2=INT(N/2)THEN 3ØØ
29Ø S=-1
3ØØ FOR C=L TO 2 STEP S
31Ø A=N/C
32Ø IF A<>INT(A) THEN 35Ø
33Ø B=D/C
34Ø IF B=INT(B)THEN 38Ø
35Ø NEXT C
36Ø PRINT:PRINTN;"/";D:PRINT" CANNOT BE SIMPLIFIED
    "
37Ø GOTO 39Ø
38Ø PRINT:PRINTN;"/";D;" = ";A;"/";B
39Ø PRINT"{DOWN}CHOOSE:"
4ØØ PRINT" 1 ANOTHER PROBLEM"
41Ø PRINT" 2 END PROGRAM"
42Ø GET A$
43Ø IF A$="1" THEN 11Ø
44Ø IF A$<>"2" THEN 42Ø
45Ø PRINT"{CLR}":END
```

Program 7-20. Simplifying Fractions

TRS-80 Color Computer; MC-10; TRS-80 Model I

```
11Ø CLS
12Ø PRINT"SIMPLIFYING FRACTIONS"
13Ø PRINT
14Ø INPUT "NUMERATOR =  ";N
15Ø INPUT "DENOMINATOR =";D
16Ø IF D<>Ø THEN 19Ø
17Ø PRINT"DENOMINATOR CANNOT = Ø"
18Ø GOTO 15Ø
19Ø IF N<>Ø THEN 22Ø
2ØØ PRINT:PRINTN;"/";D;" = Ø"
21Ø GOTO 39Ø
22Ø IF D>N THEN 25Ø
23Ø L=D
24Ø GOTO 26Ø
25Ø L=N
26Ø S=-2
27Ø IF D/2<>INT(D/2)THEN 29Ø
```

```
280 IF N/2=INT(N/2)THEN 300
290 S=-1
300 FOR C=L TO 2 STEP S
310 A=N/C
320 IF A<>INT(A) THEN 350
330 B=D/C
340 IF B=INT(B)THEN 380
350 NEXT C
360 PRINT:PRINTN;"/";D;" CANNOT BE SIMPLIFIED"
370 GOTO390
380 PRINT:PRINTN;"/";D;" = ";A;"/";B
390 PRINT:PRINT"CHOOSE:"
400 PRINT" 1 ANOTHER PROBLEM"
410 PRINT" 2 END PROGRAM"
420 A$=INKEY$
430 IF A$="1" THEN 110
440 IF A$<>"2" THEN 420
450 CLS:END
```

Program 7-21. Simplifying Fractions

TI-99/4A

```
110 CALL CLEAR
120 PRINT "SIMPLIFYING FRACTIONS"::::
130 INPUT "NUMERATOR ={3 SPACES}":N
140 INPUT "DENOMINATOR = ":D
150 IF D<>0 THEN 180
160 PRINT "DENOMINATOR CANNOT = 0"::
170 GOTO 140
180 IF D>N THEN 210
190 LIM=D
200 GOTO 220
210 LIM=N
220 S=-2
230 IF D/2<>INT(D/2)THEN 250
240 IF N/2=INT(N/2)THEN 260
250 S=-1
260 FOR C=LIM TO 2 STEP S
270 A=N/C
280 IF A<>INT(A)THEN 310
290 B=D/C
300 IF B=INT(B)THEN 340
310 NEXT C
320 PRINT ::N;"/";D;" CANNOT BE SIMPLIFIED"
330 GOTO 350
340 PRINT :N;"/";D;" = ";A;"/";B
350 PRINT :::"CHOOSE:"
360 PRINT " 1 ANOTHER PROBLEM"
370 PRINT " 2 END PROGRAM"
```

```
38Ø CALL KEY(Ø,K,S)
39Ø IF K=49 THEN 11Ø
4ØØ IF K<>5Ø THEN 38Ø
41Ø CALL CLEAR
42Ø END
```

Multiplying Fractions

Computers: VIC-20; TRS-80 Color Computer; MC-10; TRS-80 Model I; TI-99/4A

You can use this program to multiply from two to nine fractions. First, press a number key to represent the total number of fractions, then enter each numerator and denominator as you're prompted. The program multiplies the fractions and simplifies the final answer. To simplify fractions, the denominators are arranged in order, and the largest denominator is the first factor tried.

Program 7-22. Multiplying Fractions

VIC-20

```
11Ø PRINT"{CLR}"
12Ø PRINT"MULTIPLYING FRACTIONS"
13Ø PRINT"{DOWN}HOW MANY FRACTIONS? ";
14Ø GET A$:IF A$=""THEN 14Ø
15Ø IF ASC(A$)<5Ø THEN 14Ø
16Ø IF ASC(A$)>57 THEN 14Ø
17Ø PRINTA$
18Ø C=VAL(A$)
19Ø NT=1
2ØØ DT=1
21Ø FOR I=1 TO C
22Ø PRINT"{DOWN}FRACTION";I
23Ø INPUT" NUMERATOR{3 SPACES}";N(I)
24Ø NT=NT*N(I)
25Ø INPUT" DENOMINATOR ";D(I)
26Ø IF D(I)<>Ø THEN 29Ø
27Ø PRINT"DENOMINATOR CANNOT = Ø"
28Ø GOTO 25Ø
29Ø DT=DT*D(I)
3ØØ NEXT I
31Ø PRINT"{2 DOWN}** MULTIPLY **{DOWN}"
32Ø FOR I=1 TO C
33Ø PRINTN(I);"/";D(I)
34Ø NEXT I
35Ø PRINT"***************"
36Ø FOR I=1 TO C
37Ø A=NT/D(I)
38Ø IF A<>INT(A)THEN 43Ø
39Ø B=DT/D(I)
```

```
400 IF B<>INT(B)THEN 430
410 NT=A
420 DT=B
430 NEXT I
440 SW=0
450 FOR I=1 TO C-1
460 IF D(I)<=D(I+1) THEN 510
470 DD=D(I)
480 D(I)=D(I+1)
490 D(I+1)=DD
500 SW=1
510 NEXT I
520 IF SW=1 THEN 440
530 L=D(C)
540 FOR I=L TO 2 STEP -1
550 A=NT/I
560 IF A<>INT(A)THEN 590
570 B=DT/I
580 IF B=INT(B)THEN 620
590 NEXT I
600 A=NT
610 B=DT
620 IF A>=B THEN 650
630 PRINT:PRINTA;"/";B
640 GOTO 710
650 W=INT(A/B)
660 R=A-W*B
670 IF R<>0 THEN 700
680 PRINT W
690 GOTO 710
700 PRINTW:"{3 SPACES}";R;"/";B
710 PRINT"{DOWN}CHOOSE:"
720 PRINT" 1 ANOTHER PROBLEM"
730 PRINT" 2 END PROGRAM"
740 GET A$
750 IF A$="1" THEN 110
760 IF A$<>"2" THEN 740
770 PRINT"{CLR}":END
```

Program 7-23. Multiplying Fractions

TRS-80 Color Computer; MC-10; TRS-80 Model I

```
110 CLS
120 PRINT"MULTIPLYING FRACTIONS"
130 PRINT:PRINT"HOW MANY FRACTIONS?";
140 A$=INKEY$:IF A$=""THEN 140
150 IF ASC(A$)<50 THEN 140
160 IF ASC(A$)>57 THEN 140
170 PRINTA$
180 C=VAL(A$)
```

```
190 NT=1
200 DT=1
210 FOR I=1 TO C
220 PRINT"FRACTION";I
230 INPUT"  NUMERATOR =  ";N(I)
240 NT=NT*N(I)
250 INPUT"  DENOMINATOR =";D(I)
260 IF D(I)<>0 THEN 290
270 PRINT"DENOMINATOR CANNOT = 0":PRINT
280 GOTO 250
290 DT=DT*D(I)
300 NEXT I
310 PRINT:PRINT:PRINT"** MULTIPLY **":PRINT
320 FOR I=1 TO C
330 PRINTN(I);"/";D(I)
340 NEXT I
350 PRINT"----------------"
360 FOR I=1 TO C
370 A=NT/D(I)
380 IF A<>INT(A)THEN 430
390 B=DT/D(I)
400 IF B<>INT(B)THEN 430
410 NT=A
420 DT=B
430 NEXT I
440 SW=0
450 FOR I=1 TO C-1
460 IF D(I)<=D(I+1) THEN 510
470 DD=D(I)
480 D(I)=D(I+1)
490 D(I+1)=DD
500 SW=1
510 NEXT I
520 IF SW=1 THEN 440
530 L=D(C)
540 FOR I=L TO 2 STEP -1
550 A=NT/I
560 IF A<>INT(A)THEN 590
570 B=DT/I
580 IF B=INT(B)THEN 620
590 NEXT I
600 A=NT
610 B=DT
620 IF A>=B THEN 650
630 PRINT:PRINTA;"/";B
640 GOTO 710
650 W=INT(A/B)
660 R=A-W*B
670 IF R<>0 THEN 700
```

```
680 PRINT W
690 GOTO 710
700 PRINTW;"{3 SPACES}";R;"/";B
710 PRINT:PRINT"CHOOSE:"
720 PRINT" 1 ANOTHER PROBLEM"
730 PRINT" 2 END PROGRAM"
740 A$=INKEY$
750 IF A$="1" THEN 110
760 IF A$<>"2" THEN 740
770 CLS:END
```

Program 7-24. Multiplying Fractions

TI-99/4A

```
110 CALL CLEAR
120 PRINT "MULTIPLYING FRACTIONS"::::
130 PRINT "HOW MANY FRACTIONS?":::
140 CALL KEY(0,K,S)
150 IF (K<50)+(K>57)THEN 140
160 CALL HCHAR(21,23,K)
170 C=K-48
180 NT=1
190 DT=1
200 FOR I=1 TO C
210 PRINT "FRACTION";I
220 INPUT "{4 SPACES}NUMERATOR ={3 SPACES}":N(I)
230 NT=NT*N(I)
240 INPUT "{4 SPACES}DENOMINATOR = ":D(I)
250 IF D(I)<>0 THEN 280
260 PRINT :"DENOMINATOR CANNOT BE ZERO.")::
270 GOTO 240
280 DT=DT*D(I)
290 NEXT I
300 PRINT :::"** MULTIPLY **"::
310 FOR I=1 TO C
320 PRINT STR$(N(I));"/";STR$(D(I))
330 NEXT I
340 PRINT "---------------"
350 FOR I=1 TO C
360 A=NT/D(I)
370 IF A<>INT(A)THEN 420
380 B=DT/D(I)
390 IF B<>INT(B)THEN 420
400 NT=A
410 DT=B
420 NEXT I
430 SW=0
440 FOR I=1 TO C-1
450 IF D(I)<=D(I+1)THEN 500
460 DD=D(I)
```

```
470 D(I)=D(I+1)
480 D(I+1)=DD
490 SW=1
500 NEXT I
510 IF SW=1 THEN 430
520 L=D(C)
530 FOR I=L TO 2 STEP -1
540 A=NT/I
550 IF A<>INT(A)THEN 580
560 B=DT/I
570 IF B=INT(B)THEN 610
580 NEXT I
590 A=NT
600 B=DT
610 IF A>=B THEN 640
620 PRINT ::STR$(A);"/";STR$(B)
630 GOTO 700
640 W=INT(A/B)
650 R=A-W*B
660 IF R<>0 THEN 690
670 PRINT W
680 GOTO 700
690 PRINT W;"{3 SPACES}";STR$(R);"/";STR$(B)
700 PRINT :::"CHOOSE:":" 1 ANOTHER PROBLEM"
710 PRINT " 2 END PROGRAM";
720 CALL KEY(0,K,S)
730 IF K=49 THEN 110
740 IF K<>50 THEN 720
750 CALL CLEAR
760 END
```

Dividing Fractions

Computers: VIC-20; TRS-80 Color Computer; MC-10; TRS-80 Model I; TI-99/4A

You enter the numerator and denominator of each fraction, and the first fraction is divided by the second. The final answer is printed in simplified form.

Program 7-25. Dividing Fractions

VIC-20

```
110 PRINT"{CLR}"
120 PRINT"THE FIRST FRACTION IS DIVIDED BY THE"
130 PRINT"SECOND FRACTION:"
140 PRINT"(N1/D1) / (N2/D2){DOWN}"
150 INPUT "NUMERATOR 1{SHIFT-SPACE}";N1
160 INPUT "DENOMINATOR 1 ";D1
170 IF D1<>0 THEN 200
180 PRINT"SORRY CANNOT = 0"
```

```
190 GOTO 16Ø
2ØØ PRINT
21Ø INPUT "NUMERATOR 2 ";N2
22Ø IF N2<>Ø THEN 25Ø
23Ø PRINT"SORRY CANNOT = Ø"
24Ø GOTO 21Ø
25Ø INPUT "DENOMINATOR 2 ";D2
26Ø IF D2<>Ø THEN 29Ø
27Ø PRINT"SORRY CANNOT = Ø"
28Ø GOTO 25Ø
29Ø PRINT
3ØØ TN=N1*D2
31Ø TD=D1*N2
32Ø PRINTN1;"/";D1
33Ø PRINT"****************"
34Ø PRINTN2;"/";D2
35Ø PRINT"{DOWN}EQUALS"
36Ø IF N1<>Ø THEN 39Ø
37Ø PRINT"Ø"
38Ø GOTO 58Ø
39Ø IF TN<TD THEN PL=TN:GOTO 41Ø
4ØØ PL=TD
41Ø FOR P=PL TO 2 STEP -1
42Ø A=TN/P
43Ø IF A<>INT(A) THEN 46Ø
44Ø B=TD/P
45Ø IF B=INT(B) THEN 49Ø
46Ø NEXT P
47Ø A=TN
48Ø B=TD
49Ø IF A>=B THEN 52Ø
5ØØ PRINTA;"/";B
51Ø GOTO 58Ø
52Ø IF B=1 THEN 57Ø
53Ø C=INT(A/B)
54Ø R=A-C*B
55Ø PRINT C;"{2 SPACES}";R;"/";B
56Ø GOTO 58Ø
57Ø PRINT A
58Ø PRINT"{DOWN}CHOOSE:"
59Ø PRINT" 1 ANOTHER PROBLEM"
6ØØ PRINT" 2 END PROGRAM"
61Ø GET A$
62Ø IF A$="1" THEN 11Ø
63Ø IF A$<>"2" THEN 61Ø
64Ø PRINT"{CLR}":END
```

Program 7-26. Dividing Fractions

TRS-80 Color Computer; MC-10; TRS-80 Model I

```
110 CLS
120 PRINT"THE FIRST FRACTION IS DIVIDED"
130 PRINT"BY THE SECOND FRACTION:"
140 PRINT"(N1/D1) / (N2/D2)":PRINT
150 INPUT "NUMERATOR 1 =  ";N1
160 INPUT "DENOMINATOR 1 =";D1
170 IF D1<>Ø THEN 200
180 PRINT"SORRY CANNOT = Ø"
190 GOTO 160
200 PRINT
210 INPUT "NUMERATOR 2 =  ";N2
220 IF N2<>Ø THEN 250
230 PRINT"SORRY CANNOT = Ø"
240 GOTO 210
250 INPUT "DENOMINATOR 2 =";D2
260 IF D2<>Ø THEN 290
270 PRINT"SORRY CANNOT = Ø"
280 GOTO 250
290 PRINT
300 TN=N1*D2
310 TD=D1*N2
320 PRINTN1;"/";D1
330 PRINT"-----------------"
340 PRINTN2;"/";D2
350 PRINT:PRINT"EQUALS"
360 IF N1<>Ø THEN 390
370 PRINT"Ø"
380 GOTO 580
390 IF TN<TD THEN PL=TN:GOTO 410
400 PL=TD
410 FOR P=PL TO 2 STEP -1
420 A=TN/P
430 IF A<>INT(A) THEN 460
440 B=TD/P
450 IF B=INT(B) THEN 490
460 NEXT P
470 A=TN
480 B=TD
490 IF A>=B THEN 520
500 PRINTA;"/";B
510 GOTO 580
520 IF B=1 THEN 570
530 C=INT(A/B)
540 R=A-C*B
550 PRINT C;"  ";R;"/";B
560 GOTO 580
570 PRINT A
```

```
58Ø PRINT:PRINT"CHOOSE:"
59Ø PRINT" 1 ANOTHER PROBLEM"
6ØØ PRINT" 2 END PROGRAM";
61Ø A$=INKEY$
62Ø IF A$="1" THEN 11Ø
63Ø IF A$<>"2" THEN 61Ø
64Ø CLS:END
```

Program 7-27. Dividing Fractions

TI-99/4A

```
11Ø CALL CLEAR
12Ø PRINT "THE FIRST FRACTION IS"
13Ø PRINT "DIVIDED BY THE"
14Ø PRINT "SECOND FRACTION."
15Ø PRINT ::TAB(1Ø);"N1/D1"
16Ø PRINT TAB(9);"-------"
17Ø PRINT TAB(1Ø);"N2/D2":::
18Ø INPUT "ENTER N1 = ":N1
19Ø INPUT "ENTER D1 = ":D1
2ØØ IF D1<>Ø THEN 23Ø
21Ø PRINT :"DENOMINATOR CANNOT BE ZERO."::
22Ø GOTO 19Ø
23Ø PRINT
24Ø INPUT "ENTER N2 = ":N2
25Ø IF N2<>Ø THEN 28Ø
26Ø PRINT "SORRY, CANNOT = Ø"
27Ø GOTO 23Ø
28Ø INPUT "ENTER D2 = ":D2
29Ø IF D2<>Ø THEN 32Ø
3ØØ PRINT :"DENOMINATOR CANNOT BE ZERO."::
31Ø GOTO 28Ø
32Ø NT=N1*D2
33Ø DT=D1*N2
34Ø PRINT :::STR$(N1);"/";STR$(D1)
35Ø PRINT "----------------"
36Ø PRINT STR$(N2);"/";STR$(D2)
37Ø PRINT :::"EQUALS"::
38Ø IF NT<DT THEN 41Ø
39Ø L=DT
4ØØ GOTO 42Ø
41Ø L=NT
42Ø FOR I=L TO 2 STEP -1
43Ø A=NT/I
44Ø IF A<>INT(A)THEN 47Ø
45Ø B=DT/I
46Ø IF B=INT(B)THEN 5ØØ
47Ø NEXT I
48Ø A=NT
```

```
49Ø B=DT
5ØØ IF A>=B THEN 53Ø
51Ø PRINT ::STR$(A);"/";STR$(B)
52Ø GOTO 62Ø
53Ø IF B<>1 THEN 56Ø
54Ø PRINT ::A
55Ø GOTO 62Ø
56Ø C=INT(A/B)
57Ø R=A-C*B
58Ø IF R=Ø THEN 61Ø
59Ø PRINT C;"  ";STR$(R);"/";STR$(B)
6ØØ GOTO 62Ø
61Ø PRINT C
62Ø PRINT :::"CHOOSE:":" 1 ANOTHER PROBLEM"
63Ø PRINT " 2 END PROGRAM";
64Ø CALL KEY(Ø,K,S)
65Ø IF K=49 THEN 11Ø
66Ø IF K<>5Ø THEN 64Ø
67Ø CALL CLEAR
68Ø END
```

Adding Fractions

Computers: VIC-20; TRS-80 Color Computer; TRS-80 Model I; TI-99/4A

"Adding Fractions" has two main options, adding fractions with like denominators, such as 1/12+5/12+7/12, or adding fractions with unlike denominators, such as 1/2+1/3+1/4. The program will add up to nine fractions with *like* denominators or up to five fractions with *unlike* denominators.

If you choose the option of like denominators, first press the total number of fractions to be added. Then enter the denominator, followed by the numerators.

If you choose the unlike denominators option, press a number key from 2 to 5 for the number of fractions. The numerator and then the denominator are entered for each fraction.

The fractions are added, the problem is rewritten, and then the answer is printed in simplified terms.

Program 7-28. Adding Fractions

VIC-20

```
11Ø PRINT"{CLR}"
12Ø PRINT"ADDING FRACTIONS"
13Ø PRINT"{DOWN}CHOOSE:"
14Ø PRINT"1 LIKE DENOMINATORS"
15Ø PRINT"2 UNLIKE DENOMINATORS"
16Ø GET A$
```

```
170 IF A$="2" THEN 41Ø
18Ø IF A$<>"1" THEN 16Ø
19Ø PRINT"{CLR}"
2ØØ CH=1
21Ø PRINT"ADDING FRACTIONS WITH"
22Ø PRINT"LIKE DENOMINATORS"
23Ø PRINT"{DOWN}HOW MANY FRACTIONS? ";
24Ø GET A$:IF A$="" THEN 24Ø
25Ø IF ASC(A$)<49 THEN 24Ø
26Ø IF ASC(A$)>57 THEN 24Ø
27Ø PRINTA$
28Ø C=VAL(A$)
29Ø INPUT "DENOMINATOR{2 SPACES}= ";DT
3ØØ IF DT<>Ø THEN 33Ø
31Ø PRINT"DENOMINATOR CANNOT = Ø"
32Ø GOTO 29Ø
33Ø PRINT"{DOWN}ENTER NUMERATORS"
34Ø NT=Ø
35Ø FOR I=1 TO C
36Ø INPUT N(I)
37Ø NT=NT+N(I)
38Ø D(I)=DT
39Ø NEXT I
4ØØ GOTO 73Ø
41Ø PRINT"{CLR}"
42Ø PRINT"ADDING UP TO FIVE"
43Ø PRINT"FRACTIONS WHICH HAVE"
44Ø PRINT"UNLIKE DENOMINATORS"
45Ø PRINT"{DOWN}HOW MANY FRACTIONS? ";
46Ø GET A$:IF A$="" THEN 46Ø
47Ø IF ASC(A$)<49 THEN 46Ø
48Ø IF ASC(A$)>53 THEN 46Ø
49Ø PRINTA$
5ØØ C=VAL(A$)
51Ø NT=Ø
52Ø DT=1
53Ø FOR I=1 TO C
54Ø PRINT "FRACTION";I
55Ø INPUT " NUMERATOR{3 SPACES}";N(I)
56Ø INPUT " DENOMINATOR ";D(I)
57Ø IF D(I)<>Ø THEN 6ØØ
58Ø PRINT"DENOMINATOR CANNOT = Ø"
59Ø GOTO 56Ø
6ØØ IF I=1 THEN 64Ø
61Ø FOR J=1 TO I-1
62Ø IF D(I)=D(J) THEN 66Ø
63Ø NEXT J
64Ø F=D(I)
65Ø GOTO 67Ø
66Ø F=1
```

```
670 DT=DT*F
680 NEXT I
690 FOR I=1 TO C
700 F=DT/D(I)
710 NT=NT+N(I)*F
720 NEXT I
730 PRINT"{CLR}"
740 PRINT"ADDING FRACTIONS{DOWN}"
750 FOR I=1 TO C
760 PRINTN(I);"/";D(I)
770 NEXT I
780 PRINT"**************"
790 IF DT>NT THEN 820
800 L=DT
810 GOTO 830
820 L=NT
830 S=-2
840 IF DT/2<>INT(DT/2)THEN 860
850 IF NT/2=INT(NT/2)THEN 870
860 S=-1
870 FOR I=L TO 2 STEP S
880 A=NT/I
890 IF A<>INT(A) THEN 920
900 B=DT/I
910 IF B=INT(B) THEN 950
920 NEXT I
930 A=NT
940 B=DT
950 PRINTA;"/";B
960 IF A<B THEN 1030
970 W=INT(A/B)
980 R=A-W*B
990 IF R<>0 THEN 1020
1000 PRINT"OR";W
1010 GOTO 1030
1020 PRINT"OR";W;"{2 SPACES}";R;"/";B
1030 PRINT"{DOWN}CHOOSE:"
1040 PRINT" 1 ANOTHER PROBLEM"
1050 PRINT" 2 END PROGRAM"
1060 GET A$
1070 IF A$="1" THEN 110
1080 IF A$<>"2" THEN 1060
1090 PRINT"{CLR}":END
```

Program 7-29. Adding Fractions

TRS-80 Color Computer; TRS-80 Model I

```
110 CLS
120 PRINT"ADDING FRACTIONS"
130 PRINT:PRINT"CHOOSE:"
```

```
14Ø PRINT" 1 LIKE DENOMINATORS"
15Ø PRINT" 2 UNLIKE DENOMINATORS"
16Ø A$=INKEY$
17Ø IF A$="2" THEN 41Ø
18Ø IF A$<>"1" THEN 16Ø
19Ø CLS
2ØØ CH=1
21Ø PRINT"ADDING FRACTIONS WITH"
22Ø PRINT"LIKE DENOMINATORS"
23Ø PRINT:PRINT"HOW MANY FRACTIONS? ";
24Ø A$=INKEY$:IF A$=""THEN 24Ø
25Ø IF ASC(A$)<49 THEN 24Ø
26Ø IF ASC(A$)>57 THEN 24Ø
27Ø PRINTA$
28Ø C=VAL(A$)
29Ø INPUT "DENOMINATOR = ";DT
3ØØ IF DT<>Ø THEN 33Ø
31Ø PRINT"DENOMINATOR CANNOT = Ø"
32Ø GOTO 29Ø
33Ø PRINT:PRINT"ENTER NUMERATORS"
34Ø NT=Ø
35Ø FOR I=1 TO C
36Ø INPUT N(I)
37Ø NT=NT+N(I)
38Ø D(I)=DT
39Ø NEXT I
4ØØ GOTO 73Ø
41Ø CLS
42Ø PRINT"ADDING UP TO FIVE"
43Ø PRINT"FRACTIONS WHICH MAY HAVE"
44Ø PRINT"UNLIKE DENOMINATORS"
45Ø PRINT:PRINT"HOW MANY FRACTIONS? ";
46Ø A$=INKEY$:IF A$=""THEN 46Ø
47Ø IF ASC(A$)<49 THEN 46Ø
48Ø IF ASC(A$)>53 THEN 46Ø
49Ø PRINTA$
5ØØ C=VAL(A$)
51Ø NT=Ø
52Ø DT=1
53Ø FOR I=1 TO C
54Ø PRINT "FRACTION";I
55Ø INPUT " NUMERATOR = ";N(I)
56Ø INPUT " DENOMINATOR =";D(I)
57Ø IF D(I)<>Ø THEN 6ØØ
58Ø PRINT"DENOMINATOR CANNOT = Ø"
59Ø GOTO 56Ø
6ØØ IF I=1 THEN 64Ø
61Ø FOR J=1 TO I-1
62Ø IF D(I)=D(J) THEN 66Ø
```

```
63Ø NEXT J
64Ø F=D(I)
65Ø GOTO 67Ø
66Ø F=1
67Ø DT=DT*F
68Ø NEXT I
69Ø FOR I=1 TO C
7ØØ F=DT/D(I)
71Ø NT=NT+N(I)*F
72Ø NEXT I
73Ø CLS
74Ø PRINT"ADDING FRACTIONS":PRINT
75Ø FOR I=1 TO C
76Ø PRINTN(I);"/";D(I)
77Ø NEXT I
78Ø PRINT"----------------"
79Ø IF DT>NT THEN 82Ø
8ØØ L=DT
81Ø GOTO 83Ø
82Ø L=NT
83Ø ST=-2
84Ø IF DT/2<>INT(DT/2)THEN 86Ø
85Ø IF NT/2=INT(NT/2)THEN 87Ø
86Ø ST=-1
87Ø FOR I=L TO2 STEP ST
88Ø A=NT/I
89Ø IF A<>INT(A) THEN 92Ø
9ØØ B=DT/I
91Ø IF B=INT(B) THEN 95Ø
92Ø NEXT I
93Ø A=NT
94Ø B=DT
95Ø PRINTA;"/";B
96Ø IF A<B THEN 1Ø3Ø
97Ø W=INT(A/B)
98Ø R=A-W*B
99Ø IF R<>Ø THEN 1Ø2Ø
1ØØØ PRINT"OR";W
1Ø1Ø GOTO 1Ø3Ø
1Ø2Ø PRINT"OR";W;"  ";R;"/";B
1Ø3Ø PRINT:PRINT"CHOOSE:"
1Ø4Ø PRINT" 1 ANOTHER PROBLEM"
1Ø5Ø PRINT" 2 END PROGRAM";
1Ø6Ø A$=INKEY$
1Ø7Ø IF A$="1" THEN 11Ø
1Ø8Ø IF A$<>"2" THEN 1Ø6Ø
1Ø9Ø CLS:END
```

Program 7-30. Adding Fractions

TI-99/4A

```
110 CALL CLEAR
120 PRINT "** ADDING FRACTIONS **"
130 PRINT :::"CHOOSE:"
140 PRINT :"1  LIKE DENOMINATORS"
150 PRINT :"2  UNLIKE DENOMINATORS":::
160 CALL KEY(Ø,K,S)
170 IF K=5Ø THEN 41Ø
180 IF K<>49 THEN 16Ø
190 CALL CLEAR
200 CH=1
210 PRINT "ADDING FRACTIONS WITH"
220 PRINT "LIKE DENOMINATORS"
230 PRINT ::"HOW MANY FRACTIONS?"
240 CALL KEY(Ø,K,S)
250 IF (K<5Ø)+(K>57)THEN 24Ø
260 CALL HCHAR(23,23,K)
270 C=K-48
280 PRINT ::"WHAT IS THE DENOMINATOR?"
290 INPUT DT
300 IF DT<>Ø THEN 33Ø
310 PRINT "DENOMINATOR CANNOT = Ø"
320 GOTO 28Ø
330 PRINT ::"ENTER THE NUMERATORS"::
340 NT=Ø
350 FOR I=1 TO C
360 INPUT N(I)
370 NT=NT+N(I)
380 D(I)=DT
390 NEXT I
400 GOTO 72Ø
410 CALL CLEAR
420 PRINT "ADDING UP TO FIVE"
430 PRINT "FRACTIONS WHICH MAY HAVE"
440 PRINT "UNLIKE DENOMINATORS"
450 PRINT ::"HOW MANY FRACTIONS?"::
460 CALL KEY(Ø,K,S)
470 IF (K<5Ø)+(K>53)THEN 46Ø
480 CALL HCHAR(22,23,K)
490 C=K-48
500 NT=Ø
510 DT=1
520 FOR I=1 TO C
530 PRINT "FRACTION";I
540 INPUT "{3 SPACES}NUMERATOR ={3 SPACES}":N(I)
550 INPUT "{3 SPACES}DENOMINATOR = ":D(I)
560 IF D(I)<>Ø THEN 59Ø
570 PRINT :"DENOMINATOR CANOT BE ZERO"::
580 GOTO 55Ø
```

```
590 IF I=1 THEN 63Ø
6ØØ FOR J=1 TO I-1
61Ø IF D(I)=D(J)THEN 65Ø
62Ø NEXT J
63Ø F=D(I)
64Ø GOTO 66Ø
65Ø F=1
66Ø DT=DT*F
67Ø NEXT I
68Ø FOR I=1 TO C
69Ø F=DT/D(I)
7ØØ NT=NT+N(I)*F
71Ø NEXT I
72Ø CALL CLEAR
73Ø PRINT "** ADDING FRACTIONS **":::
74Ø FOR I=1 TO C
75Ø PRINT STR$(N(I));"/";STR$(D(I))
76Ø NEXT I
77Ø PRINT "---------------"::
78Ø IF DT>NT THEN 81Ø
79Ø L=DT
8ØØ GOTO 82Ø
81Ø L=NT
82Ø ST=-2
83Ø IF DT/2<>INT(DT/2)THEN 85Ø
84Ø IF NT/2=INT(NT/2)THEN 86Ø
85Ø ST=-1
86Ø FOR I=L TO 2 STEP ST
87Ø A=NT/I
88Ø IF A<>INT(A)THEN 91Ø
89Ø B=DT/I
9ØØ IF B=INT(B)THEN 94Ø
91Ø NEXT I
92Ø A=NT
93Ø B=DT
94Ø PRINT STR$(A);"/";STR$(B)
95Ø IF A<B THEN 1Ø2Ø
96Ø W=INT(A/B)
97Ø R=A-W*B
98Ø IF R<>Ø THEN 1Ø1Ø
99Ø PRINT "OR";W
1ØØØ GOTO 1Ø2Ø
1Ø1Ø PRINT :"OR ";W;"  ";STR$(R);"/";STR$(B)
1Ø2Ø PRINT ::"CHOOSE:":" 1 ANOTHER PROBLEM"
1Ø3Ø PRINT " 2 END PROGRAM"
1Ø4Ø CALL KEY(Ø,K,S)
1Ø5Ø IF K=49 THEN 11Ø
1Ø6Ø IF K<>5Ø THEN 1Ø4Ø
1Ø7Ø CALL CLEAR
1Ø8Ø END
```

Simultaneous Equations

Computers: VIC-20; TRS-80 Color Computer; TRS-80 Model I; TI-99/4A

You can solve simultaneous equations using the matrix inversion technique with this program, which solves up to six equations with six unknowns. If you have a computer with more memory, you can increase the number of equations by changing the DIMension statement and the statement limiting your choice of the number of equations. If you have a 40-column or 80-column screen, you may wish to change the printing.

First, you enter the number of equations and unknowns (the degree of the matrix). Next, enter the coefficients row by row with the corresponding B elements in the matrix of equations [A]·[X]=[B]. Here's an example problem.

Given three equations with three unknowns:

$$\begin{aligned} x_1 + x_2 + x_3 &= 12 \\ 2x_1 + x_2 + 3x_3 &= 25 \\ x_1 + 3x_2 + 2x_3 &= 25 \end{aligned}$$

Rewritten in matrix form, this is:

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & 3 \\ 1 & 3 & 2 \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 12 \\ 25 \\ 25 \end{bmatrix}$$

The group of numbers on the left-hand side of the matrix represents *A*, in the equation [A]*[X]=[B]. *A* in the equation $x_1+x_2+x_3=12$ is always *1*, while *A* in the equation $2x_1+x_2+3x_3=25$ equals 2, 1, and 3.

To solve this system using the program, you enter the degree of the matrix as 3 for the *three* equations with three unknowns. In order, the following numbers are entered:

A(1,1)=1
A(1,2)=1
A(1,3)=1
B(1) =12

A(2,1)=2
A(2,2)=1
A(2,3)=3
B(2) =25

A(3,1) = 1
A(3,2) = 3
A(3,3) = 2
B(3)   = 25

Notice that the numbers you enter correspond to the values of *A* you saw in the matrix form of the equation.

The solution is then printed:

X(1) = 3
X(2) = 4
X(3) = 5

Program 7-31. Simultaneous Equations

VIC-20

```
110 GOTO 240
120 REM SUB TO SWITCH ROWS
130 FOR C=I+1 TO N
140 IF W(C,I)=0 THEN 200
150 FOR L=1 TO N
160 DW=W(I,L):W(I,L)=W(C,L):W(C,L)=DW
170 NEXT L
180 DB=B(I):B(I)=B(C):B(C)=DB
190 RETURN
200 NEXT C
210 PRINT"{DOWN}DETERMINANT = 0."
220 PRINT"NO UNIQUE SOLUTION.{DOWN}"
230 GOTO 740
240 PRINT"{CLR}"
250 DIM A(6,6),W(6,6),X(6),B(6)
260 PRINT"SOLUTION OF"
270 PRINT"SIMULTANEOUS EQUATIONS"
280 PRINT"MATRIX INVERSION":PRINT"TECHNIQUE TO SOL
    VE"
290 PRINT" [A][X] = [B]"
350 PRINT"{DOWN}NUMBER OF EQUATIONS ORDEGREE OF MA
    TRIX,"
360 PRINT"N = ";
370 GET I$:IF I$="" THEN 370
380 IF ASC(I$)<50 OR ASC(I$)>54 THEN 370
390 N=VAL(I$):PRINTN
400 PRINT"{DOWN}ENTER [A] VALUES WITH"
410 PRINT"CORRESPONDING [B]"
420 PRINT"FOR EACH EQUATION.{DOWN}"
430 FOR I=1 TO N:FOR J=1 TO N
440 PRINT "A("+RIGHT$(STR$(I),1)+","+RIGHT$(STR$(J
    ),1)+") = ";:INPUT A(I,J)
450 W(I,J)=A(I,J):NEXT J
460 PRINT "B("+RIGHT$(STR$(I),1)+") = ";:INPUT B(I
    )
```

```
470 PRINT:NEXT I
480 REM INVERT MATRIX A
490 FOR I=1 TO N
500 IF W(I,I)=Ø THEN GOSUB 13Ø
510 W(I,I)=1/W(I,I)
520 FOR J=1 TO N
530 IF J-I=Ø THEN 59Ø
540 W(J,I)=W(J,I)*W(I,I)
550 FOR K=1 TO N
560 IF K-I=Ø THEN 58Ø
570 W(J,K)=W(J,K)-W(J,I)*W(I,K)
580 NEXT K
590 NEXT J
600 FOR K=1 TO N
610 IF K-I=Ø THEN 63Ø
620 W(I,K)=-W(I,I)*W(I,K)
630 NEXT K
640 NEXT I
650 PRINT"{2 DOWN}SOLUTION VECTOR X:{DOWN}"
660 FOR I=1 TO N
670 X(I)=Ø
680 FOR J=1 TO N
690 X(I)=X(I)+W(I,J)*B(J)
700 NEXT J
710 PRINT"X("+RIGHT$(STR$(I),1)+") = ";X(I)
720 NEXT I
730 PRINT
740 END
```

Program 7-32. Simultaneous Equations

TRS-80 Color Computer; TRS-80 Model I

```
110 GOTO 24Ø
120 REM SUB TO SWITCH ROWS
130 FOR C=I+1 TO N
140 IF W(C,I)=Ø THEN 2ØØ
150 FOR L=1 TO N
160 DW=W(I,L):W(I,L)=W(C,L):W(C,L)=DW
170 NEXT L
180 DB=B(I):B(I)=B(C):B(C)=DB
190 RETURN
200 NEXT C
210 PRINT:PRINT "SORRY, DETERMINANT = Ø."
220 PRINT "NO UNIQUE SOLUTION.":PRINT
230 GOTO 74Ø
240 CLS
250 DIM A(6,6),W(6,6),X(6),B(6)
260 PRINT"SOLUTION OF SIMULTANEOUS"
270 PRINT"EQUATIONS"
280 PRINT:PRINT"MATRIX INVERSION TECHNIQUE"
```

```
29Ø PRINT"TO SOLVE  [A][X] = [B]"
3ØØ PRINT:PRINT"PRESS <ENTER>"
31Ø I$=INKEY$
32Ø IF I$<>CHR$(13) THEN 31Ø
33Ø CLS
34Ø PRINT"[A]*[X] = [B]"
35Ø PRINT"PRESS NUMBER OF EQUATIONS OR"
36Ø PRINT"DEGREE OF THE MATRIX, N = ";
37Ø I$=INKEY$:IF I$="" THEN 37Ø
38Ø IF ASC(I$)<5Ø OR ASC(I$)>54 THEN 37Ø
39Ø N=VAL(I$):PRINTN
4ØØ PRINT:PRINT "ENTER [A] VALUES WITH"
41Ø PRINT "CORRESPONDING [B] VALUE"
42Ø PRINT "FOR EACH EQUATION.":PRINT
43Ø FOR I=1 TO N:FOR J=1 TO N
44Ø PRINT "A("+RIGHT$(STR$(I),1)+","+RIGHT$(STR$(J
    ),1)+") = ";:INPUT A(I,J)
45Ø W(I,J)=A(I,J):NEXT J
46Ø PRINT "B("+RIGHT$(STR$(I),1)+") = ";:INPUT B(I
    )
47Ø PRINT:NEXT I
48Ø REM INVERT MATRIX A
49Ø FOR I=1 TO N
5ØØ IF W(I,I)=Ø THEN GOSUB 13Ø
51Ø W(I,I)=1/W(I,I)
52Ø FOR J=1 TO N
53Ø IF J-I=Ø THEN 59Ø
54Ø W(J,I)=W(J,I)*W(I,I)
55Ø FOR K=1 TO N
56Ø IF K-I=Ø THEN 58Ø
57Ø W(J,K)=W(J,K)-W(J,I)*W(I,K)
58Ø NEXT K
59Ø NEXT J
6ØØ FOR K=1 TO N
61Ø IF K-I=Ø THEN 63Ø
62Ø W(I,K)=-W(I,I)*W(I,K)
63Ø NEXT K
64Ø NEXT I
65Ø PRINT:PRINT:PRINT "SOLUTION VECTOR X:":PRINT
66Ø FOR I=1 TO N
67Ø X(I)=Ø
68Ø FOR J=1 TO N
69Ø X(I)=X(I)+W(I,J)*B(J)
7ØØ NEXT J
71Ø PRINT "X("+RIGHT$(STR$(I),1)+") = ";X(I)
72Ø NEXT I
73Ø PRINT
74Ø END
```

Program 7-33. Simultaneous Equations

TI-99/4A

```
11Ø CALL CLEAR
12Ø PRINT "SOLVING SIMULTANEOUS"
13Ø PRINT "EQUATIONS BY THE"
14Ø PRINT "MATRIX INVERSION TECHNIQUE"
15Ø PRINT ::"SOLVE [A][X] = [B]"
16Ø PRINT :"ENTER DEGREE OF THE MATRIX"
17Ø PRINT "OR NUMBER OF EQUATIONS"::
18Ø INPUT "N = ":N
19Ø IF N<1Ø THEN 22Ø
2ØØ PRINT :"N MUST BE < 1Ø"::
21Ø GOTO 18Ø
22Ø IF N>1 THEN 25Ø
23Ø PRINT :"1<N<1Ø{3 SPACES}TRY AGAIN"::
24Ø GOTO 18Ø
25Ø PRINT ::"THE COEFFICIENTS OF X"
26Ø PRINT "ARE THE 'A' MATRIX."
27Ø PRINT :"INPUT THE VALUES ROW BY ROW:"
28Ø PRINT :"A(1,1),A(1,2),A(1,3),..."
29Ø PRINT "A(2,1),A(2,2),A(2,3),..."
3ØØ PRINT ".":".":"."
31Ø PRINT "A(N,1),A(N,2),...,A(N,N)":::
32Ø FOR I=1 TO N
33Ø FOR J=1 TO N
34Ø INPUT "A("&STR$(I)&","&STR$(J)&") = ":A(I,J)
35Ø W(I,J)=A(I,J)
36Ø NEXT J
37Ø PRINT
38Ø INPUT "B("&STR$(I)&") = ":B(I)
39Ø PRINT ::
4ØØ NEXT I
41Ø PRINT ::"--SOLVING--"::
42Ø REM  INVERT MATRIX A
43Ø FOR C=1 TO N
44Ø IF W(C,C)<>Ø THEN 46Ø
45Ø GOSUB 71Ø
46Ø W(C,C)=1/W(C,C)
47Ø FOR D=1 TO N
48Ø IF (D-C)=Ø THEN 54Ø
49Ø W(D,C)=W(D,C)*W(C,C)
5ØØ FOR E=1 TO N
51Ø IF (E-C)=Ø THEN 53Ø
52Ø W(D,E)=W(D,E)-W(D,C)*W(C,E)
53Ø NEXT E
54Ø NEXT D
55Ø FOR E=1 TO N
56Ø IF (E-C)=Ø THEN 58Ø
57Ø W(C,E)=-W(C,C)*W(C,E)
```

```
58Ø NEXT E
59Ø NEXT C
6ØØ PRINT ::"SOLUTION VECTOR X:"::
61Ø FOR I=1 TO N
62Ø X(I)=Ø
63Ø FOR J=1 TO N
64Ø X(I)=X(I)+W(I,J)*B(J)
65Ø NEXT J
66Ø PRINT :"X("&STR$(I)&") = ";X(I)
67Ø NEXT I
68Ø PRINT ::
69Ø GOTO 87Ø
7ØØ REM  SUB TO SWITCH ROWS
71Ø FOR F=C+1 TO N
72Ø IF W(F,C)=Ø THEN 82Ø
73Ø FOR E=1 TO N
74Ø DW=W(C,E)
75Ø W(C,E)=W(F,E)
76Ø W(F,E)=DW
77Ø NEXT E
78Ø DB=B(C)
79Ø B(C)=B(F)
8ØØ B(F)=DB
81Ø GOTO 86Ø
82Ø NEXT F
83Ø PRINT "SORRY, DETERMINANT=Ø."
84Ø PRINT "NO UNIQUE SOLUTION."
85Ø GOTO 87Ø
86Ø RETURN
87Ø PRINT :"PRESS 1 FOR ANOTHER PROBLEM"
88Ø PRINT TAB(7);"2 TO END PROGRAM"
89Ø CALL KEY(Ø,K,S)
9ØØ IF K=49 THEN 11Ø
91Ø IF K<>5Ø THEN 89Ø
92Ø CALL CLEAR
93Ø END
```

Chapter 8
Computer Sorting

Chapter 8
Computer Sorting

One of the functions of a computer is to organize data. You can use a sort routine to arrange numbers in ascending or descending order or to alphabetize lists. There are many kinds of sort routines. Four sorts are presented here.

To illustrate the procedure, 50 random numbers are generated and printed, then sorted. In a regular program you would list the actual data to sort, then perhaps call a sort as a subroutine.

The numbers used in these programs are contained in the A array. The DIMension statement at the beginning of the programs allow for the 50 numbers of the example. When you place these sorts in your own programs, make sure the value *n* in DIM A(*n*) reflects the total number of items to be sorted.

These sort programs arrange numbers in ascending order (smaller to larger). To change to descending order, change the less than signs to greater than signs. To use strings, when you're using names as data, for example, put dollar signs after the variable names to indicate string variables.

Bubble Sort

Computers: VIC-20; TRS-80 Color Computer; TRS-80 Model I; TI-99/4A

The bubble sort is a simple interchange sort and can be used for lists that do not contain many numbers, or for lists in which the numbers are not much out of order — it is slow for other uses. The program compares each number to the next number and exchanges numbers where necessary. If an exchange has been made during a pass through all the numbers, the loop of comparisons starts over.

Program 8-1. Bubble Sort

VIC-20

```
11Ø DIM A(5Ø)
12Ø FORI=1TO5Ø:A(I)=INT(1ØØ*RND(Ø)+1):PRINTA(I);:N
    EXT:PRINT:PRINT
13Ø L=49
14Ø S=Ø:FORI=1TOL:IFA(I)<=A(I+1)THEN16Ø
15Ø D=A(I):A(I)=A(I+1):A(I+1)=D:S=1:L=I
16Ø NEXT
17Ø IFS=1THEN14Ø
18Ø FORI=1TO5Ø:PRINTA(I);:NEXT
19Ø END
```

Program 8-2. Bubble Sort

TRS-80 Color Computer; TRS-80 Model I

```
110 DIM A(50)
120 FORI=1TO50:A(I)=RND(100):PRINTA(I);:NEXT:PRINT
    :PRINT
130 L=49
140 S=0:FORI=1TOL:IFA(I)<=A(I+1)THEN160
150 D=A(I):A(I)=A(I+1):A(I+1)=D:S=1:L=I
160 NEXT
170 IFS=1THEN140
180 FORI=1TO50:PRINTA(I);:NEXT
190 END
```

Program 8-3. Bubble Sort

TI-99/4A

```
110 DIM A(50)
120 FOR I=1 TO 50
130 RANDOMIZE
140 A(I)=INT(RND*100+1)
150 PRINT A(I);
160 NEXT I
170 PRINT ::
200 LIM=49
210 SW=0
220 FOR I=1 TO LIM
230 IF A(I)<=A(I+1)THEN 290
240 AA=A(I)
250 A(I)=A(I+1)
260 A(I+1)=AA
270 SW=1
280 LIM=I
290 NEXT I
300 IF SW=1 THEN 210
500 FOR I=1 TO 50
510 PRINT A(I);
520 NEXT I
530 END
```

Shell Sort

Computers: VIC-20; TRS-80 Color Computer; TRS-80 Model I; TI-99/4A

The shell sort is considerably faster than the bubble sort because the number of comparisons to be made is reduced.

In an array of N numbers, the program first determines a number B so that $2^B <$ N $< 2^{B+1}$, then initializes B to 2^{B-1}. The main loop varies a counter I from 1 to N − B. Within the loop, the program checks if A(I) < = A(I + B). If so, it increments I and continues the comparisons. If not, it exchanges A(I) and A(I + B) and changes the subscript. When I reaches the value of N, B is reduced by a factor of 2 and the loop starts over. When B=0, the sort is complete.

As with the bubble sort, this program generates 50 random numbers. To use this sort in your own programs, you'd have to enter DATA and have the program READ it into the array. Make sure you also change the values throughout the sort routine to show the number of items to be sorted.

Program 8-4. Shell Sort

VIC-20

```
11Ø DIM A(5Ø)
12Ø FORI=1TO5Ø:A(I)=INT(1ØØ*RND(Ø)+1):PRINTA(I);:N
    EXT:PRINT:PRINT
13Ø B=1
14Ø B=2*B:IFB<=5ØTHEN14Ø
15Ø B=INT(B/2):IFB=ØTHEN2ØØ
16Ø FORI=1TO5Ø-B:C=I
17Ø D=C+B:IFA(C)<=A(D)THEN19Ø
18Ø X=A(C):A(C)=A(D):A(D)=X:C=C-B:IFC>ØTHEN17Ø
19Ø NEXT:GOTO15Ø
2ØØ FORI=1TO5Ø:PRINTA(I);:NEXT
21Ø END
```

Program 8-5. Shell Sort

TRS-80 Color Computer; TRS-80 Model I

```
11Ø DIM A(5Ø)
12Ø FORI=1TO5Ø:A(I)=RND(1ØØ):PRINTA(I);:NEXT:PRINT
    :PRINT
13Ø B=1
14Ø B=2*B:IFB<=5ØTHEN14Ø
15Ø B=INT(B/2):IFB=ØTHEN2ØØ
16Ø FORI=1TO5Ø-B:C=I
17Ø D=C+B:IFA(C)<=A(D)THEN19Ø
18Ø X=A(C):A(C)=A(D):A(D)=X:C=C-B:IFC>ØTHEN17Ø
19Ø NEXT:GOTO15Ø
2ØØ FORI=1TO5Ø:PRINTA(I);:NEXT
21Ø END
```

Program 8-6. Shell Sort

TI-99/4A

```
11Ø DIM A(5Ø)
12Ø FOR I=1 TO 5Ø
13Ø RANDOMIZE
14Ø A(I)=INT(RND*1ØØ+1)
15Ø PRINT A(I);
16Ø NEXT I
```

```
170 PRINT ::
200 B=1
210 B=2*B
220 IF B<=50 THEN 210
230 B=INT(B/2)
240 IF B=0 THEN 500
250 FOR I=1 TO 50-B
260 C=I
270 D=C+B
280 IF A(C)<=A(D)THEN 340
290 AA=A(C)
300 A(C)=A(D)
310 A(D)=AA
320 C=C-B
330 IF C>0 THEN 270
340 NEXT I
350 GOTO 230
500 FOR I=1 TO 50
510 PRINT A(I);
520 NEXT I
530 END
```

Maximum and Minimum Sort

Computers: VIC-20; TRS-80 Color Computer; TRS-80 Model I; TI-99/4A

This program is also faster than the bubble sort if the numbers are mixed up quite a bit. The program goes through all the numbers and places the lowest value in the first spot of the array, and the highest value in the last spot of the array. Each pass through the numbers finds the minimum and maximum of the numbers left and places them at the endpoints.

Program 8-7. Maximum and Minimum Sort

VIC-20

```
110 DIMA(50):N=50:S=1
120 FORI=1TO50:A(I)=INT(100*RND(0)+1):PRINTA(I);:N
    EXT:PRINT:PRINT
130 L=A(S):J=S:U=L:K=S
140 FORI=S TO N
150 IFA(I)>U THENU=A(I):K=I
160 IFA(I)<L THENL=A(I):J=I
170 NEXT
180 IFJ=N THENJ=K
190 D=A(N):A(N)=A(K):A(K)=D:N=N-1
200 D=A(S):A(S)=A(J):A(J)=D:S=S+1
210 IFN>S THEN130
220 FORI=1TO50:PRINTA(I);:NEXT
230 END
```

Program 8-8. Maximum and Minimum Sort

TRS-80 Color Computer; TRS-80 Model I

```
11Ø DIM A(5Ø):N=5Ø:S=1
12Ø FORI=1TO5Ø:A(I)=RND(1ØØ):PRINTA(I);:NEXT:PRINT
    :PRINT
13Ø L=A(S):J=S:U=L:K=S
14Ø FORI=S TON
15Ø IFA(I)>U THENU=A(I):K=I
16Ø IFA(I)<L THENL=A(I):J=I
17Ø NEXT
18Ø IFJ=N THENJ=K
19Ø D=A(N):A(N)=A(K):A(K)=D:N=N-1
2ØØ D=A(S):A(S)=A(J):A(J)=D:S=S+1
21Ø IFN>S THEN13Ø
22Ø FORI=1TO5Ø:PRINTA(I);:NEXT
23Ø END
```

Program 8-9. Maximum and Minimum Sort

TI-99/4A

```
11Ø DIM A(5Ø)
12Ø N=5Ø
13Ø FOR I=1 TO 5Ø
14Ø RANDOMIZE
15Ø A(I)=INT(RND*1ØØ+1)
16Ø PRINT A(I);
17Ø NEXT I
18Ø PRINT ::
2ØØ S=1
21Ø MN=A(S)
22Ø IMIN=S
23Ø MX=MN
24Ø IMAX=S
25Ø FOR I=S TO N
26Ø IF A(I)<=MX THEN 29Ø
27Ø MX=A(I)
28Ø IMAX=I
29Ø IF A(I)>=MN THEN 32Ø
3ØØ MN=A(I)
31Ø IMIN=I
32Ø NEXT I
33Ø IF IMIN<>N THEN 35Ø
34Ø IMIN=IMAX
35Ø AA=A(N)
36Ø A(N)=A(IMAX)
37Ø A(IMAX)=AA
38Ø N=N-1
39Ø AA=A(S)
4ØØ A(S)=A(IMIN)
```

```
410 A(IMIN)=AA
420 S=S+1
430 IF N>S THEN 210
500 FOR I=1 TO 50
510 PRINT A(I);
520 NEXT I
530 END
```

Quick Sort

Computers: VIC-20; TRS-80 Color Computer; TRS-80 Model I; TI-99/4A

"Quick Sort" can be thought of as a combination of the maximum and minimum procedure and the shell sort. It is usually faster than either.

Program 8-10. Quick Sort

VIC-20

```
110 DIMA(50)
120 FORI=1TO50:A(I)=INT(100*RND(0)+1):PRINTA(I);:N
    EXT:PRINT:PRINT
130 N=50:S(1)=1:S(2)=N:T=1
140 IFT=0THEN260
150 T=T-1:I=2*T:L=S(I+1):M=S(I+2):X=A(L):J=L:K=M+1
160 K=K-1:IFK=J THEN220
170 IFX<=A(K)THEN160
180 A(J)=A(K)
190 J=J+1:IFK=J THEN220
200 IFX>=A(J)THEN190
210 A(K)=A(J):GOTO160
220 A(J)=X:IFM-J<2THEN240
230 I=2*T:S(I+1)=J+1:S(I+2)=M:T=T+1
240 IFK-L<2THEN140
250 I=2*T:S(I+1)=L:S(I+2)=K-1:T=T+1:GOTO140
260 FORI=1TON:PRINTA(I);:NEXT
270 END
```

Program 8-11. Quick Sort

TRS-80 Color Computer; TRS-80 Model I

```
110 DIM A(50)
120 FORI=1TO50:A(I)=RND(100):PRINTA(I);:NEXT:PRINT
    :PRINT
130 N=50:S(1)=1:S(2)=N:T=1
140 IFT=0THEN260
150 T=T-1:I=2*T:L=S(I+1):M=S(I+2):X=A(L):J=L:K=M+1
160 K=K-1:IFK=J THEN220
170 IFX<=A(K)THEN160
180 A(J)=A(K)
190 J=J+1:IFK=J THEN220
200 IFX>=A(J)THEN190
```

```
210 A(K)=A(J):GOTO160
220 A(J)=X:IFM-J<2THEN240
230 I=2*T:S(I+1)=J+1:S(I+2)=M:T=T+1
240 IFK-L<2THEN140
250 I=2*T:S(I+1)=L:S(I+2)=K-1:T=T+1:GOTO140
260 FORI=1TON:PRINTA(I);:NEXT
270 END
```

Program 8-12. Quick Sort

TI-99/4A

```
110 CALL CLEAR
120 DIM A(50)
130 N=50
140 FOR I=1 TO N
150 RANDOMIZE
160 A(I)=INT(RND*100)+1
170 PRINT A(I);
180 NEXT I
190 PRINT
200 S(1)=1
210 S(2)=N
220 T=1
230 IF T=0 THEN 520
240 T=T-1
250 I=2*T
260 L=S(I+1)
270 M=S(I+2)
280 X=A(L)
290 J=L
300 K=M+1
310 K=K-1
320 IF K=J THEN 400
330 IF X<=A(K)THEN 310
340 A(J)=A(K)
350 J=J+1
360 IF K=J THEN 400
370 IF X>=A(J)THEN 350
380 A(K)=A(J)
390 GOTO 310
400 A(J)=X
410 IF M-J<2 THEN 460
420 I=2*T
430 S(I+1)=J+1
440 S(I+2)=M
450 T=T+1
460 IF K-L<2 THEN 230
470 I=2*T
480 S(I+1)=L
490 S(I+2)=K-1
```

```
500 T=T+1
510 GOTO 230
520 PRINT
530 FOR I=1 TO N
540 PRINT A(I);
550 NEXT I
560 END
```

Test Scores

Computers: VIC-20; TRS-80 Color Computer; TRS-80 Model I; TI-99/4A

"Test Scores" illustrates the use of a sort routine in an actual program. Look this over to see how you could use a sort routine in a program of your own.

Students' scores are shown for three tests, along with a total and an average. The total scores are then sorted in descending order, and the total, average, and student's name are PRINTed.

The DATA statements contain these items separated by commas: last name, first name, and the scores for three tests. The computer READs the last name and first name and prints them on the screen. To save memory, the first name, a space, and the last name are combined for N$(). Next, the three scores are READ and PRINTed. The total is calculated and displayed, and the average of the three scores is shown. As the test scores are being READ, the class totals S() are also calculated. After the individual scores are shown, the average for each test and the overall average are displayed.

To prevent names from scrolling too fast on the screen, a line count L keeps track of how much is on the screen. You must press ENTER or RETURN to continue.

The bubble sort is used to arrange the names in descending order by total score. In the sort, the >= sign is used instead of the < = sign to sort from the largest to the smallest number.

Program 8-13. Test Scores

VIC-20

```
110 PRINT"{CLR}"
120 DIMN$(17),S(2),P(17)
130 L=0:S(0)=0:S(1)=0:S(2)=0:V=0
140 FORI=0TO17:READN$(I),F$:PRINTN$(I);",  ";F$:N$(
    I)=F$+" "+N$(I):P(I)=0
150 FORJ=0TO2:READT(J):P(I)=P(I)+T(J):S(J)=S(J)+T(
    J):PRINTT(J);:NEXTJ
160 A=INT(P(I)/3+.5):PRINT"{2 SPACES}";P(I);A:V=V+
    A
170 L=L+3:IFL=18THENGOSUB480
```

```
180 NEXTI
190 FORJ=ØTO2:S(J)=INT(S(J)/18+.5):NEXT
200 PRINT"{DOWN}AVERAGES":PRINTS(Ø);S(1);S(2);"
    {3 SPACES}";INT(V/18+.5):M=16
210 W=Ø:FORI=ØTOM:IFP(I)>=P(I+1)THEN24Ø
220 D=P(I):P(I)=P(I+1):P(I+1)=D:W=1:M=I
230 D$=N$(I):N$(I)=N$(I+1):N$(I+1)=D$
240 NEXTI
250 IFW=1THEN21Ø
260 GOSUB48Ø
270 FORI=ØTO17:PRINTP(I);INT(P(I)/3+.5);N$(I)
290 NEXTI:GOTO51Ø
300 DATA ADAMS,BOB,91,82,71
310 DATA BROWN,ANDY,96,84,72
320 DATA CARSON,GRANT,94,79,67
330 DATA ELLIS,CINDY,82,88,65
340 DATA HANSEN,ANGIE,96,67,72
350 DATA HANSEN,KELLY,72,97,65
360 DATA JENSEN,ED,88,71,71
370 DATA JENSEN,LENA,71,97,78
380 DATA KENT,RICHARD,77,73,78
390 DATA LARSEN,JOHN,92,82,83
400 DATA MARTIN,CHERY,88,71,97
410 DATA NELSON,RANDY,8Ø,72,99
420 DATA ROGERS,BRIAN,78,77,98
430 DATA SMITH,AURA,97,8Ø,64
440 DATA SMITH,BILL,62,67,87
450 DATA SMITH,JIM,64,81,87
460 DATA TAYLOR,JENNY,77,61,64
470 DATA WHITE,LEWIS,94,65,94
480 L=ØØ:PRINT"{DOWN}PRESS <RETURN>";
490 GETE$:IFE$<>CHR$(13)THEN49Ø
500 PRINT"{CLR}":RETURN
510 END
```

Program 8-14. Test Scores

TRS-80 Color Computer; TRS-80 Model I

```
110 CLS:CLEAR25Ø
120 DIMN$(17),S(2),P(17)
130 L=Ø:S(Ø)=Ø:S(1)=Ø:S(2)=Ø:V=Ø
140 FORI=ØTO17:READN$(I),F$:PRINTN$(I);", ";F$,:N$
    (I)=F$+" "+N$(I):P(I)=Ø
150 FORJ=ØTO2:READT(J):P(I)=P(I)+T(J):S(J)=S(J)+T(
    J):PRINTT(J);:NEXTJ
160 A=INT(P(I)/3+.5):PRINT"{5 SPACES}";P(I);A:V=V+
    A
170 L=L+2:IFL=14GOSUB48Ø
180 NEXTI
```

```
190 FORJ=ØTO2:S(J)=INT(S(J)/18+.5):NEXT
200 PRINT:PRINT"AVERAGES",S(Ø);S(1);S(2);"
    {1Ø SPACES}";INT(V/18+.5):M=16
210 W=Ø:FORI=ØTOM:IFP(I)>=P(I+1)THEN24Ø
220 D=P(I):P(I)=P(I+1):P(I+1)=D:W=1:M=I
230 D$=N$(I):N$(I)=N$(I+1):N$(I+1)=D$
240 NEXTI
250 IFW=1THEN21Ø
260 GOSUB48Ø
270 FORI=ØTO17:PRINTP(I);INT(P(I)/3+.5);N$(I)
280 L=L+1:IFL=12THENGOSUB48Ø
290 NEXTI:GOTO51Ø
300 DATA ADAMS,BOB,91,82,71
310 DATA BROWN,ANDY,96,84,72
320 DATA CARSON,GRANT,94,79,67
330 DATA ELLIS,CINDY,82,88,65
340 DATA HANSEN,ANGIE,96,67,72
350 DATA HANSEN,KELLY,72,97,65
360 DATA JENSEN,ED,88,71,71
370 DATA JENSEN,LENA,71,97,78
380 DATA KENT,RICHARD,77,73,78
390 DATA LARSEN,JOHN,92,82,83
400 DATA MARTIN,CHERY,88,71,97
410 DATA NELSON,RANDY,8Ø,72,99
420 DATA ROGERS,BRIAN,78,77,98
430 DATA SMITH,AURA,97,8Ø,64
440 DATA SMITH,BILL,62,67,87
450 DATA SMITH,JIM,64,81,87
460 DATA TAYLOR,JENNY,77,61,64
470 DATA WHITE,LEWIS,94,65,94
480 L=Ø:PRINT:PRINT"PRESS <ENTER>";
490 E$=INKEY$:IFE$<>CHR$(13)THEN49Ø
500 CLS:RETURN
510 PRINT:END
```

Program 8-15. Test Scores

TI-99/4A

```
110 CALL CLEAR
120 DIM N$(17),S(2),P(17)
130 FOR I=Ø TO 17
140 READ N$(I),F$
150 PRINT :N$(I);", ";F$
160 N$(I)=F$&" "&N$(I)
170 P(I)=Ø
180 FOR J=Ø TO 2
190 READ T(J)
200 P(I)=P(I)+T(J)
210 S(J)=S(J)+T(J)
220 PRINT T(J);
```

```
230 NEXT J
240 A=INT(P(I)/3+.5)
250 PRINT "  ";P(I);A
260 V=V+A
270 L=L+3
280 IF L<21 THEN 300
290 GOSUB 730
300 NEXT I
310 FOR J=0 TO 2
320 S(J)=INT(S(J)/18+.5)
330 NEXT J
340 PRINT :"AVERAGES"
350 PRINT S(0);S(1);S(2);"{7 SPACES}";INT(V/18+.5)
360 M=16
370 W=0
380 FOR I=0 TO M
390 IF P(I)>=P(I+1)THEN 480
400 D=P(I)
410 P(I)=P(I+1)
420 P(I+1)=D
430 W=1
440 M=I
450 D$=N$(I)
460 N$(I)=N$(I+1)
470 N$(I+1)=D$
480 NEXT I
490 IF W=1 THEN 370
500 GOSUB 730
510 FOR I=0 TO 17
520 PRINT P(I);INT(P(I)/3+.5);N$(I)
530 NEXT I
540 GOTO 790
550 DATA ADAMS,BOB,91,82,71
560 DATA BROWN,ANDY,96,84,72
570 DATA CARSON,GRANT,94,79,67
580 DATA ELLIS,CINDY,82,88,65
590 DATA HANSEN,ANGIE,96,67,72
600 DATA HANSEN,KELLY,72,97,65
610 DATA JENSEN,ED,88,71,71
620 DATA JENSEN,LENA,71,97,78
630 DATA KENT,RICHARD,77,73,78
640 DATA LARSEN,JOHN,92,82,83
650 DATA MARTIN,CHERY,88,71,97
660 DATA NELSON,RANDY,80,72,99
670 DATA ROGERS,BRIAN,78,77,98
680 DATA SMITH,AURA,97,80,64
690 DATA SMITH,BILL,62,67,87
700 DATA SMITH,JIM,64,81,87
710 DATA TAYLOR,JENNY,77,61,64
```

```
72Ø DATA WHITE,LEWIS,94,65,94
73Ø L=Ø
74Ø PRINT :"PRESS <ENTER>";
75Ø CALL KEY(Ø,K,ST)
76Ø IF K<>13 THEN 75Ø
77Ø CALL CLEAR
78Ø RETURN
79Ø END
```

Chapter 9
Fun in Less Than 4K

Chapter 9
Fun in Less Than 4K

It *is* possible to design and write games for computers with limited memory. You just have to make compromises from time to time. Nonetheless, games for such computers as the VIC, the Color Computer, and the TI-99 can be graphically entertaining and educational. Even with only 4K of memory, the game can have plenty of variety, enough to hold even the most avid player's attention.

Almost all the games in this chapter have been translated into versions for several computers. Each computer, then, has its own listing for you to look at and type in.

Some of the games, such as "Typing" or "States and Capitals," are teaching games that test your skill and knowledge. Others, such as "Defend," are simply fun to play. There are even games that are a cross between the two, such as "Multiplication Invasion," where the learning takes place in a game setting. Not only are the games fun to play, but you'll also learn something as you play.

There are even two programs, "Bake a Cake" and "Cookie File," which are really not games at all. They're still fun to type in and use, however, so I included them in this final chapter. You can use them as recipe holders, or you can alter them to include your own unusual recipes. Both have been listed only for the VIC-20, so if you have a different computer, you'll have to study the program to see if it can be translated to your machine's version of BASIC.

Typing

Computers: VIC-20; TRS-80 Color Computer; TRS-80 Model I

The letters on most microcomputer keyboards are in the same position as on a standard typewriter. With the capabilities of the computer, a learn-to-type program works very well. The graphic abilities of a computer can actually draw the hands on the screen, with the letters to be typed above the appropriate fingers.

This program teaches touch typing, using the computer, by showing the home position of the fingers — fingers on A, S, D, F, and J, K, L, and ; (: on Commodore computers). An introduction is presented, then a random letter drill. For more proficiency, there is a drill of actual words using the letters on the home keys.

Here's how the program works on the VIC-20:

Line	Function
2	PRINT title screen.
3	Turn volume on and define M for sound.

6 Define string variables to create graphics.

8–12 READ from DATA the variables position P, letter P$, graphic letter L, and sound S.

14–16 PRINT instructions.

18–22 Draw hands.

24 PRINT letters above appropriate fingers.

26–30 Wait for any key to be pressed.

32–48 Perform drill to type a letter as it appears above the finger.

50–56 Clear screen and READ phrases A$ from DATA. RESTORE is included to use the DATA again, ignoring the first 32 items.

58–80 Perform drill for phrases. A phrase is chosen randomly. If it is typed correctly, A$(J) is set equal to "" so the phrase cannot be used again. If it is typed incorrectly, the phrase may be used again. Five phrases must be typed correctly to complete the drill.

82–90 PRINT option to practice letters, practice words, or end, and branch appropriately.

92–94 Check letter typed.

96 Delay loop for music.

98–104 Wait for RETURN key to be pressed.

106 Clear screen and END.

Program 9-1. Typing

VIC-20

(In this program, make sure to use abbreviations for all BASIC keywords in the lines marked with an asterisk (*). If you don't, the program will not run correctly. Refer to Appendix C, "BASIC Keyword Abbreviations," for the proper abbreviations.)

```
*2 PRINT"{CLR}{5 DOWN}":PRINTTAB(8);"TYPING":PRINTT
   AB(8);"{2 DOWN}UNIT 1":PRINTTAB(4);"{DOWN}HOME P
   OSITION"
 4 POKE36878,15:M=36876
 6 F$="{RVS}{K} {OFF}":G$="{RVS} {L}{OFF}":H$=F
   $+" "+G$+F$:I$=G$+F$+" "+G$
 8 FORI=1TO8:READP(I),P$(I),L(I),S(I):NEXT
 10 DATA8010,A,1,195,7969,S,19,201,7949,D,4,207,797
    4,F,6,209
 12 DATA7978,J,10,215,7959,K,11,219,7983,L,12,223,8
    030,":",58,225
 14 PRINT"{3 DOWN}YOU WILL SEE A DIAGRAMOF THE HAND
    S."
 16 PRINT"PLACE YOUR FINGERS ON THE KEYS AS SHOWN."
    :GOSUB98
```

```
*18 PRINT"{CLR}{RED}{13 DOWN}":PRINTTAB(5);G$;TAB(1
    4);F$:PRINT"{2 SPACES}";H$;SPC(3);I$:PRINT"
    {2 SPACES}";H$;SPC(3);I$
 2Ø FORC=1TO4:PRINTG$+H$;SPC(3);I$+F$:NEXT
 22 PRINT"{RVS}{9 SPACES}{OFF}{3 SPACES}{RVS}
    {9 SPACES}{OFF} {RVS}{9 SPACES}{OFF}{3 SPACES}
    {RVS}{9 SPACES}{OFF}";
 24 FORC=1TO8:POKEM,S(C):POKEP(C),L(C):POKEP(C)+3Ø7
    2Ø,Ø:GOSUB96:POKEM,Ø:NEXTC
 26 PRINT"{HOME}{BLU}PLACE YOUR FINGERS IN POSITION
    ."
 28 PRINT"{GRN}PRESS ANY KEY TO":PRINT"CONTINUE."
 3Ø GETE$:IFE$=""THEN3Ø
*32 PRINT"{HOME}{BLU}TYPE EACH LETTER AS{3 SPACES}I
    T APPEARS.{4 SPACES}":PRINT"{31 SPACES}"
 34 FORI=1TO8:POKEP(I),32:NEXT
 36 FORI=1TO3:FORJ=1TO8:POKEM,S(J):POKEP(J),L(J):GO
    SUB92:NEXTJ,I
 38 FORI=1TO3Ø
 4Ø J=INT(8*RND(Ø))+1:IFJ=K THEN4Ø
 42 K=J:POKEM,S(J):POKEP(J),L(J):GOSUB92:NEXTI
 44 PRINT"{HOME}CHOOSE:{17 SPACES}1 TRY AGAIN":PRIN
    T"{2 SPACES}2 CONTINUE PROGRAM"
 46 GETE$:IFE$="1"THEN32
 48 IFE$<>"2"THEN46
 5Ø PRINT"{CLR}":RESTORE:FORI=1TO32:READE$:NEXT
 52 DATA"A SAD LAD:","A FAD:","ASK A LAD:",A SAD FA
    D,A LAD ASKS DAD
 54 DATA"ALFALFA:",ALAS A SAD DAD,"DAD ASKS A LAD:"
    ,"ASK DAD:"
 56 FORI=1TO9:READA$(I):NEXT
 58 FORI=1TO5
 6Ø J=INT(9*RND(Ø)+1):IFA$(J)=""THEN6Ø
*62 PRINT"{CLR}TYPE THE PHRASE":PRINT"THEN PRESS <R
    ETURN>{5 DOWN}":PRINTTAB(3);A$(J):PRINTTAB(3);
 64 B$="":FORK=1TO17
 66 GETE$:IFE$=""THEN66
 68 IFASC(E$)=13THEN72
 7Ø PRINTE$;:B$=B$+E$:NEXTK
 72 IFB$=A$(J)THEN76
 74 POKEM,159:GOSUB96:POKEM,135:GOSUB96:POKEM,Ø:PRI
    NT:PRINT"{DOWN}{PUR}{3 SPACES}WRONG":GOSUB98:GO
    TO6Ø
 76 FORD=1TOI:POKE793Ø+D,83:POKE793Ø+D+3Ø72Ø,2:NEXT
    D
*78 POKEM,195:GOSUB96:POKEM,2Ø7:GOSUB96:POKEM,215:G
    OSUB96:POKEM,225:GOSUB96:GOSUB96:POKEM,Ø:A$(J)=
    ""
 8Ø NEXTI
```

```
*82 PRINT:PRINT"{5 DOWN}{BLU}CHOOSE:":PRINT"
    {2 SPACES}1 PRACTICE LETTERS{4 SPACES}2 PRACTIC
    E WORDS":PRINT"{2 SPACES}3 END PROGRAM"
 84 GETE$:IFE$="1"THEN18
 86 IFE$="2"THEN5Ø
 88 IFE$="3"THEN1Ø6
 9Ø GOTO84
 92 GETE$:IFE$<>P$(J)THEN92
 94 POKEM,Ø:POKEP(J),32:RETURN
 96 FORD=1TO99:NEXT:RETURN
 98 PRINT"{DOWN}{GRN}PRESS <RETURN>{BLU}"
 1ØØ GETE$:IFE$=""THEN1ØØ
 1Ø2 IFASC(E$)<>13THEN1ØØ
 1Ø4 RETURN
 1Ø6 PRINT"{CLR}{BLU}":END
```

The program's organization is slightly different on the Color Computer:

Line	Function
1–2	READ from DATA position P, letter P$, and sound S.
3–4	Detect key pressed and compare it with correct letter.
5–7	PRINT title screen and instructions.
8–14	Draw hands with letters above appropriate fingers.
15	Clear letters.
16–24	Perform drill of typing letters — three times in order from left to right, then 30 random letters. A letter appears, and you must type it correctly to continue.
25–27	PRINT instructions, RESTORE DATA and ignore first 24 items; then READ words and phrases A$ in from DATA for next drill.
28–41	Perform drill of typing words and phrases, which are chosen randomly from A$ array. If a phrase is typed correctly, A$(L) is set equal to "" (null) so it won't be chosen again. You must type five phrases correctly to complete the drill. B$ is the phrase you type.
42–46	PRINT option for more practice or end program, branch appropriately, clear screen and END.

Program 9-2. Typing

TRS-80 Color Computer; MC-10

```
1 FORI=1TO8:READP(I),P$(I),S(I):NEXT:GOTO5
2 DATA322,A,89,261,S,1Ø8,232,D,125,267,F,133,276,J
  ,147,247,K,159,282,L,17Ø,349,;,176
3 E$=INKEY$:IFE$<>P$(J) THEN3
4 PRINT@P(J)," ";:RETURN
```

```
5 CLS:PRINT@1Ø9,"TYPING":PRINT@2Ø5,"UNIT 1":PRINT@
  265,"HOME POSITION":PRINT@352,"YOU WILL SEE A DI
  AGRAM OF THE"
6 PRINT"HANDS.  PLACE YOUR FINGERS ON{3 SPACES}THE
   KEYS AS SHOWN.":PRINT:PRINT"PRESS ANY KEY TO ST
  ART."
7 E$=INKEY$:IF E$="" THEN 7
8 CLS:PRINT"PLACE YOUR FINGERS IN POSITION.":E$=CH
  R$(191)+CHR$(191):P$=E$+" "+E$+" "+E$+"
  {8 SPACES}"+E$+" "+E$+" "+E$
9 PRINT@295,E$:PRINT@311,E$:PRINT@324,P$:PRINT@356
  ,P$:P$=E$+" "+P$+" "+E$:PRINT@385,P$:PRINT@417,P
  $:PRINT@449,P$:PRINT@461,E$;:PRINT@465,E$;
1Ø P$=E$+E$+E$+E$+E$+CHR$(191):PRINT@481,P$+" "+E$
   +"  "+E$+" "+P$;
11 FORI=1TO8:SOUNDS(I),2:PRINT@P(I),P$(I);:NEXT
12 PRINT@429,"SPACE";:PRINT@64,"PRESS ENTER TO CON
   TINUE."
13 E$=INKEY$:IF E$="" THEN 13
14 IF ASC(E$)<>13 THEN 13
15 FORI=1TO8:PRINT@P(I)," ";:NEXT
16 PRINT @64,"TYPE THE LETTER AS IT APPEARS."
17 FORI=1TO3:FORJ=1TO8:SOUNDS(J),2:PRINT@P(J),P$(J
   );:GOSUB3:NEXTJ,I
18 FORI=1TO3Ø
19 J=RND(8):IFJ=L THEN19
2Ø SOUNDS(J),2:PRINT@P(J),P$(J);:GOSUB3:L=J:NEXT
21 PRINT@Ø,"PRESS 1 TO TRY AGAIN{1Ø SPACES}":PRINT@
   38,"2 TO CONTINUE":PRINT"{3Ø SPACES}"
22 E$=INKEY$:IFE$="2"THEN25
23 IFE$<>"1"THEN22
24 PRINT@Ø,"{2Ø SPACES}":PRINT@38,"{13 SPACES}":GOTO
   16
25 CLS:PRINT"NOW TRY TYPING WORDS.":PRINT"TYPE THE
    PHRASE SHOWN":PRINT"THEN PRESS ENTER.":RESTORE
   :FORI=1TO24:READE$:NEXT
26 DATA A SAD LAD;,A FAD;,ASK A LAD;,A SAD FAD,A L
   AD ASKS DAD;,ASK DAD;,ALFALFA;,ALAS A SAD DAD;,
   DAD ASKS A LAD;
27 FORI=1TO9:READA$(I):NEXT
28 FORI=1TO5
29 L=RND(9):IFA$(L)=""THEN29
3Ø B$="":PRINT@298,A$(L):SOUND227,1
31 FORJ=1TO2Ø
32 E$=INKEY$:IFE$=""THEN32
33 IFASC(E$)=13THEN35
34 PRINT@329+J,E$:B$=B$+E$:NEXTJ
35 IFB$=A$(L) THEN4Ø
36 SOUND125,2:SOUND89,2:PRINT@416,"WRONG":PRINT@44
   8,"PRESS ENTER"
```

```
37 E$=INKEY$:IFE$=""THEN37
38 IFASC(E$)<>13THEN37
39 PRINT@416,"{5 SPACES}":PRINT@448,"{11 SPACES}":I
   =I-1:GOTO41
4Ø SOUND89,1:SOUND125,1:SOUND147,1:SOUND176,2:A$(L
   )=""
41 PRINT@298,"{15 SPACES}":PRINT@33Ø,"{2Ø SPACES}":N
   EXTI
42 CLS:PRINT"PRESS 1 MORE PRACTICE ON LETTERS
   {6 SPACES}2 MORE PRACTICE ON PHRASES{6 SPACES}3
    END PROGRAM"
43 E$=INKEY$:IFE$="1"THEN8
44 IFE$="2"THEN25
45 IFE$<>"3"THEN43
46 CLS:END
```

"Typing" works this way on the TRS-80 Model I:

Line	Function
10	Branch past subroutines.
20	SET points for graphics.
100–170	Define print position P and letter P$ to print the letter above the finger.
200–210	Check letter typed. If it's correct, it's erased.
300–360	PRINT title screen and instructions.
490–730	Draw hands on screen.
750–770	Label fingers with letters.
780–840	When ENTER is pressed, the labels are cleared.
845–970	Perform drill to type letters. Letters are presented in order from left to right three times, then randomly for 30 letters. The letter must be typed correctly to continue.
980–1000	PRINT instructions for phrase drill.
1010–1030	READ words and phrases from DATA for A$.
1040–1220	Perform drill typing phrases. If a phrase has been typed correctly, A$(L) is set equal to "" (null) so it won't be chosen again. Five phrases must be typed correctly to complete the drill. B$ is what you enter.
1230–1300	PRINT option to practice more or end program and branch appropriately.
1310	Clear screen and END.

Program 9-3. Typing

TRS-80 Model I

```
1Ø GOTO3ØØ
2Ø FORJ=BTO41:FORI=A TOA+6:SET(I,J):NEXTI,J:RETURN
```

```
100 P=642:P$="A":PRINT@P,P$;:RETURN
110 P=519:P$="S":PRINT@P,P$;:RETURN
120 P=460:P$="D":PRINT@P,P$;:RETURN
130 P=529:P$="F":PRINT@P,P$;:RETURN
140 P=558:P$="J":PRINT@P,P$;:RETURN
150 P=499:P$="K":PRINT@P,P$;:RETURN
160 P=568:P$="L":PRINT@P,P$;:RETURN
170 P=701:P$=";":PRINT@P,P$;:RETURN
200 E$=INKEY$:IFE$<>P$ THEN200
210 PRINT@P," ";:RETURN
300 CLS
310 PRINT@218,"T Y P I N G"
320 PRINT@469,"UNIT 1:  HOME POSITION"
330 PRINT@768,"YOU WILL SEE A DIAGRAM OF THE HANDS
    ."
340 PRINT"PLACE YOUR FINGERS ON THE 'HOME' KEYS."
350 PRINT:PRINT"PRESS ANY KEY TO START."
360 E$=INKEY$:IFE$=""THEN360
490 CLS
500 PRINT"PLACE YOUR HANDS IN THIS POSITION ON THE
     KEYS."
510 A=2:B=33:GOSUB20
520 A=12:B=27:GOSUB20
530 A=22:B=24:GOSUB20
540 A=32:B=27:GOSUB20
550 FORJ=42TO47
560 FORI=2TO38:SET(I,J):NEXTI,J
590 FORJ=45TO47:FORI=42TO50
600 SET(I,J):NEXTI,J
610 FORJ=42TO44:FORI=44TO50
620 SET(I,J):NEXTI,J
630 A=120:B=33:GOSUB20
640 A=110:B=27:GOSUB20
650 A=100:B=24:GOSUB20
660 A=90:B=27:GOSUB20
670 FORJ=42TO47:FORI=90TO126:SET(I,J):NEXTI,J
710 FORJ=45TO47:FORI=78TO86:SET(I,J):NEXTI,J
730 FORJ=42TO44:FORI=78TO84:SET(I,J):NEXTI,J
750 GOSUB100:GOSUB110:GOSUB120:GOSUB130
760 GOSUB140:GOSUB150:GOSUB160:GOSUB170
770 PRINT@860,"SPACE BAR";
780 PRINT@64,"PRESS ENTER TO CONTINUE."
790 E$=INKEY$:IFE$=""THEN790
800 IFASC(E$)<>13THEN790
810 PRINT@0,"{46 SPACES}"
830 P=642:GOSUB210:P=519:GOSUB210:P=460:GOSUB210:P
    =529:GOSUB210
840 P=701:GOSUB210:P=568:GOSUB210:P=499:GOSUB210:P
    =558:GOSUB210
845 PRINT@0,"TYPE EACH LETTER AS IT APPEARS."
```

```
846 PRINT"{18 SPACES}"
85Ø FORI=1TO3
86Ø GOSUB1ØØ:GOSUB2ØØ:GOSUB11Ø:GOSUB2ØØ:GOSUB12Ø:G
    OSUB2ØØ:GOSUB13Ø:GOSUB2ØØ
87Ø GOSUB14Ø:GOSUB2ØØ:GOSUB15Ø:GOSUB2ØØ:GOSUB16Ø:G
    OSUB2ØØ:GOSUB17Ø:GOSUB2ØØ
88Ø NEXTI
89Ø FORI=1TO3Ø
9ØØ L=RND(8):IFL=L1 THEN9ØØ
91Ø ONL GOSUB1ØØ,11Ø,12Ø,13Ø,14Ø,15Ø,16Ø,17Ø
92Ø GOSUB2ØØ:L1=L
93Ø NEXTI
94Ø PRINT@Ø,"PRESS 1 TO TRY AGAIN{14 SPACES}"
95Ø PRINT"{6 SPACES}2 TO CONTINUE"
96Ø E$=INKEY$:IFE$="1"THEN845
97Ø IFE$<>"2"THEN96Ø
98Ø CLS
99Ø PRINT"NOW TRY TYPING WORDS."
1ØØØ PRINT"TYPE THE PHRASE SHOWN THEN PRESS ENTER.
     "
1Ø1Ø RESTORE
1Ø2Ø DATA A SAD LAD;,A FAD;,ASK A LAD;,A SAD FAD,A
      LAD ASKS DAD;,ASK DAD;,ALFALFA;,ALAS A SAD D
     AD;,DAD ASKS A LAD;
1Ø3Ø FORI=1TO9:READA$(I):NEXTI
1Ø4Ø FORI=1TO5
1Ø5Ø L=RND(9):IFA$(L)=""THEN1Ø5Ø
1Ø55 B$=""
1Ø6Ø PRINT@47Ø,A$(L)
1Ø7Ø FORJ=1TO2Ø
1Ø8Ø E$=INKEY$:IFE$=""THEN1Ø8Ø
1Ø85 IFASC(E$)=13THEN112Ø
1Ø9Ø PRINT@533+J,E$
11ØØ B$=B$+E$
111Ø NEXTJ
112Ø IFB$=A$(L) THEN1195
113Ø PRINT@832,"WRONG"
114Ø PRINT@896,"PRESS ENTER."
115Ø E$=INKEY$:IFE$=""THEN115Ø
116Ø IFASC(E$)<>13THEN115Ø
117Ø PRINT@832,"{5 SPACES}"
118Ø PRINT@896,"{12 SPACES}"
1185 I=I-1
119Ø GOTO12ØØ
1195 A$(L)=""
12ØØ PRINT@47Ø,"{15 SPACES}"
121Ø PRINT@534,"{2Ø SPACES}"
122Ø NEXTI
123Ø CLS
```

```
1240 PRINT"PRESS 1 MORE PRACTICE ON LETTERS"
1250 PRINTTAB(6);"2 MORE PRACTICE ON PHRASES"
1260 PRINTTAB(6);"3 END PROGRAM"
1270 E$=INKEY$
1280 IFE$="1"THEN490
1290 IFE$="2"THEN980
1300 IFE$<>"3"THEN1270
1310 CLS:END
```

Teeth Wisdom

Computers: VIC-20; TI-99/4A

"Teeth Wisdom" teaches the names of the teeth — central incisors, lateral incisors, cuspids, bicuspids, and molars. The computer draws the teeth on the screen with the labels. After you've studied the names, press ENTER or RETURN to clear the labels. A quiz is given by blinking certain teeth. To answer, you press the number corresponding to the name.

Teeth Wisdom works like this on the VIC:

Line	Function
5–6	PRINT title screen.
7	READ names of teeth in N$ array and create delay.
8	Define B$ and C$ strings for use in clearing labels.
9	Turn on volume and set S equal to a speaker value for sounds.
10–12	PRINT instruction screen.
15	Change screen color to pink.
16–20	Draw and label teeth.
22	Wait for RETURN to be pressed before continuing.
30–32	Clear labels.
35	PRINT "NAME THE TEETH".
40	Set elements of W$ array equal to element of N$ array.
42–48	Randomly list names of teeth.
50–66	Perform quiz.
52–54	Randomly choose teeth.
56–58	Blink appropriate teeth while waiting for response.
60–62	If answer is incorrect, play sound and return for another response.
65–66	If answer is correct, play appropriate music, set A(I)=0 so tooth will not be chosen again, and go to next problem.
70–72	PRINT option to try again and branch appropriately.

74–76 If answer is Y for *yes,* clear previous list of answers and branch to beginning of quiz.

80–89 Blink teeth by POKEing memory locations with color numbers.

92 Delay loop for sounds.

94–97 Wait for RETURN key to be pressed before continuing program.

99 DATA containing names of teeth.

100 Clear screen, return to normal screen and border colors, and END.

Program 9-4. Teeth Wisdom

VIC-20

(In this program, make sure to use abbreviations for all BASIC keywords in the lines marked with an asterisk (*). If you don't, the program will not run correctly. Refer to Appendix C, "BASIC Keyword Abbreviations," for the proper abbreviations.)

```
 5 PRINT"{CLR}{BLU}{4 DOWN}{2 SPACES}**************
   **":PRINT"{SHIFT-SPACE} *";SPC(14);"*":PRINT"
   {2 SPACES}* TEETH WISDOM *"
 6 PRINT"{2 SPACES}*";SPC(14);"*":PRINT"{2 SPACES}*
   ***************"
 7 FORC=1TO5:READN$(C):NEXT:FORC=1TO900:NEXT
 8 B$="{16 SPACES}":C$="{10 SPACES}"
 9 POKE36878,15:S=36876
 10 PRINT"{CLR}{2 DOWN}YOU WILL SEE A DIAGRAMOF THE
     TEETH WITH THE NAMES OF THE TEETH."
 11 PRINT"{DOWN}WHEN YOU KNOW THE{5 SPACES}NAMES, P
    RESS <RETURN>."
*12 PRINT"{DOWN}THE LABELS WILL CLEAR AND YOU WILL
    {SPACE}BE GIVEN A QUIZ.":PRINT"{3 DOWN}{GRN}PRE
    SS RETURN{BLU}":GOSUB94
 15 POKE36879,169
*16 PRINT"{CLR}{DOWN}":PRINTTAB(4);"CENTRAL INCISOR
    S":PRINTTAB(4);"T":PRINTTAB(4);"T{A}LATERAL I
    NCISORS{2 SPACES}{WHT}ZQQZ"
 18 PRINT" Z{4 SPACES}Z{BLU}**CUSPIDS":PRINT"
    {WHT}A";SPC(6);"A {BLU}I":PRINT"{WHT}A";SPC(6);
    "A {BLU}KBICUSPIDS"
 20 PRINT"{WHT}X";SPC(6);"X {BLU}I":PRINT"{WHT}X";S
    PC(6);"X {BLU}{Q}*MOLARS":PRINT"{WHT}X";SPC(
    6);"X {BLU}K"
 22 PRINT"{3 DOWN}{GRN}PRESS RETURN":GOSUB94
*30 PRINT"{UP}";B$:PRINT"{HOME}{2 DOWN}{4 SPACES}";
    B$:PRINT"{5 SPACES}":PRINT"{6 SPACES}";B$:PRINT
    TAB(7);C$:PRINTTAB(9);" ":PRINTTAB(9);C$
 32 PRINTTAB(9);" ":PRINTTAB(9);C$:PRINTTAB(9);" "
 35 PRINT"{DOWN}{BLU}NAME THE TEETH{DOWN}"
 40 FORC=1TO5:W$(C)=N$(C):NEXT
 42 FORC=1TO5
 44 I=INT(5*RND(0)+1)
 46 IFW$(I)=""THEN44
 48 PRINTC;W$(I):A(I)=C:W$(I)="":NEXTC
 50 FORC=1TO5
 52 I=INT(5*RND(0)+1)
 54 IFA(I)=0THEN52
 56 ON I GOSUB80,82,84,86,88
 58 GETA$:IFA$=""THEN56
 60 IF ASC(A$)=A(I)+48THEN 65
 62 POKES,159:GOSUB92:POKES,135:GOSUB92:POKES,0:GOT
    O56
*65 POKES,195:GOSUB92:POKES,207:GOSUB92:POKES,215:G
    OSUB92:POKES,225:GOSUB92:GOSUB92:POKES,0:A(I)=0
 66 NEXTC
 70 PRINT"{DOWN}{GRN}TRY AGAIN? (Y/N){BLU}";
 72 GETA$:IFA$="N"THEN100
 74 IFA$<>"Y"THEN72
 76 PRINT"{7 UP}{BLU}":FORC=1TO7:PRINTB$;"
    {3 SPACES}":NEXT:PRINT"{8 UP}":GOTO40
 80 POKE38513,2:POKE38514,2:POKE38513,1:POKE38514,1
    :RETURN
 82 POKE38512,2:POKE38515,2:POKE38512,1:POKE38515,1
    :RETURN
 84 POKE38533,2:POKE38538,2:POKE38533,1:POKE38538,1
    :RETURN
*86 POKE38554,2:POKE38561,2:POKE38576,2:POKE38583,2
    :POKE38554,1:POKE38561,1:POKE38576,1:POKE38583,
    1:RETURN
 88 POKE38598,2:POKE38605,2:POKE38620,2:POKE38627,2
    :POKE38642,2:POKE38649,2
 89 POKE38598,1:POKE38605,1:POKE38620,1:POKE38627,1
    :POKE38642,1:POKE38649,1:RETURN
 92 FORD=1TO150:NEXT:RETURN
 94 FORD=1TO10:GETA$:NEXT
 95 GETA$:IFA$=""THEN95
 96 IF ASC(A$)<>13THEN95
 97 RETURN
 99 DATACENTRAL INCISORS,LATERAL INCISORS,CUSPIDS,B
    ICUSPIDS,MOLARS
 100 PRINT"{CLR}":POKE36879,27:END
```

Here's the explanation for the TI version:

Line	Function
110–170	Clear screen and PRINT title screen.
180–230	Define graphics characters and blink asterisks on screen.
240–330	DATA containing graphics definitions.

340–360 Define color sets 9 through 13 as white on light red for teeth.

370–390 Define color sets for light red on transparent for graphics surrounding teeth.

400–510 Clear screen, PRINT instructions, and define strings as groups of characters for later printing.

520–560 READ in names of five groups of teeth as N$ array and set the W$ array elements equal to the N$ array elements.

570 PRINT message to press ENTER and wait for response.

580–690 Clear screen and PRINT teeth with labels.

700 PRINT message to press ENTER and wait for response.

710–760 Clear message and labels.

770 PRINT quiz title.

780–850 Randomly PRINT names of teeth on screen from the W$ array. A(I) will be the correct corresponding answer.

860–1060 Perform quiz.

870–880 Randomly choose teeth.

890–920 Blink teeth blue and white while waiting for response.

930–940 If number 1–5 is pressed, show which number was pressed; otherwise, return to line 890.

950–990 If answer is incorrect, play sound effect and return for another response.

1000–1030 If answer is correct, play music.

1040 Clear answer chosen.

1050–1060 Set element A to zero so tooth will not be chosen again and return for next problem.

1070–1100 PRINT option to try again, wait for response, and branch appropriately.

1110–1140 If you want to try again, set W$ array elements equal to names of teeth and branch to beginning of exercise.

1150 Stop.

1160–1190 PRINT "PRESS ENTER" and wait for response.

1200–1210 Clear screen and END.

Program 9-5. Teeth Wisdom

TI-99/4A

```
110 CALL CLEAR
120 PRINT TAB(4);"******************"
130 PRINT TAB(4);"*";TAB(21);"*"
140 PRINT TAB(4);"*  TEETH WISDOM   *"
150 PRINT TAB(4);"*";TAB(21);"*"
160 PRINT TAB(4);"******************":::::::
170 PRINT:::
180 FOR C=94 TO 157
190 CALL COLOR(2,13,1)
200 READ C$
210 CALL CHAR(C,C$)
220 CALL COLOR(2,16,1)
230 NEXT C
240 DATA ØØØØØØ1F1Ø1Ø1Ø1,ØØØØØØFF
250 DATA ØØØØØF1F3F7F7F7F,ØØØØ83C7C7E7EFEF,ØØØØEØF
    ØF8FCFEFE,FFFFFFFFFFFFFFFF,EFEFEFEFEFEFEFEF,Ø
    ,,""
260 DATA 7ØF8FCFCFEFEFEFF,FEF8C,ØE1F3F3F7F7F7FFF,7
    F1FØ3,,,,,ØØØØØØØØØØØØØØF8,Ø1Ø1Ø1Ø1Ø1Ø1,FCFEFF
    FFFEFCF83
270 DATA ØØØØØØØØØØØØØØ1F,3F7FFFFF7F3F1FØC,8Ø8Ø8Ø8
    Ø8Ø8,,,ØF1F1F1F1F1FØE,EØF8F8F8F8F8F,Ø71F1F1F
    1FØF
280 DATA FØF8F8F8F8F87,1F3F3F3F3F1F,FØF8F8F8F8F8
    F,ØF1F1F1F1FØF,F8FCFCFCFCF8,7FFFFFFFFFFF7F
290 DATA EØFØFØFØFØFØE,Ø7ØFØFØFØFØ7,FEFFFFFFFFFF
    FE,Ø3Ø3Ø3Ø3Ø3Ø3Ø1,FFFFFFFFFFFFFFFF,CØCØCØCØC
    ØCØCØ8,""
300 DATA FFFFFCFØEØCØ8,FFFF7E181,FFFF3FØFØ7Ø3Ø1,FF
    FEFEFCFCF8F8F8,FF7F7F3F3F1F1F1F,FØFØFØFØFØFØ
    F
310 DATA ØFØFØFØFØFØFØF,EØEØEØCØCØ8Ø8,Ø7Ø7Ø7Ø7
    Ø3Ø3Ø1Ø1,ØØØØØØØØØØØ71FFF,ØØØØØ31FFFFFFFFF,ØØF
    FFFFFFFFFFFFF
320 DATA ØØØØCØF8FFFFFFFF,ØØØØØØØØØØ7ØF8FF,Ø1Ø7ØF1
    F3F3F7FFF,8ØEØFØF8FCFCFEFF,Ø1Ø1Ø3Ø3Ø3Ø7Ø7ØF,8Ø
    8ØCØCØCØEØEØF
330 DATA ØFØF1F1F1F3F3F3F,FØFØF8F8F8FCFCFC,3F7F7F7
    F7F7F7F7F7F,FCFEFEFEFEFEFEFE
340 FOR C=9 TO 13
350 CALL COLOR(C,16,1Ø)
360 NEXT C
370 CALL COLOR(14,1Ø,1)
380 CALL COLOR(15,1Ø,1)
390 CALL COLOR(16,1Ø,1)
400 CALL CLEAR
410 CALL COLOR(2,2,1)
420 PRINT "YOU WILL SEE A DIAGRAM OF"
```

```
43Ø PRINT :"THE TEETH WITH THE NAMES"
44Ø PRINT :"OF THE TEETH."
45Ø A$=CHR$(128)&CHR$(129)
46Ø B$=CHR$(13Ø)&CHR$(131)
47Ø PRINT ::"WHEN YOU KNOW THE NAMES,"
48Ø PRINT :"PRESS <ENTER>."
49Ø D$=CHR$(132)&CHR$(133)&CHR$(134)
5ØØ PRINT ::"THE LABELS WILL CLEAR AND"
51Ø PRINT :"YOU WILL BE GIVEN A QUIZ."::::
52Ø FOR C=1 TO 5
53Ø READ N$(C)
54Ø W$(C)=N$(C)
55Ø NEXT C
56Ø DATA CENTRAL INCISORS,LATERAL INCISORS,CUSPIDS
    ,BICUSPIDS,MOLARS
57Ø GOSUB 116Ø
58Ø CALL CLEAR
59Ø PRINT TAB(8);"^_CENTRAL INCISORS"
6ØØ PRINT TAB(5);CHR$(145)&CHR$(146)&CHR$(147)&CHR
    $(148)&CHR$(149)
61Ø PRINT TAB(4);CHR$(15Ø);"e`abe";CHR$(151);"__LA
    TERAL INCISORS"
62Ø PRINT "  ";CHR$(15Ø);"phcdcjs";CHR$(151)
63Ø PRINT " ";CHR$(152);"qrieeektu";CHR$(153);"CUS
    PIDS"
64Ø PRINT " ";CHR$(154);"xyeeeeez{";CHR$(155);"BIC
    USPIDS"
65Ø PRINT " ";CHR$(156);"|{,}e";CHR$(136)&CHR$(137
    )&CHR$(138);"e{,}";CHR$(127)&CHR$(157)
66Ø PRINT " e";A$;CHR$(139);"   ";CHR$(14Ø);B$;"e"
    ;"MOLARS"
67Ø PRINT " e";A$;CHR$(141);"   ";CHR$(142);B$;"e"
68Ø PRINT " ";D$;CHR$(143);"   ";CHR$(144);D$
69Ø PRINT " eee     eee":::
7ØØ GOSUB 116Ø
71Ø CALL HCHAR(23,16,32,13)
72Ø CALL HCHAR(1Ø,1Ø,32,18)
73Ø CALL HCHAR(12,13,32,18)
74Ø CALL HCHAR(14,15,32,7)
75Ø CALL HCHAR(15,15,32,9)
76Ø CALL HCHAR(17,15,32,6)
77Ø PRINT TAB(8);"NAME THE TEETH"::
78Ø FOR C=1 TO 5
79Ø RANDOMIZE
8ØØ I=INT(5*RND+1)
81Ø IF W$(I)="" THEN 8ØØ
82Ø PRINT TAB(9);C;W$(I)
83Ø A(I)=C
84Ø W$(I)=""
```

```
850 NEXT C
860 FOR C=1 TO 5
870 I=INT(5*RND+1)
880 IF A(I)=0 THEN 870
890 CALL KEY(0,K,S)
900 CALL COLOR(I+8,6,10)
910 CALL COLOR(I+8,16,10)
920 IF S<1 THEN 890
930 IF (K<49)+(K>53)THEN 890
940 CALL HCHAR(18+K-48,11,62)
950 IF K-48=A(I)THEN 1000
960 CALL SOUND(150,330,0)
970 CALL SOUND(150,262,0)
980 CALL VCHAR(19,11,32,5)
990 GOTO 890
1000 CALL SOUND(150,262,0)
1010 CALL SOUND(150,330,0)
1020 CALL SOUND(150,392,0)
1030 CALL SOUND(200,523,0)
1040 CALL VCHAR(19,11,32,5)
1050 A(I)=0
1060 NEXT C
1070 PRINT ::"TRY AGAIN? (Y/N)"
1080 CALL KEY(0,K,S)
1090 IF K=78 THEN 1200
1100 IF K<>89 THEN 1080
1110 FOR C=1 TO 5
1120 W$(C)=N$(C)
1130 NEXT C
1140 GOTO 580
1150 STOP
1160 PRINT TAB(14);"PRESS <ENTER>"
1170 CALL KEY(0,K,S)
1180 IF K<>13 THEN 1170
1190 RETURN
1200 CALL CLEAR
1210 END
```

In this program, you'll see spaces between quotation marks, such as " ". Type in the quote marks and press the space bar the correct number of times. If it's hard to tell how many spaces to enter, you can use the line above as a reference. For example, in line 690, you can tell that five spaces are needed between the e's by counting the characters in line 680 directly above the gap.

Bake a Cake

Computer: VIC-20

Baking a cake, but you don't have a recipe? Here's a program for the VIC-20 that will help. The program contains recipes for several cakes. You can select the recipe, see it on the screen, and even enlarge or reduce it. If you would like to enlarge the recipe enter a number such as 3 to triple the recipe. To reduce it, enter a decimal fraction, such as .5 to halve the recipe. The converted recipe will then appear on the screen.

You don't know what to make? Select the ingredient list. As the inventory is listed, press *Y* for yes if you have the ingredient or *N* if you don't. After the inventory list is complete, the computer will tell you which recipes you can make, given that inventory.

For convenience in programming, the amounts in the recipes are given in decimals. For example, ⅔ cup sugar is written as .67 c. sugar.

The DATA statements in lines 101-107 contain the measure, then the ingredient for 25 ingredients. Line 3 READs A$, the measure, and B$(N,0), the name of the ingredient, then assigns I$(N) equal to the measure plus a space plus the ingredient name. B$ is used in the inventory list, and I$ is used in printing the recipe. Z is the number of ingredients, minus one (because the subscripts start with the number zero).

As you are typing the DATA statements, you may notice two or more commas together with nothing between them (,,,). Be sure to get the right number of commas as you are typing. The commas indicate a null string, or a string variable equal to “”.

The recipes are in the DATA statements in lines 109-123. The first item is the name of the cake. The next items are the amounts of the ingredients in the following order:

cups shortening
cups flour
cups sugar
cups brown sugar
tsp. baking powder
tsp. salt
tsp. soda
cups cherry juice
number of cherries
cups bananas (mashed)
cups sauerkraut
cups milk
cups buttermilk
number of eggs
number of egg whites
tsp. red food coloring
ounces chocolate
tbsp. cocoa
tsp. vanilla
tsp. cinnamon
tsp. nutmeg
tsp. vinegar
cups salad oil
cups water
cups oatmeal

The DATA will contain a null string if the cake does not contain that ingredient.

An example is the first part of line 109, the data for Banana Cake:
DATA BANANA,.67,2.5,1.67,,1,1,1,,,1,,,.67,2,,,,,,,,,,,,
The name of the cake is BANANA. The recipe is .67 cup shortening (two-thirds), 2.5 cups flour, 1.67 cup sugar, 1 tsp. baking powder, 1 tsp. salt, 1 tsp. soda, 1 cup mashed bananas, .67 cup buttermilk, and 2 eggs.

To print a recipe, the data is RESTOREd first, then the first 50 ingredients and measures are read and ignored (line 73). The key you press to choose a recipe is E$; the ASCII code will be 65 for A, 66 for B, and so on. Lines 75-77 figure out which recipe was chosen and read through the title and ingredients to get to the appropriate recipe. Line 79 PRINTs the title of the cake. Line 81 READs the amount from the DATA statement. If the amount is a null or zero, that ingredient is not PRINTed. However, if there *is* a value, the value is shown. The corresponding measure and name of the ingredient are READ from the I$ array.

Line 83 defines variables M(I) for measure and C$(I) for ingredient for only those ingredients in the recipe. These values are used in printing the converted recipe, lines 97-99.

For the inventory list, the computer keeps track of your Y or N answers in the BR(N,1) array, where N varies from 0 to 24 for the ingredients. Line 33 checks to see if an N is stored as a no answer for flour, sugar, or salt. If any one of these three items has a *no*, no cakes can be baked. Y is a variable for the number of Y answers; if there are not enough ingredients with a Y answer, you cannot make a cake (checked in line 35).

Line 47 RESTOREs the DATA, then ignores the first 50 items. Lines 49-51 check through the recipe for each cake. If there is an amount listed for an ingredient, the corresponding B$(N,1) value is checked. If it is N for no, you are lacking one of the ingredients required for the cake. The rest of the ingredients are skipped over, and the computer goes to the next recipe. If each of the required ingredients also has a B$ value of Y for yes, the cake can be made and the name of the cake is printed.

These are real recipes that really work. No baking instructions are given because there isn't room in the VIC version, but usually the person baking knows how to bake the cake and just needs to be reminded of the amount of each ingredient. In case you want to try these recipes, all of these recipes are for cakes to be baked in two 9-inch layers. Mix the shortening with the sugars, add the dry ingredients alternately with the liquid, and then add the eggs and vanilla. Bake at 350° to 375°.

In the cherry cake, cut up the cherries before adding to the mixture. For the oatmeal cake, first boil the water, and then add the oatmeal. Let that mixture cool while you mix everything else — add the oatmeal mixture last. The "wacky" cake you can mix all at once — just dump everything into one big bowl and mix it up. For the red velvet cake, combine the ingredients as usual, except for the vinegar and soda. Mix the vinegar and soda together and fold into the rest of the batter. The sauerkraut is a moist chocolate cake — just don't tell anyone what it is until after they've eaten it. Rinse the sauerkraut well, then chop it into small pieces before combining it with the rest of the cake batter.

This program comes very close to filling all the available memory. As with other VIC-20 programs in this book, be sure to use the abbreviations for all the BASIC words, such as ? for PRINT and D SHIFT A for DATA in the lines marked with the asterisk.

Be sure you copy the DATA statements exactly for the correct recipes. There are no spaces except in the names of the cakes. If there are commas together, do not put spaces between the commas.

Program 9-6. Bake a Cake

VIC-20

(In this program, make sure to use abbreviations for all BASIC keywords in the lines marked with an asterisk (*). If you don't, the program will not run correctly. Refer to Appendix C, "BASIC Keyword Abbreviations," for the proper abbreviations.)

```
1 DIMI$(24),B$(24,1),M(11),C$(11):Z=24
3 FORN=ØTOZ:READA$,B$(N,Ø):I$(N)=A$+" "+B$(N,Ø):NE
  XT
5 PRINT"{CLR}{BLU}{5 DOWN}{5 RIGHT}BAKE A CAKE"
7 PRINT"{2 DOWN}CHOOSE:":PRINT"{DOWN} 1 NEED TO KN
  OW":PRINT"{3 SPACES}WHAT CAN BE MADE."
9 PRINT"{DOWN} 2 WANT TO SEE":PRINT"{3 SPACES}A CE
  RTAIN RECIPE":PRINT"{DOWN} 3 END PROGRAM"
11 GETE$:IFE$="3"THEN2ØØ
13 IFE$="2"THEN61
15 IFE$<>"1"THEN11
17 PRINT"{CLR}{BLU}IN THE FOLLOWING LIST PRESS 'Y'
    IF YOU HAVE THE INGREDIENT,"
19 PRINT"'N' IF YOU DO NOT.{4 SPACES}PRESS 'S' TO
   {SPACE}START{4 SPACES}OVER.{2 DOWN}"
21 Y=Ø:FORN=ØTOZ:PRINTB$(N,Ø);"{2 SPACES}{RED}--
   {BLK}";:GOSUB15Ø
23 GETE$:IFE$="S"THEN17
25 IFE$="N"THENPRINT"N{BLU}":GOTO31
27 IFE$<>"Y"THEN23
29 PRINT"Y{BLU}":Y=Y+1
31 B$(N,1)=E$:NEXTN:C=Ø:PRINT"{2 DOWN}YOU CAN MAKE
   :"
```

```
 33 IFB$(1,1)="N"ORB$(2,1)="N"ORB$(5,1)="N"THEN37
 35 IFY>7THEN47
 37 PRINT"NOTHING TODAY.":PRINT"YOU NEED MORE SUPPL
    IES"
 39 PRINT"{2 DOWN}{GRN}PRESS RETURN{BLU}";:GOSUB15Ø
 41 GETE$:IFE$=""THEN41
 43 IF ASC(E$)<>13THEN41
 45 GOTO5
 47 RESTORE:FORN=ØTOZ*2+1:READE$:NEXT:READA$
 49 FORN=ØTOZ:READE$:IFE$=""ORE$="Ø"THEN53
 51 IFB$(N,1)="N"THENFORI=N+1TOZ:READE$:NEXTI:GOTO5
    5
 53 NEXTN:PRINTA$;" CAKE":C=C+1
 55 READA$:IFA$<>"Z"THEN49
 57 IFC=ØTHEN37
 59 PRINT"{DOWN}GO AHEAD AND BAKE!":GOTO39
*61 PRINT"{CLR}{BLU}{DOWN}CHOOSE:{2 DOWN}":PRINT"A
    {SPACE}BANANA CAKE":PRINT"B CHERRY CAKE":PRINT"
    C CHOCOLATE CAKE"
 63 PRINT"D DEVIL'S FOOD CAKE{3 SPACES}E GOLD LAYER
     CAKE":PRINT"F OATMEAL CAKE"
 65 PRINT"G RED VELVET CAKE":PRINT"H SAUERKRAUT CAK
    E":PRINT"I SPICE CAKE"
 67 PRINT"J TWO-EGG CAKE":PRINT"K WACKY CAKE":PRINT
    "L WHITE CAKE":GOSUB15Ø
 69 GETE$:IFE$=""THEN69
 71 A=ASC(E$):IFA<65ORA>76THEN69
 73 RESTORE:PRINT"{CLR}{BLU}";:FORN=ØTO2*Z+1:READE$
    :NEXT
 75 IFA=65THEN79
 77 FORN=1TOA-65:READA$:FORI=ØTOZ:READE$:NEXTI,N
 79 READA$:PRINTA$;" CAKE{DOWN}":I=Ø
 81 FORN=ØTOZ:READE$:IFE$=""ORVAL(E$)=ØTHEN85
 83 M(I)=VAL(E$):C$(I)=I$(N):PRINTM(I);TAB(6);C$(I)
    :I=I+1
 85 NEXTN
 87 PRINT"{2 DOWN}{RED}CONVERT RECIPE? (Y/N){BLU}"
 89 GETE$:IFE$="N"THEN39
 91 IFE$<>"Y"THEN89
 93 PRINT"{DOWN}MULTIPLY BY WHAT":PRINT"NUMBER OR D
    ECIMAL?":PRINT"{RED} --{BLU}";
 95 INPUTF:IFF<=ØTHENPRINT"SORRY, F>Ø":GOTO93
 97 F=INT(F*1ØØ)/1ØØ:PRINT"{CLR}{BLU}";F;"TIMES ORI
    GINAL{DOWN}":PRINTA$;" CAKE{DOWN}"
 99 FORN=ØTOI-1:PRINTF*M(N);TAB(6);C$(N):NEXT:GOTO8
    7
1Ø1 DATAC.,SHORTENING,C.,FLOUR,C.,SUGAR,C.,BROWN S
     UGAR,TSP.,BAKING PDR,TSP.
1Ø3 DATASALT,TSP.,SODA,C.,CHERRY JUICE,,CHERRIES,C
     .,BANANAS,C.,SAUERKRAUT,C.,MILK
 1Ø5 DATAC.,BUTTERMILK,,EGGS,,EGG WHITES,TSP.,RED C
     OLOR,ØZ.,CHOCOLATE,TBSP.,COCOA,TSP.
 1Ø7 DATAVANILLA,TSP.,CINNAMON,TSP.,NUTMEG,TSP.,VIN
     EGAR,C.,SALAD OIL,C.,WATER,C.,OATMEAL
*1Ø9 DATABANANA,.67,2.5,1.67,,1,1,1,,,1,,,.67,2,,,,
     ,,,,,,,,CHERRY,.5,2.25,1.33,,3,.5,,.25
 111 DATA16,,,.5,,,4,,,,,,,,,,,,CHOCOLATE,.67,2.5,1.
     75,,,.5,1,,,,,,,,2,,,2,,1,,,,,1.25,Ø
*113 DATADEVIL'S FOOD,.67,2.25,2,,1,1,1,,,,,1.25,,3
     ,,1,3,,,,,,,,,,GOLD LAYER,.5,2.25,1.5,,3
*115 DATA1,,,,,,1.67,,2,,,,,1.5,,,,,,,,OATMEAL,.5,1.
     5,1,1,,.5,1,,,,,,,,2,,,,,1,.75,.25,,,1.25
*117 DATA1,RED VELVET,.5,2.75,1.5,,,.5,1.5,,,,,,,1,2
     ,,6,,2,1,,,1,,,,SAUERKRAUT,.67,2.25,1.25
*119 DATAØ,1,.25,,,,,.67,,,3,,,,8,1,,,,,1.25,,SPICE
     ,.75,2.25,1,,1,1,1,,,,,,1,3,,,,,,1,.5,,Ø
*121 DATAØ,,TWO-EGG,.5,2.25,1.5,,2.5,1,,,,,,1,,2,,,
     ,,1,,,,,,,,WACKY,,2.5,1.5,,,1,1,,,,,,,,,Ø
 123 DATAØ,,6,1,,,1.5,.75,1.5,,WHITE,.75,2.25,1.5,,
     3,1,,,,,,,1,,,5,,,,1.5,,,,,,,,Z
 15Ø FORI=ØTO9:GETE$:NEXTI:RETURN
 2ØØ PRINT"{CLR}{BLU}":END
```

Cookie File

Computer: VIC-20

The DATA method used in "Bake a Cake" can be used for a variety of things — recipes of any kind, craft instructions, and even a type of word processor. "Cookie File" illustrates how the program can be adapted for cookie recipes.

The general baking instructions are as follows:

- Almond cookies — roll into balls, flatten slightly, and place blanched almond on top. Brush with egg if desired.
- Ball cookies — drop cookies onto sheet, then flatten with ice cube or moist cloth; sprinkle colored cake decors on top; and bake just until golden brown around the edges.
- Brownies — melt the cocoa with the shortening first and bake in square pan.
- Butterscotch bars — melt shortening (or butter) with brown sugar; cool, then add other ingredients; and bake in rectangular glass baking dish.
- Chocolate chip cookies — make as drop cookies.
- Chocolate drop cookies — make as drop cookies.
- Mexican Wedding Cookies — roll into balls, bake about 20 minutes, and roll in powdered sugar while still warm, then again when cool.

- Oatmeal chocolate chips — make as drop cookies.
- Oatmeal crisps (refrigerator cookie) — form into long roll and slice, then bake.
- Snickerdoodles — roll dough into balls, and then roll in cinnamon and sugar mixture before baking.
- Toffee bars — press into 9 x 13 pan or on cookie sheet (about ½ inch thick).

Program 9-7. Cookie File

VIC-20

(In this program, make sure to use abbreviations for all BASIC keywords in the lines marked with an asterisk (*). If you don't, the program will not run correctly. Refer to Appendix C, "BASIC Keyword Abbreviations," for the proper abbreviations.)

```
 1 DIMG$(17),V$(17,1),M(9),F$(9):POKE36879,30
 2 FORI=0TO17:READA$,V$(I,0):G$(I)=A$+" "+V$(I,0):N
   EXT
*3 PRINT"{CLR}{DOWN}{5 SPACES}COOKIE FILE":PRINT"
   {DOWN}CHOOSE{2 DOWN}":PRINT" 1 NEED TO KNOW":PRI
   NT"{3 SPACES}WHAT CAN BE MADE"
 4 PRINT"{2 DOWN} 2 WANT TO SEE RECIPE":PRINT"
   {2 DOWN} 3 END PROGRAM"
 5 GETR$:IFR$="3"THEN62
 6 IFR$="2"THEN33
 7 IFR$<>"1"THEN5
*8 PRINT"{CLR}{BLU}IN THE FOLLOWING LIST PRESS 'Y'
   {SPACE}IF YOU HAVE THE INGREDIENT":PRINT"'N' IF
   {SPACE}YOU DO NOT."
 9 PRINT"PRESS 'S' TO START{4 SPACES}OVER.{2 DOWN}"
   :YS=0:FORK=0TO17:PRINTV$(K,0);"{2 SPACES}{RED}
   {2 P}{BLK}";
 10 GETR$:IFR$=""THEN10
 12 IFR$="S"THEN8
 13 IFR$="N"THENPRINT"N":GOTO16
 14 IFR$<>"Y"THEN10
 15 PRINT"Y":YS=YS+1
 16 V$(K,1)=R$:PRINT"{BLU}";:NEXT:C=0:PRINT"
    {3 DOWN}YOU CAN MAKE:"
 17 IFV$(0,1)="N"ORV$(6,1)="N"THEN19
 18 IFYS>4THEN24
 19 PRINT"NOTHING TODAY.":PRINT"YOU NEED MORE SUPPL
    IES"
 20 PRINT"{3 DOWN}{GRN}PRESS RETURN.{BLU}"
 21 GETR$:IFR$=""THEN21
 22 IFASC(R$)<>13THEN21
 23 GOTO3
 24 RESTORE:FORJ=1TO36:READB$:NEXT:READA$
 25 FORJ=0TO17:READB$:IFB$=""THEN27
 26 IFV$(J,1)="N"THEN28
 27 NEXT:PRINTA$:C=C+1
 28 READA$:IFA$="ZZZ"THEN31
 29 IFLEN(A$)<6THEN28
 3Ø GOTO25
 31 IFC=ØTHEN19
 32 PRINT"{DOWN}GO AHEAD AND BAKE!":GOTO2Ø
 33 PRINT"{CLR}CHOOSE:":PRINT"{DOWN}A ALMOND COOKIE
    S{6 SPACES}B BALL COOKIES":PRINT"C BROWNIES"
 34 PRINT"D BUTTERSCOTCH BARS{3 SPACES}E CHOCO CHIP
     COOKIES{2 SPACES}F CHOCO DROP COOKIES"
*35 PRINT"G MEX WEDDING COOKIES H OATMEAL CHOCO CHI
    PS I OATMEAL CRISPS":PRINT"J SNICKERDOODLES"
 36 PRINT"K SUGAR COOKIES":PRINT"L TOFFEE BARS"
 37 GETR$:IFR$=""THEN37
 38 A=ASC(R$):IFA<65ORA>76THEN37
 39 RESTORE:PRINT"{CLR}{DOWN}":FORI=1TO36:READB$:NE
    XT:IFA=65THEN41
 4Ø FORI=1TOA-65:READA$:FORK=ØTO18:READR$:NEXTK,I
 41 READA$:PRINTA$:K=Ø:PRINT
 42 FORJ=ØTO17:READB$:IFB$=""ORVAL(B$)=ØTHEN44
 43 M(K)=VAL(B$):F$(K)=G$(J):PRINTM(K);TAB(6);F$(K)
    :K=K+1
 44 NEXTJ:READT:PRINT"{DOWN}BAKE AT";T
 45 PRINT"{DOWN}{RED}CONVERT RECIPE? (Y/N)"
 46 GETR$:IFR$="N"THEN2Ø
 47 IFR$<>"Y"THEN46
 48 PRINT"{DOWN}MULTIPLY BY WHAT":PRINT"NUMBER OR D
    ECIMAL?":PRINT"⁅2 P⁆{BLU}";
 49 INPUTF:IFF<=ØTHENPRINT"SORRY, F>Ø":GOTO48
 5Ø F=(INT(F*1ØØ)/1ØØ):PRINT"{CLR}":PRINTF;"TIMES O
    RIGINAL":PRINT:PRINTA$:PRINT
 51 FORI=ØTOK-1:PRINTF*M(I);F$(I):NEXT:GOTO45
 52Ø DATAC.,SHORTENING,C.,SUGAR,C.,BROWN SUGAR,C.
 525 DATAPDRD SUGAR,,EGGS,TSP.,VANILLA,C.,FLOUR,TSP
 53Ø DATABAKING PDR,TSP.,BAKING SODA,TSP.,SALT,TSP.
     ,CINNAMON,TBSP.,COCOA,TSP.
 535 DATAALMOND EXT,C.
 54Ø DATAMILK,C.,OATMEAL,OZ.,CHOCO CHIPS,DOZ.,ALMON
     DS,TSP.
 545 DATACAKE DECORS,ALMOND COOKIES,2,2,,,2
 55Ø DATA"",4,2,,,,,2,,,,4,,375,BALL COOKIES,.5,.33
     ,,,1,.5,.75,,,,,,,,,,,,2,375
 555 DATABROWNIES,.5
 56Ø DATA1,,,2,1,.75,.5,,.5,,6,,,,,,,,35Ø
 565 DATABUTTERSCOTCH BARS,.5,,2,,2,1,1.75,2,,.25,,
     ,,,,,,,35Ø
 57Ø DATACHOCO CHIP COOKIES,.5,.25,.5,,1,.5,1,,.5,.
     5,,,,,,,6,,,375
 575 DATACHOCO DROP COOKIES,.5,,1,Ø
 58Ø DATAØ,1,1.67,,.5,.5,,6,,.5,,,,,35Ø
```

```
585 DATAMEXICAN WEDDING,.75,,,.67,,1,1.5,,,.25,1,,
    ,,.75,,,Ø
59Ø DATA325
595 DATAOATMEAL CHOCO,1,1,.5,,2,1,2,,1,1,,,,,2,6,,
    ,35Ø,OATMEAL CRISPS,1,1,1,,2,1,1.5
6ØØ DATAØ,1,1,,,,,3,,,,35Ø,SNICKERDOODLES,1,1.5,,,
    2,,2.75,3,,.5,,,,,,,,,4ØØ
6Ø5 DATASUGAR COOKIES
61Ø DATA.67,.75,,,1,.5,2,1.5,,.25,,,,.25,,,,,375
615 DATATOFFEE BARS,1,1,,,,1,2,,,,,,,,,,6,,,35Ø,ZZZ
62Ø PRINT"{CLR}":END
```

States and Capitals

Computers: VIC-20; MC-10; TRS-80 Model I; TI-99/4A

Can you name the capital of each state? This program gives you practice in just that. States are chosen in a random order. You have two chances to get the capital city correct, and spelling counts. If the city is missed twice, you'll see the right answer, and that state will appear again. The drill continues until all 50 states' capitals have been correctly entered. The number of total guesses is then shown. The closer your score is to 50, the better your knowledge is of states and their capitals.

All three versions work about the same.

Program control first branches past subroutines. PRESS RETURN or PRESS ENTER will display, and the computer will wait for the response. The VIC-20 version has a delay subroutine and also initializes sound parameters.

The screen clears and the title is printed. The DIMension statement allows for state names S$ and capitals C$ in elements zero to 49. The states and corresponding capitals are READ in from DATA. Next the instructions are printed.

G, the number of guesses, is initialized to zero. N is the number of states that have had the capitals named correctly. F is a flag which ordinarily is zero but is 1 if the capital has been missed once. After two misses, the capital is given and that state remains in the list. B$ is the string variable that keeps the letters the user has typed in. If the back arrow is pressed to erase, B$ is reduced and a space is printed. After the capital has been named correctly, S$(R) is set equal to null, "", so that state will not be chosen again.

The last three lines print the total number of guesses and end the program.

Program 9-8. States and Capitals

VIC-20

(In this program, make sure to use abbreviations for all BASIC keywords in the lines marked with an asterisk (*). If you don't, the program will not run correctly. Refer to Appendix C, "BASIC Keyword Abbreviations," for the proper abbreviations.)

```
2 T=36876:POKE36878,15:GOTO14
4 PRINT"{DOWN}{GRN}PRESS <RETURN>"
6 GETE$:IFE$=""THEN6
8 IFASC(E$)<>13THEN6
1Ø PRINT"{CLR}{BLU}":RETURN
12 FORD=1TO1ØØ:NEXT:RETURN
14 PRINT"{CLR}{DOWN} STATES AND CAPITALS"
16 DIMS$(49),C$(49)
18 FORI=ØTO49:READS$(I),C$(I):NEXT
2Ø DATAALABAMA,MONTGOMERY,ALASKA,JUNEAU,ARIZONA,PH
   OENIX,ARKANSAS,LITTLE ROCK,CALIFORNIA
22 DATASACRAMENTO,COLORADO,DENVER,CONNECTICUT,HART
   FORD
23 DATADELAWARE,DOVER,FLORIDA,TALLAHASSEE
24 DATAGEORGIA,ATLANTA,HAWAII,HONOLULU,IDAHO,BOISE
   ,ILLINOIS,SPRINGFIELD,INDIANA
26 DATAINDIANAPOLIS,IOWA,DES MOINES,KANSAS,TOPEKA
27 DATAKENTUCKY,FRANKFORT,LOUISIANA,BATON ROUGE
28 DATAMAINE,AUGUSTA,MARYLAND,ANNAPOLIS,MASSACHUSE
   TTS,BOSTON,MICHIGAN,LANSING,MINNESOTA
3Ø DATAST. PAUL,MISSISSIPPI,JACKSON,MISSOURI,JEFFE
   RSON CITY,MONTANA,HELENA,NEBRASKA
32 DATALINCOLN,NEVADA,CARSON CITY,NEW HAMPSHIRE,CO
   NCORD,NEW JERSEY,TRENTON,NEW MEXICO
34 DATASANTA FE,NEW YORK,ALBANY,NORTH CAROLINA,RAL
   EIGH
35 DATANORTH DAKOTA,BISMARCK,OHIO,COLUMBUS
36 DATAOKLAHOMA,OKLAHOMA CITY,OREGON,SALEM,PENNSYL
   VANIA,HARRISBURG
37 DATARHODE ISLAND,PROVIDENCE
38 DATASOUTH CAROLINA,COLUMBIA,SOUTH DAKOTA,PIERRE
39 DATATENNESSEE,NASHVILLE,TEXAS,AUSTIN,UTAH
4Ø DATASALT LAKE CITY,VERMONT,MONTPELIER,VIRGINIA,
   RICHMOND,WASHINGTON,OLYMPIA
42 DATAWEST VIRGINIA,CHARLESTON,WISCONSIN,MADISON,
   WYOMING,CHEYENNE
44 PRINT"{DOWN}TYPE IN THE NAME OF{3 SPACES}THE CA
   PITAL AS THE{4 SPACES}STATE APPEARS."
46 PRINT"{DOWN}CITIES MUST BE SPELLEDCORRECTLY TO
   {SPACE}BE":PRINT"COUNTED AS CORRECT.":GOSUB4
48 G=Ø:M=7879
5Ø FORN=1TO5Ø
52 F=Ø:R=INT(5Ø*RND(Ø)):IFS$(R)=""THEN52
54 PRINT"{CLR}";N:PRINT"{3 DOWN}STATE: ";S$(R)
```

```
56 PRINT"{DOWN}CAPITAL: ":B$="":POKET,225:GOSUB12:
   POKET,Ø:G=G+1
58 FORI=1TO2Ø:POKE3Ø72Ø+M+I,Ø
6Ø GETE$:IFE$=""THEN6Ø
62 IFASC(E$)=13THEN8Ø
64 IFASC(E$)<>2ØTHEN7Ø
66 I=I-1:IFI=ØTHENI=1:GOTO6Ø
68 B$=LEFT$(B$,LEN(B$)-1):POKEM+I,32:GOTO6Ø
7Ø IFASC(E$)=32THENE=32:GOTO78
72 IFASC(E$)=46THENE=46:GOTO78
74 IFASC(E$)<65OR ASC(E$)>9ØTHEN6Ø
76 E=ASC(E$)-64
78 POKEM+I,E:B$=B$+E$:NEXTI
8Ø IFB$=C$(R)THEN88
82 POKET,159:GOSUB12:POKET,135:GOSUB12:POKET,Ø
84 F=F+1:IFF=1THENPRINT"{3 UP}":FORJ=M TOM+2Ø:POKE
   J,32:NEXTJ:GOTO56
86 PRINT"{4 DOWN}{2 SPACES}"C$(R):GOSUB4:N=N-1:GOT
   O94
88 PRINT"{5 DOWN}{RED}CORRECT!{BLU}"
9Ø POKET,195:GOSUB12:POKET,2Ø7:GOSUB12:POKET,215:G
   OSUB12:POKET,225:GOSUB12:POKET,Ø
92 S$(R)=""
94 NEXTN:PRINT"{CLR}{DOWN}YOU ANSWERED ALL 5Ø."
96 PRINT"{DOWN}NUMBER OF GUESSES:";G:PRINT
98 END
```

Program 9-9. States and Capitals

MC-10

```
1 GOTO6
2 PRINT@495,"PRESS <ENTER>";
3 E$=INKEY$:IFE$=""THEN3
4 IFASC(E$)<>13THEN3
5 CLS:RETURN
6 CLS:PRINT@165,"**STATES AND CAPITALS**
7 DIMS$(49),C$(49)
8 FORI=ØTO49:READS$(I),C$(I):NEXT
9 DATAALABAMA,MONTGOMERY,ALASKA,JUNEAU,ARIZONA,PHO
  ENIX,ARKANSAS,LITTLE ROCK,CALIFORNIA,SACRAMENTO,
  COLORADO,DENVER
1Ø DATACONNECTICUT,HARTFORD,DELAWARE,DOVER,FLORIDA
   ,TALLAHASSEE,GEORGIA,ATLANTA,HAWAII,HONOLULU,ID
   AHO,BOISE,ILLINOIS,SPRINGFIELD
11 DATAINDIANA,INDIANAPOLIS,IOWA,DES MOINES,KANSAS
   ,TOPEKA,KENTUCKY,FRANKFORT,LOUISIANA,BATON ROUG
   E,MAINE,AUGUSTA,MARYLAND
12 DATAANNAPOLIS,MASSACHUSETTS,BOSTON,MICHIGAN,LAN
   SING,MINNESOTA,ST. PAUL,MISSISSIPPI,JACKSON,MIS
   SOURI,JEFFERSON CITY,MONTANA
```

```
13 DATAHELENA,NEBRASKA,LINCOLN,NEVADA,CARSON CITY,
   NEW HAMPSHIRE,CONCORD,NEW JERSEY,TRENTON,NEW ME
   XICO,SANTA FE,NEW YORK
14 DATAALBANY,NORTH CAROLINA,RALEIGH,NORTH DAKOTA,
   BISMARCK,OHIO,COLUMBUS,OKLAHOMA,OKLAHOMA CITY,O
   REGON,SALEM,PENNSYLVANIA
15 DATAHARRISBURG,RHODE ISLAND,PROVIDENCE,SOUTH CA
   ROLINA,COLUMBIA,SOUTH DAKOTA,PIERRE,TENNESSEE,N
   ASHVILLE,TEXAS,AUSTIN,UTAH
16 DATASALT LAKE CITY,VERMONT,MONTPELIER,VIRGINIA,
   RICHMOND,WASHINGTON,OLYMPIA,WEST VIRGINIA,CHARL
   ESTON
17 DATAWISCONSIN,MADISON,WYOMING,CHEYENNE
18 PRINT@288,"TYPE IN THE NAME OF THE CAPITAL AS T
   HE STATE APPEARS."
19 PRINT"CITIES MUST BE SPELLED CORRECTLYTO BE COU
   NTED AS CORRECT.":GOSUB2
2Ø G=Ø
21 FORN=1TO5Ø
22 F=Ø:R=RND(5Ø)-1:IFS$(R)=""THEN22
23 PRINT@28,N
24 PRINT@16Ø,"STATE:   ";S$(R)
25 PRINT@256,"CAPITAL:":B$="":SOUND227,1:G=G+1
26 FORI=1TO2Ø
27 E$=INKEY$:IFE$=""THEN27
28 IFASC(E$)=13THEN33
29 IFASC(E$)<>8THEN32
3Ø I=I-1:IFI=ØTHENI=1:GOTO27
31 B$=LEFT$(B$,LEN(B$)-1):PRINT@265+I," ";:GOTO27
32 PRINT@265+I,E$;:B$=B$+E$:NEXT
33 IFB$=C$(R) THEN37
34 SOUND125,2:SOUND89,2:PRINT@266,"{2Ø SPACES}"
35 F=F+1:IFF=1THEN25
36 PRINT@266,C$(R):GOSUB2:N=N-1:GOTO39
37 PRINT@394,"CORRECT!"
38 SOUND89,1:SOUND125,1:SOUND147,1:SOUND176,3:S$(R
   )=""
39 CLS:NEXTN
4Ø CLS:PRINT@128,"YOU ANSWERED ALL 5Ø."
41 PRINT@224,"NUMBER OF GUESSES: ";G:PRINT@448
42 END
```

Program 9-10. States and Capitals

TRS-80 Model I

```
1 GOTO6
2 PRINT@99Ø,"PRESS <ENTER>";
3 E$=INKEY$:IFE$=""THEN3
4 IFASC(E$)<>13THEN3
5 CLS:RETURN
```

```
6 CLS:PRINT@2Ø5,"**STATES AND CAPITALS**
7 DIMS$(49),C$(49)
8 FORI=ØTO49:READS$(I),C$(I):NEXT
9 DATAALABAMA,MONTGOMERY,ALASKA,JUNEAU,ARIZONA,PHO
  ENIX,ARKANSAS,LITTLE ROCK,CALIFORNIA,SACRAMENTO,
  COLORADO,DENVER
1Ø DATACONNECTICUT,HARTFORD,DELAWARE,DOVER,FLORIDA
   ,TALLAHASSEE,GEORGIA,ATLANTA,HAWAII,HONOLULU,ID
   AHO,BOISE,ILLINOIS,SPRINGFIELD
11 DATAINDIANA,INDIANAPOLIS,IOWA,DES MOINES,KANSAS
   ,TOPEKA,KENTUCKY,FRANKFORT,LOUISIANA,BATON ROUG
   E,MAINE,AUGUSTA,MARYLAND
12 DATAANNAPOLIS,MASSACHUSETTS,BOSTON,MICHIGAN,LAN
   SING,MINNESOTA,ST. PAUL,MISSISSIPPI,JACKSON,MIS
   SOURI,JEFFERSON CITY,MONTANA
13 DATAHELENA,NEBRASKA,LINCOLN,NEVADA,CARSON CITY,
   NEW HAMPSHIRE,CONCORD,NEW JERSEY,TRENTON,NEW ME
   XICO,SANTA FE,NEW YORK
14 DATAALBANY,NORTH CAROLINA,RALEIGH,NORTH DAKOTA,
   BISMARCK,OHIO,COLUMBUS,OKLAHOMA,OKLAHOMA CITY,O
   REGON,SALEM,PENNSYLVANIA
15 DATAHARRISBURG,RHODE ISLAND,PROVIDENCE,SOUTH CA
   ROLINA,COLUMBIA,SOUTH DAKOTA,PIERRE,TENNESSEE,N
   ASHVILLE,TEXAS,AUSTIN,UTAH
16 DATASALT LAKE CITY,VERMONT,MONTPELIER,VIRGINIA,
   RICHMOND,WASHINGTON,OLYMPIA,WEST VIRGINIA,CHARL
   ESTON
17 DATAWISCONSIN,MADISON,WYOMING,CHEYENNE
18 PRINT@512,"TYPE IN THE NAME OF THE CAPITAL AS T
   HE STATE APPEARS."
19 PRINT"CITIES MUST BE SPELLED CORRECTLYTO BE COU
   NTED AS CORRECT.":GOSUB2
2Ø G=Ø
21 FORN=1TO5Ø
22 F=Ø:R=RND(5Ø)-1:IFS$(R)=""THEN22
23 PRINT@56,N
24 PRINT@32Ø,"STATE:  ";S$(R)
25 PRINT@512,"CAPITAL:":B$="":G=G+1:L=521
26 FORI=1TO2Ø
27 E$=INKEY$:IFE$=""THEN27
28 IFASC(E$)=13THEN33
29 IFASC(E$)<>8THEN32
3Ø I=I-1:IFI=ØTHENI=1:GOTO27
31 B$=LEFT$(B$,LEN(B$)-1):PRINT@L+I," ";:GOTO27
32 PRINT@L+I,E$;:B$=B$+E$:NEXT
33 IFB$=C$(R) THEN37
34 PRINT@788,"INCORRECT, TRY AGAIN":FORD=1TO5ØØ:NE
   XT:PRINT@788,"{2Ø SPACES}":PRINT@L,"{21 SPACES}"
35 F=F+1:IFF=1THEN25
```

```
36 PRINT@532,C$(R):GOSUB2:N=N-1:GOTO39
37 PRINT@788,"CORRECT!"
38 FORD=1TO5ØØ:NEXT:S$(R)=""
39 CLS:NEXTN
4Ø CLS:PRINT@256,"YOU ANSWERED ALL 5Ø."
41 PRINT@448,"NUMBER OF GUESSES: ";G:PRINT@896,""
42 END
```

Program 9-11. States and Capitals

TI-99/4A

```
1ØØ GOTO 16Ø
11Ø PRINT :::"PRESS <ENTER>";
12Ø CALL KEY(Ø,K,S)
13Ø IF K<>13 THEN 12Ø
14Ø CALL CLEAR
15Ø RETURN
16Ø CALL CLEAR
17Ø PRINT "** STATES AND CAPITALS **"::::
18Ø PRINT "TYPE IN THE NAME OF THE"
19Ø PRINT "CAPITAL AS THE STATE NAME"
2ØØ PRINT "APPEARS."
21Ø PRINT :"CITIES MUST BE SPELLED":"CORRECTLY TO
    BE COUNTED":"AS CORRECT."
22Ø PRINT :"PRESS LEFT ARROW TO ERASE."::
23Ø DIM S$(49),C$(49)
24Ø FOR I=Ø TO 49
25Ø READ S$(I),C$(I)
26Ø NEXT I
27Ø DATA ALABAMA,MONTGOMERY,ALASKA,JUNEAU,ARIZONA,
    PHOENIX,ARKANSAS,LITTLE ROCK,CALIFORNIA,SACRAM
    ENTO,COLORADO
28Ø DATA DENVER,CONNECTICUT,HARTFORD,DELAWARE,DOVE
    R,FLORIDA,TALLAHASSEE,GEORGIA,ATLANTA,HAWAII,H
    ONOLULU,IDAHO
29Ø DATA BOISE,ILLINOIS,SPRINGFIELD,INDIANA,INDIAN
    APOLIS,IOWA,DES MOINES,KANSAS,TOPEKA,KENTUCKY,
    FRANKFORT
3ØØ DATA LOUISIANA,BATON ROUGE,MAINE,AUGUSTA,MARYL
    AND,ANNAPOLIS,MASSACHUSETTS,BOSTON,MICHIGAN,LA
    NSING
31Ø DATA MINNESOTA,ST. PAUL,MISSISSIPPI,JACKSON,MI
    SSOURI,JEFFERSON CITY,MONTANA,HELENA,NEBRASKA,
    LINCOLN
32Ø DATA NEVADA,CARSON CITY,NEW HAMPSHIRE,CONCORD,
    NEW JERSEY,TRENTON,NEW MEXICO,SANTA FE,NEW YOR
    K,ALBANY
33Ø DATA NORTH CAROLINA,RALEIGH,NORTH DAKOTA,BISMA
    RCK,OHIO,COLUMBUS,OKLAHOMA,OKLAHOMA CITY,OREGO
    N,SALEM
```

```
340 DATA PENNSYLVANIA,HARRISBURG,RHODE ISLAND,PROV
    IDENCE,SOUTH CAROLINA,COLUMBIA,SOUTH DAKOTA,PI
    ERRE
350 DATA TENNESSEE,NASHVILLE,TEXAS,AUSTIN,UTAH,SAL
    T LAKE CITY,VERMONT,MONTPELIER,VIRGINIA,RICHMO
    ND
360 DATA WASHINGTON,OLYMPIA,WEST VIRGINIA,CHARLEST
    ON,WISCONSIN,MADISON,WYOMING,CHEYENNE
370 G=Ø
380 P=19
390 GOSUB 11Ø
400 FOR N=1 TO 5Ø
410 CALL CLEAR
420 F=Ø
430 RANDOMIZE
440 R=INT(5Ø*RND)
450 IF S$(R)="" THEN 44Ø
460 PRINT N
470 PRINT ::::"STATE:  ";S$(R)
480 PRINT ::"CAPITAL: ":::::
490 B$=""
500 G=G+1
510 CALL SOUND(1ØØ,1497,2)
520 FOR I=1 TO 2Ø
530 CALL KEY(Ø,K,S)
540 IF K=13 THEN 67Ø
550 IF S<1 THEN 53Ø
560 IF K<>8 THEN 64Ø
570 I=I-1
580 IF I>Ø THEN 61Ø
590 I=1
600 GOTO 53Ø
610 B$=SEG$(B$,1,LEN(B$)-1)
620 CALL HCHAR(P,11+I,32)
630 GOTO 53Ø
640 CALL HCHAR(P,11+I,K)
650 B$=B$&CHR$(K)
660 NEXT I
670 IF B$=C$(R)THEN 78Ø
680 CALL SOUND(1ØØ,33Ø,4)
690 CALL SOUND(1ØØ,262,4)
700 CALL HCHAR(P,12,32,2Ø)
710 F=F+1
720 IF F=1 THEN 49Ø
730 FOR I=1 TO LEN(C$(R))
740 CALL HCHAR(P,11+I,ASC(SEG$(C$(R),I,1)))
750 NEXT I
760 GOSUB 11Ø
770 GOTO 41Ø
```

```
780 PRINT "CORRECT!"
790 CALL SOUND(100,262,2)
800 CALL SOUND(100,330,2)
810 CALL SOUND(100,392,2)
820 CALL SOUND(150,523,2)
830 S$(R)=""
840 NEXT N
850 CALL CLEAR
860 PRINT "YOU ANSWERED ALL 50."
870 PRINT ::::"NUMBER OF GUESSES: ";G::::
880 END
```

South America

Computers: VIC-20; TRS-80 Color Computer; TRS-80 Model I; TI-99/4A

Although shorter than "States and Capitals," this program, which is a drill for the capitals of the countries of South America, is similar. A list of the capital cities is printed on the screen. The computer randomly chooses a country and prints the country, and you then press the letter corresponding to the capital city of that country.

All the versions work like this:

The countries and capitals are READ in from DATA, alphabetically by capital. The country is S$ and the capital is C$. After a country has been chosen, S$() is set equal to "" so it will not be chosen again.

(Note: The names and spellings of the countries and capitals change periodically. You may need to adjust this list to correspond with your sources of information.)

Program 9-12. South America

VIC-20

```
110 DIMS$(11),C$(11)
120 PRINT"{CLR}** SOUTH AMERICA **"
130 PRINT"{DOWN}MATCH THE COUNTRY WITHTHE CAPITAL
    {SPACE}CITY.":PRINT"PRESS THE LETTER."
140 FORI=0TO11:READS$(I),C$(I):NEXT
150 DATACOLOMBIA,BOGOTA,BRAZIL,BRASILIA,ARGENTINA,
    BUENOS AIRES,VENEZUELA,CARACAS
155 DATAFRENCH GUIANA,CAYENNE,GUYANA,GEORGETOWN,PE
    RU,LIMA,URUGUAY,MONTEVIDEO
160 DATASURINAME,PARAMARIBO,ECUADOR,QUITO,CHILE,SA
    NTIAGO,BOLIVIA,SUCRE
170 PRINT"{3 DOWN}PRESS A KEY TO START."
180 GETA$:IFA$=""THEN180
190 S=0
200 FORT=1TO12:PRINT"{CLR}"
210 R=INT(12*RND(0)):IFS$(R)=""THEN210
220 FORI=0TO11:PRINTCHR$(65+I);" ";C$(I):NEXTI:PRI
    NT
```

```
230 PRINTS$(R);"{3 SPACES}??? ";
240 GETA$:IFA$=""THEN24Ø
250 IFASC(A$)<65THEN24Ø
260 IFASC(A$)>76THEN24Ø
270 PRINTA$:IFASC(A$)-65=R THEN29Ø
280 PRINT"{DOWN}THE CAPITAL IS":PRINTC$(R):GOTO3ØØ
290 PRINT"CORRECT!":S=S+1
300 S$(R)="":PRINT"{3 SPACES}PRESS <RETURN>";
310 GETA$:IFA$<>CHR$(13)THEN31Ø
320 NEXTT
330 PRINT"{CLR}{DOWN}SOUTH AMERICA":PRINT"{DOWN}YO
    UR SCORE WAS ";S:PRINT"OUT OF 12 COUNTRIES
    {DOWN}"
340 END
```

Program 9-13. South America

TRS-80 Color Computer; TRS-80 Model I

```
11Ø DIMS$(11),C$(11)
12Ø CLS:PRINTTAB(5);"** SOUTH AMERICA **"
13Ø PRINT:PRINT"MATCH THE COUNTRY WITH THE
    {6 SPACES}CAPITAL CITY.":PRINT"PRESS THE CORRE
    CT LETTER."
14Ø FORI=ØTO11:READS$(I),C$(I):NEXT
15Ø DATACOLOMBIA,BOGOTA,BRAZIL,BRASILIA,ARGENTINA,
    BUENOS AIRES,VENEZUELA,CARACAS,FRENCH GUIANA,C
    AYENNE,GUYANA,GEORGETOWN
16Ø DATAPERU,LIMA,URUGUAY,MONTEVIDEO,SURINAME,PARA
    MARIBO,ECUADOR,QUITO,CHILE,SANTIAGO,BOLIVIA,SU
    CRE
17Ø PRINT:PRINT:PRINT"PRESS ANY KEY TO START."
18Ø A$=INKEY$:IFA$=""THEN18Ø
19Ø S=Ø
2ØØ FORT=1TO12:CLS
21Ø R=RND(12)-1:IFS$(R)=""THEN21Ø
22Ø FORI=ØTO11:PRINTTAB(4);CHR$(65+I);" ";C$(I):NE
    XTI:PRINT
23Ø PRINTS$(R);"{3 SPACES}??? ";
24Ø A$=INKEY$:IFA$=""THEN24Ø
25Ø IFASC(A$)<65THEN24Ø
26Ø IFASC(A$)>76THEN24Ø
27Ø PRINTA$;:IFASC(A$)-65=R THEN29Ø
28Ø PRINT:PRINT"THE CAPITAL IS ";C$(R):GOTO3ØØ
29Ø PRINT"  CORRECT!":S=S+1
3ØØ S$(R)="":PRINT"{5 SPACES}PRESS <ENTER>";
31Ø A$=INKEY$:IFA$<>CHR$(13)THEN31Ø
32Ø NEXTT
33Ø CLS:PRINT"SOUTH AMERICA":PRINT:PRINT"YOUR SCOR
    E WAS ";S:PRINT"OUT OF 12 COUNTRIES.":PRINT
34Ø END
```

Program 9-14. South America

TI-99/4A

```
11Ø DIM S$(11),C$(11)
12Ø CALL CLEAR
13Ø PRINT TAB(4);"** SOUTH AMERICA **"
14Ø PRINT ::"MATCH THE COUNTRY WITH THE"::"CAPITAL
     CITY."::"PRESS THE CORRECT LETTER."::::
15Ø FOR I=Ø TO 11
16Ø READ S$(I),C$(I)
17Ø NEXT I
18Ø DATA COLOMBIA,BOGOTA,BRAZIL,BRASILIA,ARGENTINA
    ,BUENOS AIRES,VENEZUELA,CARACAS,FRENCH GUIANA,
    CAYENNE
19Ø DATA GUYANA,GEORGETOWN,PERU,LIMA,URUGUAY,MONTE
    VIDEO,SURINAME,PARAMARIBO,ECUADOR,QUITO
2ØØ DATA CHILE,SANTIAGO,BOLIVIA,SUCRE
21Ø PRINT ::"PRESS ANY KEY TO START."
22Ø CALL KEY(Ø,K,S)
23Ø IF S<1 THEN 22Ø
24Ø SC=Ø
25Ø FOR T=1 TO 12
26Ø CALL CLEAR
27Ø RANDOMIZE
28Ø R=INT(12*RND)
29Ø IF S$(R)="" THEN 28Ø
3ØØ FOR I=Ø TO 11
31Ø PRINT TAB(4);CHR$(65+I);" ";C$(I)
32Ø NEXT I
33Ø PRINT ::S$(R);"  ???"
34Ø CALL KEY(Ø,K,S)
35Ø IF (K<65)+(K>76)THEN 34Ø
36Ø CALL HCHAR(23,26,K)
37Ø IF K-65=R THEN 4ØØ
38Ø PRINT :"THE CAPITAL IS ";C$(R)
39Ø GOTO 42Ø
4ØØ PRINT :"CORRECT!"
41Ø SC=SC+1
42Ø S$(R)=""
43Ø PRINT :::"PRESS <ENTER>";
44Ø CALL KEY(Ø,K,S)
45Ø IF K<>13 THEN 44Ø
46Ø NEXT T
47Ø CALL CLEAR
48Ø PRINT "SOUTH AMERICA":::"YOUR SCORE WAS";SC::
    "OUT OF 12 COUNTRIES.":::
49Ø END
```

Countries

Computers: VIC-20; TRS-80 Color Computer; TRS-80 Model I; TI-99/4A

A quiz-style game, "Countries" can be used while you study elementary geography. The seven continents are listed on the screen by the computer, which then prints a country's name. You must press the number of the continent where the country is located. There are ten countries for each quiz. Fifty countries are listed in the program, so it is possible to go through the quiz four times without repeating. If the country is located correctly, it will not appear again, but if it is located incorrectly it could be chosen again.

Countries works like this:

The first DATA statements contain the name of the country and a corresponding continent number, READ in as C$() and C(). The continents are A$(). If the continent is named correctly, C$(R) is set to "" so it will not be chosen again. T is the number of times the test has been given. After four times, the DATA is RESTOREd for more quizzes.

(Note: Check periodically to make sure the country names and spellings are current.)

Program 9-15. Countries

VIC-20

```
4 DIMA$(7),C$(5Ø),C(5Ø)
6 PRINT"{CLR}YOU WILL BE GIVEN THE NAME OF A COUNT
  RY.":PRINT"WHERE IS IT LOCATED?"
8 PRINT"{DOWN}SEVEN CONTINENTS ARE{2 SPACES}LISTED
  .{2 SPACES}CHOOSE THE{3 SPACES}RIGHT NUMBER."
1Ø FORI=ØTO5Ø:READC$(I),C(I):NEXT
12 DATAFRANCE,3,BRAZIL,2,LIBYA,5,UNITED STATES,1,C
   ANADA,1,MEXICO,1,PERU,2,NORWAY,3
14 DATAU.S.S.R;4,CHILE,2,CHINA,4,JAPAN,4,SPAIN,3,D
   ENMARK,3,ITALY,3,PORTUGAL,3,ENGLAND,3
16 DATAINDIA,4,IRAN,4,GERMANY,3,ARGENTINA,2,URUGUA
   Y,2,PARAGUAY,2,ECUADOR,2,BOLIVIA,2
18 DATACOLOMBIA,2,VENEZUELA,2,KOREA,4,IRELAND,3,QU
   EENSLAND,6,NEW SOUTH WALES,6
2Ø DATAWESTERN AUSTRALIA,6,MONGOLIA,4,NIGER,5,CHAD
   ,5,ZAIRE,5,HONDURAS,1,EGYPT,5
22 DATASWITZERLAND,3,CAMBODIA,4,VIETNAM,4,FINLAND,
   3,SWEDEN,3,GUATEMALA,1,IRAQ,4
24 DATAZAMBIA,5,BANGLADESH,4,AUSTRIA,3,POLAND,3,CZ
   ECHOSLAVAKIA,3,GUYANA,2
26 FORI=1TO7:READA$(I):NEXT
28 DATANORTH AMERICA,SOUTH AMERICA,EUROPE,ASIA,AFR
   ICA,AUSTRALIA,ANTARCTICA
```

```
3Ø PRINT"{DOWN}PRESS F1 TO BEGIN."
32 GETE$:IFE$<>CHR$(133)THEN32
34 T=Ø
36 SC=Ø
38 FORI=1TO1Ø:PRINT"{CLR}"
4Ø FORJ=1TO7:PRINTJ;A$(J):NEXTJ:PRINT"{DOWN}"
42 R=INT(51*RND(Ø)):IFC$(R)=""THEN42
44 PRINTC$(R):PRINT"{DOWN}WHICH CONTINENT? ";
46 GETE$:IFE$=""THEN46
48 IFASC(E$)<49ORASC(E$)>55THEN46
5Ø PRINTE$:IFVAL(E$)=C(R)THEN54
52 PRINT"{DOWN}SORRY, ";C$(R):PRINT"IS IN
   {SHIFT-SPACE}";A$(C(R)):GOTO56
54 PRINT"{DOWN}CORRECT!":SC=SC+1:C$(R)=""
56 PRINT"{DOWN}PRESS <RETURN>";
58 GETE$:IFE$<>CHR$(13)THEN58
6Ø NEXTI
62 PRINT"{CLR}{DOWN}OUT OF 1Ø COUNTRIES":PRINT"
   {DOWN}YOUR SCORE IS";SC
64 PRINT"{2 DOWN}TRY AGAIN? (Y/N)"
66 GETE$:IFE$="N"THEN74
68 IFE$<>"Y"THEN66
7Ø T=T+1:IFT<5THEN36
72 RESTORE:FORI=ØTO5Ø:READC$(I),C(I):NEXT:GOTO34
74 END
```

Program 9-16. Countries

TRS-80 Color Computer; TRS-80 Model I

```
2 DIMA$(7),C$(5Ø),C(5Ø)
3 CLS:PRINT"YOU WILL BE GIVEN THE NAME OF A COUNTR
  Y.  WHERE IS IT LOCATED?"
4 PRINT:PRINT"SEVEN CONTINENTS ARE LISTED.
  {4 SPACES}CHOOSE THE RIGHT NUMBER."
5 FORI=ØTO5Ø:READC$(I),C(I):NEXT
6 DATA FRANCE,3,BRAZIL,2,LIBYA,5,UNITED STATES,1,C
  ANADA,1,MEXICO,1,PERU,2,NORWAY,3,U.S.S.R,4,CHILE
  ,2,CHINA,4,JAPAN,4,SPAIN,3
7 DATA DENMARK,3,ITALY,3,PORTUGAL,3,ENGLAND,3,INDI
  A,4,IRAN,4,GERMANY,3,ARGENTINA,2,URUGUAY,2,PARAG
  UAY,2,ECUADOR,2,BOLIVIA,2
8 DATA COLOMBIA,2,VENEZUELA,2,KOREA,4,IRELAND,3,QU
  EENSLAND,6,NEW SOUTH WALES,6,WESTERN AUSTRALIA,6
  ,MONGOLIA,4,NIGER,5
9 DATA CHAD,5,ZAIRE,5,HONDURAS,1,EGYPT,5,SWITZERLA
  ND,3,CAMBODIA,4,VIETNAM,4,FINLAND,3,SWEDEN,3,GUA
  TEMALA,1,IRAQ,4,ZAMBIA,5
1Ø DATABANGLADESH,4,AUSTRIA,3,POLAND,3,CZECHOSLAVA
   KIA,3,GUYANA,2
11 FORI=1TO7:READA$(I):NEXT
```

```
12 DATANORTH AMERICA,SOUTH AMERICA,EUROPE,ASIA,AFR
   ICA,AUSTRALIA,ANTARCTICA
13 PRINT:PRINT"PRESS <ENTER> TO BEGIN."
14 E$=INKEY$:IFE$<>CHR$(13)THEN14
15 T=Ø
16 SC=Ø
17 FORI=1TO1Ø:CLS:FORJ=1TO7:PRINTJ;A$(J):NEXTJ:PRI
   NT
18 R=RND(51)-1:IFC$(R)=""THEN18
19 PRINTC$(R):PRINT:PRINT"WHICH CONTINENT? ";
2Ø E$=INKEY$:IFE$=""THEN2Ø
21 IFASC(E$)<49THEN2Ø
22 IFASC(E$)>55THEN2Ø
23 PRINTE$:IFVAL(E$)=C(R)THEN25
24 PRINT:PRINT"SORRY, ";C$(R);" IS IN":PRINTA$(C(R
   )):GOTO26
25 PRINT:PRINT"CORRECT!":SC=SC+1:C$(R)=""
26 PRINT:PRINT"PRESS <ENTER>";
27 E$=INKEY$:IFE$<>CHR$(13)THEN27
28 NEXTI
29 CLS:PRINT"OUT OF 1Ø COUNTRIES,":PRINT"YOUR SCOR
   E IS";SC:PRINT:PRINT"TRY AGAIN? (Y/N)"
3Ø E$=INKEY$:IFE$="N"THEN34
31 IFE$<>"Y"THEN3Ø
32 T=T+1:IFT<5THEN16
33 RESTORE:FORI=ØTO5Ø:READC$(I),C(I):NEXT:GOTO15
34 END
```

Program 9-17. Countries

TI-99/4A

```
11Ø DIM A$(7),C$(5Ø),C(5Ø)
12Ø CALL CLEAR
13Ø PRINT "YOU WILL BE GIVEN THE"::"NAME OF A COUN
    TRY."
14Ø PRINT :"WHERE IS IT LOCATED?"
15Ø PRINT ::"SEVEN CONTINENTS ARE LISTED."::"CHOOS
    E THE RIGHT NUMBER."
16Ø FOR I=Ø TO 5Ø
17Ø READ C$(I),C(I)
18Ø NEXT I
19Ø DATA FRANCE,3,BRAZIL,2,LIBYA,5,UNITED STATES,1
    ,CANADA,1,MEXICO,1,PERU,2,NORWAY,3,U.S.S.R.,4,
    CHILE,2
2ØØ DATA CHINA,4,JAPAN,4,SPAIN,3,DENMARK,3,ITALY,3
    ,PORTUGAL,3,ENGLAND,3,INDIA,4,IRAN,4,GERMANY,3
21Ø DATA ARGENTINA,2,URUGUAY,2,PARAGUAY,2,ECUADOR,
    2,BOLIVIA,2,COLOMBIA,2,VENEZUELA,2,KOREA,4,IRE
    LAND,3
```

```
22Ø DATA QUEENSLAND,6,NEW SOUTH WALES,6,WESTERN AU
    STRALIA,6,MONGOLIA,4,NIGER,5,CHAD,5,ZAIRE,5,HO
    NDURAS,1
23Ø DATA EGYPT,5,SWITZERLAND,3,CAMBODIA,4,VIETNAM,
    4,FINLAND,3,SWEDEN,3,GUATEMALA,1,IRAQ,4
24Ø DATA ZAMBIA,5,BANGLADESH,4,AUSTRIA,3,POLAND,3,
    CZECHOSLAVAKIA,3,GUYANA,2
25Ø FOR I=1 TO 7
26Ø READ A$(I)
27Ø NEXT I
28Ø DATA NORTH AMERICA,SOUTH AMERICA,EUROPE,ASIA,A
    FRICA,AUSTRALIA,ANTARCTICA
29Ø PRINT ::"PRESS ANY KEY TO BEGIN."
3ØØ CALL KEY(Ø,K,S)
31Ø IF S<1 THEN 3ØØ
32Ø T=1
33Ø SC=Ø
34Ø FOR I=1 TO 1Ø
35Ø CALL CLEAR
36Ø PRINT "1 NORTH AMERICA":"2 SOUTH AMERICA":"3 E
    UROPE":"4 ASIA"
37Ø PRINT "5 AFRICA":"6 AUSTRALIA":"7 ANTARCTICA":
    ::
38Ø RANDOMIZE
39Ø R=INT(51*RND)
4ØØ IF C$(R)="" THEN 39Ø
41Ø PRINT C$(R)
42Ø PRINT ::"WHICH CONTINENT?"
43Ø CALL KEY(Ø,K,S)
44Ø IF (K<49)+(K>55)THEN 43Ø
45Ø CALL HCHAR(23,21,K)
46Ø IF K=C(R)+48 THEN 49Ø
47Ø PRINT ::"SORRY, ";C$(R);" IS IN":A$(C(R))
48Ø GOTO 52Ø
49Ø PRINT :"CORRECT!"
5ØØ SC=SC+1
51Ø C$(R)=""
52Ø PRINT ::"PRESS <ENTER>";
53Ø CALL KEY(Ø,K,S)
54Ø IF K<>13 THEN 53Ø
55Ø NEXT I
56Ø CALL CLEAR
57Ø PRINT "OUT OF 1Ø PROBLEMS,"
58Ø PRINT :"YOUR SCORE IS";SC
59Ø PRINT ::::"TRY AGAIN? (Y/N)"
6ØØ CALL KEY(Ø,K,S)
61Ø IF K=78 THEN 71Ø
62Ø IF K<>89 THEN 6ØØ
63Ø T=T+1
```

```
640 IF T<5 THEN 330
650 PRINT ::"ONE MOMENT PLEASE."
660 RESTORE
670 FOR I=0 TO 50
680 READ C$(I),C(I)
690 NEXT I
700 GOTO 330
```

Multiplication Invasion

Computers: VIC-20; TRS-80 Color Computer; TRS-80 Model I; TI-99/4A

In game format, this program presents a drill of the multiplication tables. Three random multiplication problems start at the bottom of the screen and move upward. Use the arrow keys to control your position at the top of the screen. Get above a particular problem and answer the problem by pressing number keys. If the answer is only one digit, you must press the ENTER or RETURN key. If the answer is correct, that problem disappears, and another problem appears at the bottom. You'll score ten points for every correct answer. If the answer is incorrect, the problem keeps moving upward, and your score is reduced by five points. The game is over when a problem reaches the top of the screen before you can answer it.

In all program versions, L is the counter limit that adjusts the speed of the game. The counter increments during loops when no key is pressed. After the counter gets to L, the computer PRINTs a blank line, which scrolls the problems upward. SC is the score, and T is the number of problems answered correctly.

Y is the position of your marker — 1, 2, or 3, for left, center, or right sections of the screen. C(Y) and P(Y) are coordinates or positions to draw your marker and PRINT the problem in a particular column.

M and N are random numbers from 0 to 9 for the problems, and A(Y) is the answer for the problem in a particular column. R(Y) counts which row the problem is on so it can be erased when the answer is correct. R is decreased by one for each PRINT statement used in scrolling. R also detects when a problem has reached the top of the screen.

Program 9-18. Multiplication Invasion

VIC-20

(In this program, make sure to use abbreviations for all BASIC keywords in the lines marked with an asterisk (*). If you don't, the program will not run correctly. Refer to Appendix C, "BASIC Keyword Abbreviations," for the proper abbreviations.)

```
20 DIMA(3),C(3),R(3),P(3)
30 PRINT"{CLR}* MULTIPLY INVASION *":PRINT"{DOWN}U
   SE F1 TO MOVE LEFT{7 SPACES}F3 TO MOVE RIGHT."
40 PRINT"{DOWN}GET ABOVE THE PROBLEM THEN TYPE THE
    ANSWER{2 SPACES}AND PRESS <RETURN>."
50 PRINT"{DOWN}DON'T LET THE INVADERSFLOAT TO THE
   {SPACE}TOP OR{3 SPACES}THE GAME ENDS!"
60 PRINT"{DOWN}CORRECT = 1Ø POINTS{3 SPACES}WRONG
   {3 SPACES}= -5"
70 PRINT"{DOWN}CHOOSE: 1 BEGINNER":PRINTTAB(8);"2
   {SPACE}MEDIUM":PRINTTAB(8);"3 FAST";
8Ø GETE$:IFE$=""THEN8Ø
9Ø IFE$="1"THENL=8Ø:GOTO13Ø
1ØØ IFE$="2"THENL=2Ø:GOTO13Ø
11Ø IFE$<>"3"THEN8Ø
12Ø L=7
13Ø PRINT"{CLR}{21 DOWN}":SC=Ø:T=Ø:S=36876:POKES+2
    ,15
14Ø X=7691:Y=2
15Ø C(1)=7682:C(2)=X:C(3)=7699
16Ø P(1)=8164:P(2)=8173:P(3)=8181:FORI=8164TO8185:
    POKEI+3Ø72Ø,Ø:NEXT
17Ø FORI=1TO3:R(I)=21:FORJ=C(I)-2TOC(I)+2:POKEJ,1Ø
    2:POKEJ+3Ø72Ø,6
175 POKEC(I)+3Ø742,2:POKEC(I)+3Ø743,2:NEXTJ,I
18Ø POKEX+22,9Ø
19Ø FORI=1TO3
2ØØ M=INT(1Ø*RND(Ø)):N=INT(1Ø*RND(Ø))
21Ø A(I)=M*N:B=P(I):POKEB,48+M:POKEB+1,42:POKEB+2,
    48+N:POKEB+3,61:POKEB+4,63
22Ø NEXTI
23Ø K=Ø
24Ø GETE$:IFE$=""THEN45Ø
25Ø IFE$<>CHR$(133)THEN28Ø
26Ø POKEC(Y),32:Y=Y-1:IFY<1THENY=1
27Ø GOTO46Ø
28Ø IFE$<>CHR$(134)THEN31Ø
29Ø POKEC(Y),32:Y=Y+1:IFY>3THENY=3
3ØØ GOTO46Ø
31Ø IFASC(E$)<48THEN47Ø
32Ø IFASC(E$)>57THEN47Ø
33Ø POKEC(Y)+22,ASC(E$)
34Ø GETF$:IFF$=""THEN34Ø
35Ø IFF$<>CHR$(13)THEN37Ø
```

```
360 Z=VAL(E$):GOTO4ØØ
37Ø IFASC(F$)<48THEN34Ø
38Ø IFASC(F$)>57THEN34Ø
39Ø POKEC(Y)+23,ASC(F$):POKEC(Y)+3Ø743,2:Z=VAL(E$+
    F$)
4ØØ IFZ=A(Y)THEN42Ø
41Ø POKES,159:GOSUB54Ø:POKES,135:GOSUB54Ø:SC=SC-5:
    GOTO47Ø
42Ø FORI=225TO231:POKES,I:GOSUB54Ø:NEXTI:SC=SC+1Ø
43Ø FORI=ØTO4:POKEP(Y)-(21-R(Y))*22+I,32:NEXTI:POK
    EC(Y)+22,32:POKEC(Y)+23,32
44Ø M=INT(1Ø*RND(Ø)):N=INT(1Ø*RND(Ø)):A(Y)=M*N:R(Y
    )=21:B=P(Y):POKEB,48+M:POKEB+1,42
445 POKEB+2,48+N:POKEB+3,61:POKEB+4,63:FORJ=BTOB+4
    :POKEJ+3Ø72Ø,Ø:NEXT:T=T+1
45Ø K=K+1:IFK<L THEN24Ø
46Ø POKEC(Y)+22,9Ø
* 47Ø PRINT:FORI=1TO3:FORJ=C(I)-2TOC(I)+2:POKEJ,1Ø2:
    NEXTJ,I:POKEC(Y)+22,9Ø:POKEC(Y)+3Ø742,2
475 FORJ=384ØØTO38421:POKEJ,6:NEXTJ
48Ø FORI=1TO3:R(I)=R(I)-1:IFR(I)=-1THEN51Ø
49Ø NEXTI
5ØØ GOTO23Ø
51Ø POKES,159:FORD=1TO2ØØ:NEXT:POKES,135:FORD=1TO2
    ØØ:NEXT:POKES,Ø
52Ø PRINT"{CLR}GAME OVER":PRINT"{2 DOWN}SCORE = ";
    SC:PRINT"{DOWN}PROBLEMS = ";T:PRINT:GOTO55Ø
54Ø FORD=1TO99:NEXT:POKES,Ø:RETURN
55Ø END
```

Program 9-19. Multiplication Invasion

TRS-80 Color Computer

```
2Ø DIMA(3),C(3),R(3),P(3)
3Ø CLS:PRINT"**MULTIPLICATION INVASION**":PRINT:PR
   INT"USE THE ARROW KEYS TO MOVE":PRINT"LEFT OR R
   IGHT."
4Ø PRINT"GET ABOVE THE PROBLEM THEN TYPE THE ANSWE
   R AND PRESS <ENTER>."
5Ø PRINT:PRINT"DON'T LET THE INVADERS FLOAT
   {4 SPACES}TO THE TOP OR THE GAME ENDS!"
6Ø PRINT:PRINT"CORRECT=1Ø POINTS; WRONG=-5"
7Ø PRINT:PRINT"CHOOSE: 1 BEGINNER":PRINTTAB(8);"2
   MEDIUM":PRINTTAB(8);"3 FAST";
8Ø E$=INKEY$:IFE$=""THEN8Ø
9Ø IFE$="1"THEN L=8Ø:GOTO13Ø
1ØØ IFE$="2"THEN L=5Ø:GOTO13Ø
11Ø IFE$<>"3"THEN8Ø
12Ø L=25
```

```
130 CLS:SC=Ø:T=Ø
140 X=32:Y=2
150 C(1)=8:C(2)=3Ø:C(3)=52
160 P(1)=449:P(2)=46Ø:P(3)=471
170 FORI=1TO3:R(I)=15:FORJ=ØTO5:SET(C(I)+J,Ø,4):SE
    T(C(I)+J,1,4):NEXTJ,I
180 SET(X,1,2):SET(X+1,1,2)
190 FORI=1TO3
200 M=RND(1Ø)-1:N=RND(1Ø)-1
210 A(I)=M*N:PRINT@P(I),M;"*";N;"=?";
220 NEXTI
230 K=Ø
240 E$=INKEY$:IFE$=""THEN 45Ø
250 IFE$<>CHR$(8)THEN28Ø
260 SET(X,1,4):SET(X+1,1,4):X=X-22:Y=Y-1:IFX<1ØTHE
    NX=1Ø:Y=1
270 GOTO46Ø
280 IFE$<>CHR$(9)THEN31Ø
290 SET(X,1,4):SET(X+1,1,4):X=X+22:Y=Y+1:IFX>54THE
    NX=54:Y=3
300 GOTO46Ø
310 IFASC(E$)<48 THEN 47Ø
320 IFASC(E$)>57 THEN 47Ø
330 PRINT@X/2+32,E$;
340 F$=INKEY$:IFF$=""THEN34Ø
350 IFF$<>CHR$(13)THEN37Ø
360 Z=VAL(E$):GOTO4ØØ
370 IFASC(F$)<48THEN34Ø
380 IFASC(F$)>57THEN34Ø
390 PRINT@X/2+33,F$;:Z=VAL(E$+F$)
400 IFZ=A(Y)THEN42Ø
410 SOUND125,2:SOUND89,2:SC=SC-5:GOTO47Ø
420 FORI=176TO193:SOUNDI,1:NEXTI:SC=SC+1Ø
430 PRINT@P(Y)-(15-R(Y))*32,"{9 SPACES}";
440 M=RND(1Ø)-1:N=RND(1Ø)-1:A(Y)=M*N:R(Y)=15:PRINT
    @P(Y),M;"*";N;"=?";:T=T+1
450 K=K+1:IFK<L THEN 24Ø
460 SET(X,1,2):SET(X+1,1,2)
470 PRINT@48Ø,"  ":FORI=1TO3:FORJ=ØTO5:SET(C(I)+J,Ø
    ,4):SET(C(I)+J,1,4):NEXTJ,I:SET(X,1,2):SET(X+1
    ,1,2)
480 FORI=1TO3:R(I)=R(I)-1:IFR(I)=1THEN51Ø
490 NEXTI
500 GOTO23Ø
510 SOUND58,5:SOUND5,8
520 CLS:PRINT"GAME OVER":PRINT:PRINT"SCORE = ";SC:
    PRINT:PRINT"NUMBER OF PROBLEMS = ";T:PRINT
530 END
```

Program 9-20. Multiplication Invasion

TRS-80 Model I

```
15 CLEAR3ØØ
2Ø DIMA(3),R(3),P(3)
3Ø CLS:PRINTTAB(5);"**MULTIPLICATION INVASION**":P
   RINT:PRINT"USE THE ARROW KEYS TO MOVE LEFT OR R
   IGHT."
4Ø PRINT"GET ABOVE THE PROBLEM THEN TYPE THE ANSWE
   R AND PRESS <ENTER>."
5Ø PRINT:PRINT"DON'T LET THE INVADERS FLOAT TO THE
    TOP OR THE GAME ENDS!"
6Ø PRINT:PRINT"1Ø POINTS FOR CORRECT ANSWERS; -5 F
   OR INCORRECT ONES"
7Ø PRINT:PRINT"CHOOSE: 1 BEGINNER":PRINTTAB(8);"2
   MEDIUM":PRINTTAB(8);"3 FAST";
8Ø E$=INKEY$:IFE$=""THEN8Ø
9Ø IFE$="1"THEN L=8Ø:GOTO13Ø
1ØØ IFE$="2"THEN L=5Ø:GOTO13Ø
11Ø IFE$<>"3"THEN8Ø
12Ø L=25
13Ø CLS:SC=Ø:T=Ø:H$=CHR$(191):G$=H$+H$+H$+"
    {18 SPACES}"+H$+H$+H$+"{18 SPACES}"+H$+H$+H$
14Ø Y=2:FORI=1TO3:READX(I),P(I):NEXT
15Ø DATA75,9Ø4,96,924,117,944
17Ø PRINT@1Ø,G$:PRINT@X(Y),H$
19Ø FORI=1TO3:R(I)=15
2ØØ M=RND(1Ø)-1:N=RND(1Ø)-1
21Ø A(I)=M*N:PRINT@P(I),M;"*";N;"=?";
22Ø NEXTI
23Ø K=Ø
24Ø E$=INKEY$:IFE$=""THEN 45Ø
25Ø IFE$<>CHR$(8)THEN28Ø
26Ø PRINT@X(Y)," ":Y=Y-1:IFY<1THENY=1
27Ø GOTO46Ø
28Ø IFE$<>CHR$(9)THEN31Ø
29Ø PRINT@X(Y)," ":Y=Y+1:IFY>3THENY=3
3ØØ GOTO46Ø
31Ø IFASC(E$)<48 THEN 47Ø
32Ø IFASC(E$)>57 THEN 47Ø
33Ø PRINT@X(Y),E$;
34Ø F$=INKEY$:IFF$=""THEN34Ø
35Ø IFF$<>CHR$(13)THEN37Ø
36Ø Z=VAL(E$):GOTO4ØØ
37Ø IFASC(F$)<48THEN34Ø
38Ø IFASC(F$)>57THEN34Ø
39Ø PRINT@X(Y)+1,F$;:Z=VAL(E$+F$)
4ØØ IFZ=A(Y)THEN42Ø
41Ø FORI=1TO2Ø:PRINT@X(Y)-3,"--------";:PRINT@X(Y)
    -3,"{8 SPACES}";:NEXT:SC=SC-5:GOTO47Ø
```

```
420 FORI=1TO35:PRINT@X(Y)-64,"*";:PRINT@X(Y)-64,H$
    ;:NEXTI:SC=SC+1Ø:PRINT@X(Y),"  ";
43Ø PRINT@P(Y)-(15-R(Y))*64,"{1Ø SPACES}";
44Ø M=RND(1Ø)-1:N=RND(1Ø)-1:A(Y)=M*N:R(Y)=15:PRINT
    @P(Y),M;"*";N;"= ?";:T=T+1
45Ø K=K+1:IFK<L THEN 24Ø
46Ø PRINT@X(Y),H$
47Ø PRINT@96Ø," ":PRINT@1Ø,G$:PRINT@X(Y),H$
48Ø FORI=1TO3:R(I)=R(I)-1:IFR(I)=2THEN51Ø
49Ø NEXTI
5ØØ GOTO23Ø
51Ø FORI=1TO2Ø:PRINT@X(Y)," ":PRINT@X(Y),H$:NEXT
52Ø CLS:PRINT"GAME OVER":PRINT:PRINT"SCORE = ";SC:
    PRINT:PRINT"NUMBER OF PROBLEMS = ";T:PRINT
53Ø END
```

Program 9-21. Multiplication Invasion

TI-99/4A

```
11Ø DIM A(3),C(3),R(3),P(3)
12Ø CALL CLEAR
13Ø PRINT "* MULTIPLICATION INVASION *"
14Ø PRINT ::"USE THE ARROW KEYS TO MOVE  LEFT OR R
    IGHT."
15Ø PRINT :"GET ABOVE THE PROBLEM THEN  TYPE THE A
    NSWER AND":"PRESS <ENTER>"
16Ø PRINT :"DON'T LET THE INVADERS FLOATTO THE TOP
     OR THE GAME ENDS!"
17Ø PRINT ::"CORRECT = 1Ø POINTS":"WRONG
    {3 SPACES}= -5 POINTS"
18Ø CALL CHAR(96,"FFFFFFFFFFFFFFFF")
19Ø CALL COLOR(9,7,1)
2ØØ CALL CHAR(97,"Ø8Ø8Ø8Ø8Ø87F3E1CØ8")
21Ø PRINT ::"CHOOSE: 1 BEGINNER":TAB(9);"2 MEDIUM"
    :TAB(9);"3 FAST";
22Ø CALL KEY(Ø,K,S)
23Ø IF (K<49)+(K>51)THEN 22Ø
24Ø L=(4-(K-48))*2
25Ø CALL CLEAR
26Ø SC=Ø
27Ø T=Ø
28Ø C(1)=3
29Ø C(2)=14
3ØØ C(3)=25
31Ø FOR I=1 TO 3
32Ø CALL HCHAR(1,C(I),96,5)
33Ø NEXT I
34Ø FOR Y=1 TO 3
35Ø RANDOMIZE
36Ø GOSUB 88Ø
```

```
370 NEXT Y
380 Y=2
390 CALL HCHAR(2,C(Y)+2,97)
400 KL=Ø
410 CALL KEY(Ø,K,S)
420 IF S<1 THEN 770
430 IF K<>83 THEN 490
440 CALL HCHAR(2,C(Y)+2,32)
450 Y=Y-1
460 IF Y>Ø THEN 770
470 Y=1
480 GOTO 770
490 IF K<>68 THEN 550
500 CALL HCHAR(2,C(Y)+2,32)
510 Y=Y+1
520 IF Y<4 THEN 770
530 Y=3
540 GOTO 770
550 IF (K<48)+(K>57)THEN 770
560 CALL HCHAR(2,C(Y)+2,K)
570 B=K-48
580 CALL KEY(Ø,K,S)
590 IF S<1 THEN 580
600 IF K=13 THEN 640
610 IF (K<48)+(K>57)THEN 580
620 CALL HCHAR(2,C(Y)+3,K)
630 B=1Ø*B+(K-48)
640 IF B<>A(Y)THEN 740
650 FOR I=262 TO 392 STEP 12
660 CALL SOUND(-99,I,2)
670 NEXT I
680 CALL HCHAR(R(Y),C(Y),32,5)
690 CALL HCHAR(2,C(Y)+2,32,2)
700 GOSUB 880
710 SC=SC+1Ø
720 T=T+1
730 GOTO 760
740 CALL SOUND(1ØØ,-6,4)
750 SC=SC-5
760 CALL HCHAR(2,C(Y)+2,32,2)
770 KL=KL+1
780 CALL HCHAR(2,C(Y)+2,97)
790 IF KL<L THEN 410
800 PRINT
810 FOR I=1 TO 3
820 CALL HCHAR(1,C(I),96,5)
830 R(I)=R(I)-1
840 IF R(I)<=2 THEN 980
850 NEXT I
860 CALL HCHAR(2,C(Y)+2,97)
```

```
870 GOTO 400
880 M=INT(10*RND)
890 N=INT(10*RND)
900 A(Y)=M*N
910 CALL HCHAR(23,C(Y),M+48)
920 CALL HCHAR(23,C(Y)+1,42)
930 CALL HCHAR(23,C(Y)+2,N+48)
940 CALL HCHAR(23,C(Y)+3,61)
950 CALL HCHAR(23,C(Y)+4,63)
960 R(Y)=23
970 RETURN
980 FOR I=1 TO 16
990 CALL SCREEN(I)
1000 NEXT I
1010 CALL CLEAR
1020 PRINT "GAME OVER"
1030 PRINT ::"SCORE = ";SC::"NUMBER OF PROBLEMS =
     ";T::
1040 END
```

Defend

Computer: VIC-20

This is a simple intercept, shooting game for one person. You are positioned at the left of the screen and need to defend your territory — don't let the invaders coming from the right of the screen get to your border. Line up horizontally with an invader by pressing the up arrow to move up and the down arrow to go down, then shoot with the space bar or the f7 key. You score ten points for each invader you successfully shoot, but you lose five points if you miss.

After you have played this game once or twice, adapt it for your own game — use different graphics and screen colors. Change the motion to vertical instead of horizontal. Adapt the scoring. Perhaps you could change the shapes of the invaders after certain scores are reached — and vary the speed they travel.

Here's how "Defend" works:

Line	Function
1	Turn on volume for sound and initialize TS for the top score.
2	Define function R(X) to calculate the location number for a random row and branch to line 200.
10	Clear screen and set screen and border color. Initialize variables — N is the location of your ship, SC is the score, and D is difficulty level.

20 Place defending ship on screen.

22–25 Randomly place invaders, making sure they are not on the same row.

30 Detect your key press. If it is one of the firing keys (space bar or f7), branch to line 60.

32–34 If arrow keys are pressed, move up or down.

35 Increment L to determine speed of invaders.

36 Increment invaders' positions — move one spot to left.

37–42 If an invader reaches left side of screen, branch to line 100 to end game.

44–50 Move invaders and branch back to receive next key press.

60 Make firing noise and blink defender.

62–68 Check positions of invaders to see if one was shot.

70 Decrease score by 5 if shot missed.

72–78 Procedure if invader is shot and new invader position chosen.

80 Score is increased by 10, clear invader, and stop noise.

82–84 PRINT score and branch back for next key press.

90–94 Check boundary positions of defender, then draw defender on screen in new position.

100–110 Procedure at game's end.

120–160 PRINT ending message, score, and high score.

170–190 PRINT option to try again and branch appropriately.

200–280 PRINT instruction screen.

290 END.

Program 9-22. Defend

VIC-20

(In this program, make sure to use abbreviations for all BASIC keywords in the lines marked with an asterisk (*). If you don't, the program will not run correctly. Refer to Appendix C, "BASIC Keyword Abbreviations," for the proper abbreviations.)

```
1 POKE36878,15:TS=Ø
2 DEF FNR(X)=7745+22*(INT(RND(Ø)*2Ø)):GOTO 2ØØ
1Ø PRINT"{CLR}":POKE36879,1Ø7:N=7922:SC=Ø:D=1Ø
2Ø POKEN,9Ø
22 I=FNR(X):J=FNR(X):K=FNR(X):H=FNR(X)
24 IFH=IORH=JORH=KORI=JORI=KORJ=K THEN 22
25 POKEH,42:POKEI,42:POKEJ,42:POKEK,42
3Ø GETA$:IFA$=CHR$(136)OR A$=CHR$(32)THEN 6Ø
32 IFA$=CHR$(94)THENPOKEN,32:N=N-22:GOTO9Ø
34 IFA$=CHR$(17)THENPOKEN,32:N=N+22:GOTO9Ø
35 L=L+1:IFL<DTHEN3Ø
36 H=H-1:I=I-1:J=J-1:K=K-1:L=Ø
```

```
 37 IF(H-768Ø)/22=INT((H-768Ø)/22)THEN1ØØ
 38 IF(I-768Ø)/22=INT((I-768Ø)/22)THEN1ØØ
 4Ø IF(J-768Ø)/22=INT((J-768Ø)/22)THEN1ØØ
 42 IF(K-768Ø)/22=INT((K-768Ø)/22)THEN1ØØ
*44 POKE36876,24Ø:POKEH+1,32:POKEI+1,32:POKEJ+1,32:
    POKEK+1,32:POKEH,42:POKEI,42:POKEJ,42:POKEK,42:
    POKE36876,Ø
 5Ø GOTO3Ø
 6Ø FORM=2ØØTO22Ø:POKE36876,M:POKEN+3Ø72Ø,2:POKEN+3
    Ø72Ø,1:NEXT:POKE36876,Ø
 62 IFH>NANDH<N+22THEN72
 64 IFI>NANDI<N+22THEN74
 66 IFJ>NANDJ<N+22THEN76
 68 IFK>NANDK<N+22THEN78
 7Ø SC=SC-5:GOTO82
 72 POKE36877,16Ø:POKEH,1Ø2:B=H:H=FNR(X):GOTO8Ø
 74 POKE36877,15Ø:POKEI,1Ø2:B=I:I=FNR(X):GOTO8Ø
 76 POKE36877,18Ø:POKEJ,1Ø2:B=J:J=FNR(X):GOTO8Ø
 78 POKE36877,19Ø:POKEK,1Ø2:B=K:K=FNR(X)
 8Ø SC=SC+1Ø:POKEB,32:POKE36877,Ø
 82 PRINT"{HOME}{BLU}{RVS}{22 SPACES}{OFF}":PRINT"
    {HOME}{WHT}SCORE =";SC:IFSC>5ØØTHEND=5
 84 GOTO3Ø
 9Ø IFN<7724THENN=7724
 92 IFN>8164THENN=8164
 94 POKEN,9Ø:GOTO3Ø
 1ØØ POKE36877,22Ø:FORC=38445TO38885STEP22:POKEC,2:
     NEXTC:FORC=1TO1ØØ:NEXTC
 11Ø FORC=38445TO38885STEP22:POKEC,1:NEXTC:POKE3687
     7,Ø
 12Ø PRINT"{WHT}GAME OVER"
 13Ø FORC=1TO1ØØØ:NEXT:POKE36879,27
 14Ø PRINT"{CLR}{BLU}{2 DOWN}YOUR FINAL SCORE WAS
     {3 SPACES}":PRINT"{RED}";SC:PRINT"{BLU}
     {2 DOWN}"
 15Ø IF SC>TS THEN TS=SC
 16Ø PRINT"HIGH SCORE = ";TS
 17Ø PRINT"{GRN}{3 DOWN}TRY AGAIN? (Y/N)"
 18Ø GETA$:IF A$="Y"THEN 1Ø
 185 IF A$<>"N" THEN 18Ø
 19Ø PRINT"{BLU}":END
 2ØØ POKE36879,186:PRINT"{CLR}{BLK}":PRINTTAB(5);"*
     * DEFEND **{2 DOWN}"
 22Ø PRINT"{2 DOWN}PRESS ↑ TO MOVE UP":PRINT"PRESS
     {SPACE}CRSR TO GO DOWN"
 23Ø PRINT"{DOWN}PRESS F7 OR SPACE":PRINT"TO FIRE.
     {3 DOWN}"
 24Ø PRINT"KEEP THE INVADERS FROMREACHING YOUR SIDE
      OF THE SCREEN."
```

```
250 PRINT"{2 DOWN}{RED}PRESS RETURN TO START";
260 GETA$:IFA$=""THEN 260
270 IF ASC(A$)=13 THEN 10
280 GOTO260
290 END
```

Appendices

A Beginner's Guide to Typing In Programs

What Is a Program?

A computer cannot perform any task by itself. Like a car without gas, a computer has *potential*, but without a program, it isn't going anywhere. Most of the programs published in this book are written in a computer language called BASIC. BASIC is easy to learn and is built into most computers (on some computers, you have to purchase an optional BASIC cartridge).

BASIC Programs

To start out, type in only programs written for your machine. Later, when you gain experience with your computer's BASIC, you can try typing in and converting certain programs from one computer to yours.

Computers can be picky. Unlike the English language, which is full of ambiguities, BASIC usually has only one right way of stating something. Every letter, character, or number is significant. A common mistake is substituting a letter such as O for the numeral 0, a lowercase l for the numeral 1, or an uppercase B for the numeral 8. Also, you must enter all punctuation such as colons and commas just as they appear in the book. Spacing can be important. To be safe, type in the listings *exactly* as they appear.

Braces and Special Characters

The exception to this typing rule is when you see the braces, such as {DOWN}. Anything within a set of braces is a special character or characters that cannot easily be listed on a printer. When you come across such a special statement, refer to the appropriate key for your computer. For example, if you have a VIC-20, refer to the VIC's section in "How To Type In Programs."

About DATA Statements

Some programs contain a section or sections of DATA statements. These lines provide information needed by the program. Some DATA statements contain actual programs (called machine language); others contain graphics codes. These lines are especially sensitive to errors.

If a single number in any one DATA statement is mistyped, your machine could lock up, or crash. The keyboard, break key, and RESET (or STOP) keys may all seem dead, and the screen may go blank. Don't panic — no damage is done. To regain control, you have to turn off your computer, then turn it back on. This will erase whatever program was in memory, so always SAVE a copy of your program before you RUN it. If your computer crashes, you can LOAD the program and look for your mistake.

Sometimes a mistyped DATA statement will cause an error message when the program is RUN. The error message may refer to the program line that READs the data. *The error is still in the DATA statements, though.*

Get to Know Your Machine

You should familiarize yourself with your computer before attempting to type in a program. Learn the statements you use to store and retrieve programs from tape or disk. You'll want to save a copy of your program, so that you won't have to type it in every time you want to use it. Learn to use your machine's editing functions. How do you change a line if you made a mistake? You can always retype the line, but you at least need to know how to backspace. Do you know how to enter reverse video, lowercase, and control characters? It's all explained in your computer's manuals.

A Quick Review

1) Type in the program a line at a time, in order. Press RETURN or ENTER at the end of each line. Use backspace or the back arrow to correct mistakes.
2) Check the line you've typed against the line in the book. You can check the entire program again if you get an error when you RUN the program.
3) Make sure you've entered statements in brackets as the appropriate control key (see Appendix B, "How To Type In Programs").

APPENDIX B

How to Type In Programs

To make it easy to tell exactly what to type when entering one of these programs into your computer, we have established the following listing conventions. There is a separate key for each computer. Refer to the appropriate tables when you come across an unusual symbol in a program listing. If you are unsure how to actually enter a control character, consult your computer's manuals.

VIC-20

Generally, any VIC-20 program listings will contain words within braces which spell out any special characters: {DOWN} would mean to press the cursor down key. {5 SPACES} would mean to press the space bar five times.

To indicate that a key should be *shifted* (hold down the SHIFT key while pressing the other key), the key would be underlined in our listings. For example, S would mean to type the S key while holding the Shift key. This would appear on your screen as a "heart" symbol. If you find an underlined key enclosed in braces (e.g., {10 N}), you should type the key as many times as indicated (in our example, you would enter ten shifted N's).

If a key is enclosed in special brackets, [< >], you should hold down the *Commodore key* while pressing the key inside the special brackets. (The Commodore key is the key in the lower-left corner of the keyboard.) Again, if the key is preceded by a number, you should press the key as many times as necessary.

Rarely, you'll see a solitary letter of the alphabet enclosed in braces. You should never have to enter such a character on the VIC-20, but if you do, you would have to leave the quote mode (press RETURN and cursor back up to the position where the control character should go), press CTRL-9 (RVS ON), the letter in braces, and then CTRL-0 (RVS OFF).

About the *quote mode*: you know that you can move the cursor around the screen with the CRSR keys. Sometimes a programmer will want to move the cursor under program control. That's why you see all the {LEFT}'s, {HOME}'s, and {BLU}'s in our programs. The only way the computer can tell the difference between direct and programmed cursor control is the quote mode.

Once you press the quote (the double quote, SHIFT-2), you are in the quote mode. If you type something and then try to change it by moving the cursor left, you'll only get a bunch of reverse-video lines. These are the symbols for cursor left. The only editing key that isn't programmable is the DEL key; you can still use DEL to back up and edit the line. Once you type another quote, you are out of quote mode.

You also go into quote mode when you INSerT spaces into a line. In any case, the easiest way to get out of quote mode is to just press RETURN. You'll then be out of quote mode and you can cursor up to the mistyped line and fix it.

Use the following table when entering cursor and color control keys:

When You Read:	Press:	See:	When You Read:	Press:	See:
{CLR}	SHIFT CLR/HOME		{GRN}	CTRL 6	
{HOME}	CLR/HOME		{BLU}	CTRL 7	
{UP}	SHIFT ↑ CRSR ↓		{YEL}	CTRL 8	
{DOWN}	↑ CRSR ↓		{F1}	f1	
{LEFT}	SHIFT ← CRSR →		{F2}	SHIFT f1	
{RIGHT}	← CRSR →		{F3}	f3	
{RVS}	CTRL 9		{F4}	SHIFT f3	
{OFF}	CTRL 0		{F5}	f5	
{BLK}	CTRL 1		{F6}	SHIFT f5	
{WHT}	CTRL 2		{F7}	f7	
{RED}	CTRL 3		{F8}	SHIFT f7	
{CYN}	CTRL 4		←	←	
{PUR}	CTRL 5		↑	SHIFT ↑	

Texas Instruments 99/4

The only special characters used are in PRINT statements to indicate where two or more spaces should be left between words. For example, ENERGY {10 SPACES} MANAGEMENT means that ten spaces should be left between the words ENERGY and MANAGEMENT. Do not type in the braces or the words 10 SPACES. Enter all programs with the ALPHA LOCK on (in the down position). Release the ALPHA LOCK to enter lowercase text.

TRS-80 Color Computer

As with the TI-99/4A, some special characters are used in the PRINT statements. To indicate where two or more spaces should be left between words, the program listing will show something like LOCATING {5 SPACES} POINTS. This means that five spaces should be left between the words LOCATING and POINTS. Don't type in the braces or the actual words 5 SPACES. All the programs should be entered in uppercase on the TRS-80 Color Computer.

Another character you'll see in the program listing is the up caret (∧). This represents the exponential symbol you'll see on your screen as an up arrow (↑). Simply press the up arrow key on the left side of the keyboard when you see the ∧ symbol in a program.

APPENDIX C

BASIC Keyword Abbreviations

Most keywords in VIC BASIC can be abbreviated by typing the first one or two letters of the word followed by the SHIFTed next letter. The use of abbreviations will not only save time when entering programs or commands but will reduce the total number of characters per line. Abbreviated keywords will LIST in the full, unabbreviated form when used in a program line.

Most Commonly Used Keyword Abbreviations

Keyword:	Press:	See:	Keyword:	Press:	See:
AND	A SHIFT N	A ╱	ON	NONE	ON
CLR	C SHIFT L	C └	OR	NONE	OR
DATA	D SHIFT A	D ♠	PEEK	P SHIFT E	P ─
DIM	D SHIFT I	D ╮	POKE	P SHIFT O	P ┌
FOR	F SHIFT O	F ┌	PRINT	?	?
GET	G SHIFT E	G ─	READ	R SHIFT E	R ─
GOSUB	GO SHIFT S	GO ♥	REM	NONE	REM
GOTO	G SHIFT O	G ┌	RESTORE	RE SHIFT S	RE ♥
IF	NONE	IF	RETURN	RE SHIFT T	RE ▏
INPUT	NONE	INPUT	RND	R SHIFT N	R ╱
INT	NONE	INT	RUN	R SHIFT U	R ╭
LET	L SHIFT E	L ─	SAVE	S SHIFT A	S ♠
LIST	L SHIFT I	L ╮	STEP	ST SHIFT E	ST ─
LOAD	L SHIFT O	L ┌	TAB(	T SHIFT A	T ♠
NEXT	N SHIFT E	N ─	THEN	T SHIFT H	T ▕

Other Abbreviations

Keyword:	Press:	See:	Keyword:	Press:	See:
ABS	A SHIFT B	A ◫	MID$	M SHIFT I	M ╮
ASC	A SHIFT S	A ♥	NOT	N SHIFT O	N □
ATN	A SHIFT T	A ▏	OPEN	O SHIFT P	O □
CHR$	C SHIFT H	C ▕	POS	NONE	POS
CLOSE	CL SHIFT O	CL □	PRINT#	P SHIFT R	P ▁
CMD	C SHIFT M	C ╲	RIGHT$	R SHIFT I	R ╮
CONT	C SHIFT O	C □	SGN	S SHIFT G	S ▏
COS	NONE	COS	SIN	S SHIFT I	S ╮
DEF	D SHIFT E	D ▔	SPC(	S SHIFT P	S □
END	E SHIFT N	E ╱	SQR	S SHIFT Q	S ●
EXP	E SHIFT X	E ♣	STOP	S SHIFT T	S ▏
FN	NONE	FN	STR$	ST SHIFT R	ST ▁
FRE	F SHIFT R	F ▁	SYS	S SHIFT Y	S ▕
GET#	NONE	GET#	TAN	NONE	TAN
INPUT#	I SHIFT N	I ╱	USR	U SHIFT S	U ♥
LEFT$	LE SHIFT F	LE ▁	VAL	V SHIFT A	V ♠
LEN	NONE	LEN	VERIFY	V SHIFT E	V ▔
LOG	NONE	LOG	WAIT	W SHIFT A	W ♠

Index

Notes

Notes

Notes

Notes

Notes

Your computer may have limited memory, but it's still a powerful tool — with the right programs. In less than 4K of memory, you can play games, check homework, calculate a loan, or create dazzling graphic displays. When you try out the dozens of different programs in this book, you'll discover the power of your computer.

C. Regena, columnist for **COMPUTE! Magazine** and **COMPUTE!'s Gazette** and author of the best-selling **Programmer's Reference Guide to the TI-99/4A,** knows how to get the most out of a small computer. Her programs for the VIC-20, the TRS-80 Color Computer, and the TI-99/4A are not only excellent software, but also examples of effective, memory-saving programming. You'll see how to translate programs from one computer to another, as well as the way that the programs work. The programs and information include:

- How to conserve memory and get more into your computer
- Graphics and sound
- Games that are fun to play — while they teach
- Homework helpers
- Computer sorting
- Typing, geography, and mathematics aids

ISBN 0-942386-38-8