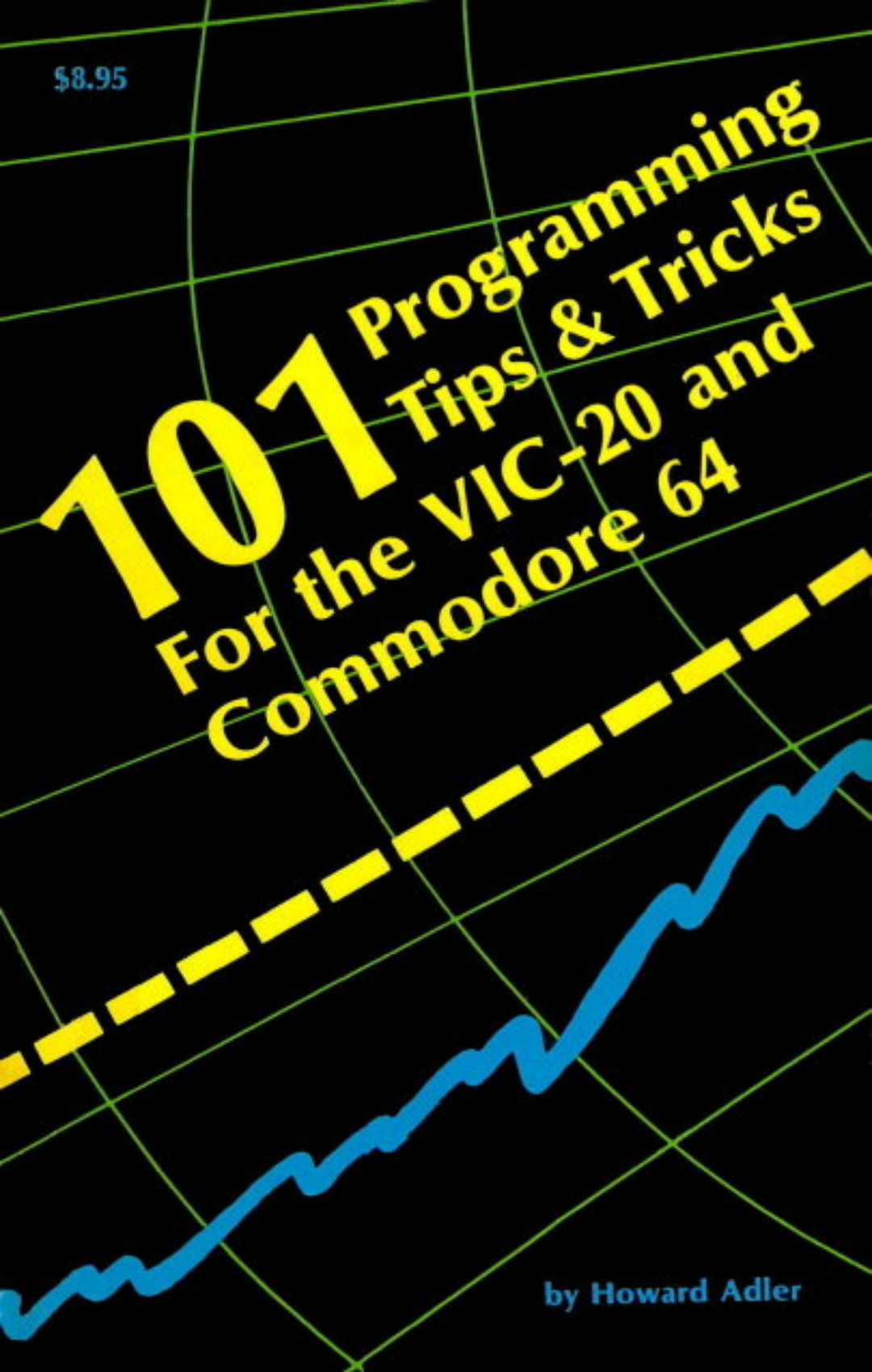
$8.95

101 Programming Tips & Tricks For the VIC-20 and Commodore 64



by Howard Adler

101 Programming Tips & Tricks For the VIC-20 and Commodore 64

by Howard Adler

ARCsoft Publishers
WOODSBORO, MARYLAND

FIRST EDITION
FIRST PRINTING

Library of Congress (LC) catalog number: 83-5973

ISBN 0-86668-030-6

Preface

The microcomputers from Commodore Business Machines Inc. are among the most popular around the world for use in the home, in the classroom, in the business office. In fact, the VIC-20 may be the all-time best selling personal computer to date.

The lightweight desktop design of the ultramodern VIC and Model 64 and their powerful BASIC language capabilities place them in the forefront of the new wave of computers for hobbyists, students, teachers, parents, professionals and business persons who want to learn the new technology.

These powerful microcomputers are not toys! Their hardware and software combinations make them highly useful tools in the business environment and the classroom as well as in the home.

The total number of applications to which the various Commodore models can be put is limited only by the scope of the imagination. In this book, we have attempted to create and share 101 new practical sets of program ideas and appropriate applications software for your use.

This book, as well as all published by *ARCsoft Publishers,* is written for newcomers, novices and first-timers, as well as for advanced users of microcomputers. Our intention has been to provide easy-to-type-and-run programs for the VIC-20, Commodore 64 and other computers using the Commodore version of the BASIC programming language. You type these programs into your computer and it does the rest. You do not have to be a programmer or program writer to use this book.

This volume is a companion book to *34 VIC-20 Computer Programs for Home, School & Office* and the *VIC-20 Computer Program Writing Workbook.*

—Howard Adler

Table of Contents

Text on Text

Gee Whiz

Number Crunching

Money Matters

Colorful Graphics

Introduction

This is a book of programming advice and actual working programs for the Commodore VIC-20, Commodore Model 64, and other personal microcomputers using the Commodore Business Machines Inc. version of the BASIC programming language.

There continues to be a great need for practical and useful software for the new generation of popular personal computers. The Commodore 64, Commodore VIC-20, and other personal microcomputers from Commodore Business Machines Inc., are among the world's most popular computers for use in the home, in the classroom and even in the business office.

Commodore computers are powerful, versatile and flexible—but what can they do?

Once you've purchased the hardware, you need down-to-earth workable programs to run the computer. The aim of this book is to provide more than 101 new and different program ideas and appropriate complete easy-to-type ready-to-run all-new and different sets of program

listings for you to use in your own Commodore computer, to make your computer work for you.

These programs are useful in themselves. They also make good starting points for further development as you learn more and more about how to program your own computer. Read these 101 programs. Type them into your computer. Watch them run. Analyze how the lines in the listings cause the computer to step through a sequence of operations to reach a final goal. You'll see how programs are organized, how they work. You can use these fun and practical programs and then, later, modify them to do more or different work. Expand them to suit your needs as your interests grow.

These programs are designed to be typed into your Commodore computer, just as you find them in this book, via the typewriter-style keyboard. No other programming is needed.

We assume you have read the owner's manual and instructional pamphlets which came with your Commodore computer and any accessories you may have. You know how to hook the computer to your TV and to other accessories you have. You know how to type programs into the computer. If you don't know these beginning steps, please review the instructions in the Commodore owner's manual.

You *do not have to be a programmer* to use these pieces of software. Just type them in, as you find them here, and run them. They will work!

Computer printouts

To make sure no errors appear in this book, we have written and tested each of these programs on our own VIC-20 *and* we have printed every one on a VIC-1525E graphic printer. The hardcopy printout from that line printer is reproduced directly in this book.

The Commodore computer operated the printer and listed these programs. No human hands came between the computer and these listings so no re-typing or proofreading errors have been introduced. You should find that these programs will run exactly as reproduced here.

If, after typing in a program, you get an error message from your Commodore computer, compare your typed

program carefully with the program lines in this book. Remember that every space, punctuation mark, letter, number and symbol counts and must be in the computer exactly as in this book.

If you do get such an error message, the most likely cause will be found to have been a typing error in transferring the program from this book into the computer. However, should you find an error in this book, please call it to the attention of the author by sending a postcard or letter to him in care of *ARCsoft Publishers,* P.O. Box 132, Woodsboro, MD 21798 USA. The author will appreciate being able to make any necessary corrections to future editions of this book.

How to use this book

This book has been organized into seven sections for convenience in locating programs. The sections are:

Fun & Games includes many new ideas you can use in writing game programs for your Commodore computer;

Text on Text has programming tips for handling words in your Commodore computer;

Gee Whiz is a group of quick-type-and-run programs you can use to excite your family, friends and neighbors;

Number Crunching includes many hints, secrets and shortcuts for handling numbers in your Commodore computer;

Money Matters holds interesting programs to help you handle your household budget as well as store and office routines for business persons;

Colorful Graphics are programming shortcuts designed to help you get unusual video displays from your Commodore machine;

Please note that each and every one of the 101 tips in this book includes a complete, ready-to-run program in BASIC for you to use. Try them all. They are great fun to run. They are especially designed to be short so you won't have to spend hours typing in one program.

As your knowledge of BASIC and how it makes your Commodore computer work grows, you will be able to build on these elementary programs to create ever-more-complex schemes for practical applications.

Endless running

Many of the programs in this book will continue to run until you command them off manually via the RUN STOP key. You may stop any run, at any time, by pressing the RUN STOP key on the left side of the keyboard. This RUN STOP function in the Commodore computers is the same as what is called BREAK on other microcomputers.

REMarks

As you read through the 101 programs in this book, you will notice very few REM, or remarks, statements. The author's training in writing BASIC-language computer programs included an emphasis on brevity and saving memory space. A sharp editing pencil was in order—and still is!

REMarks and explanations in software are out. Honing, fine tuning and waste trimming are in. Use of coding form program-writing worksheets is encouraged. Such worksheets can be found in the publication, *VIC-20 Computer Program Writing Workbook,* available from ARCsoft Publishers. Your objective always should be to make the most efficient use of available memory.

Here's another important note to remember: even though they may be headed toward the same goal, no two programmers will write the exact same list of BASIC instructions, or program lines, from scratch. As you load these programs into your Commodore computer, one at a time, you'll make modifications to suit your personal needs and interests if you want to. For instance, the exact wording of PRINT statements can be changed. Or two or more programs can be combined into one grand scheme. Your applications may vary.

By the way, if you want to load more than one of these programs into your computer at the same time, be sure to use different sets of line numbers for different programs. For example, only one line can be numbered 10. There cannot be two lines numbered 10 in the computer at the same time.

We use the words ENTER and RETURN interchangeably. Programmers today generally mix the use of the two words, to mean the same thing. In this case, we mean the RETURN key on the right hand side of the Commodore keyboard. For example, you ENTER a line into

program memory by typing it into the computer and pressing the RETURN key at the end of the line. Pressing RETURN causes it to be ENTERed into the computer's program memory.

Other computers

These programs will run on other computers which use the BASIC language. However, to run these programs on other non-Commodore computers, you probably will have to make at least slight modifications to program lines. Graphics commands, especially, will differ elsewhere. Also, use of multiple-statement lines, using the colon (:), can be quite different in other brand-names of computer hardware.

Also, if you use a non-Commodore computer, such things as line numbering, spacing, logical tests, multiplication symbols, print statements and other instructions may be different.

Refer to the owner's manual which came with your non-Commodore computer. Compare its version of the BASIC language with Commodore BASIC.

The author would like to have your suggestions for future editions of this work, or for other titles in this series for the Commodore computers. The author may be addressed in care of *ARCsoft Publishers.*

Standalone vs. subroutine

The 101 programs in this book may be used as portions of larger lists of instructions to your Commodore computer. They can be written in as GOTO or GOSUB objects. To do so, make appropriate changes to the first line (usually numbered 10 in this book) and to the last line of each program.

If you create a subroutine, remember that every GOSUB must have a RETURN. RETURN must be the last line of every subroutine.

If you work one of these programs into a larger set of instructions, be especially careful of your memory (variable) names or labels. They must agree with, and fit into, those you are using in the main program. Also, be careful of line numbers. No two programs can occupy the exact same set of line numbers.

Learning programming

These programs are written to be typed into your Commodore computer just as you find them here—with no extra programming needed. We assume you know how to turn on your computer and how to go about typing in a program.

Many of the programs and most of the programming advice in this book will be of interest to old hands, as well as newcomers, since we are presenting new twists and more powerful exercises aimed at making your computer do more work, more quickly.

Amidst the 101 programs in this book, you will find countless ideas for using your computer. Each piece of software is intended to make you a more-versatile programmer and make your programming chores lighter.

This is not a replacement for the owner's manual which came with your Commodore computer. You need to read it thoroughly, first, then use this book as a supplement.

Use this book to stimulate your thinking about how to approach various software problems and projects. Use it to get good ideas for new and different approaches to all of your programming goals. As you grow and develop as a program author, modify these programs to make your computer do even more.

Happy programming!

Fun & Games

1 Coin Toss

Here's a handy way to settle arguments. Toss a coin. Only this time, let the computer do the work!

Type in the program. Run it. The computer will report *heads* or *tails* after each toss.

For a new toss, press the RETURN key on your computer's keyboard.

Line 10 clears the screen. A random number—either zero or one—is generated at line 20 and tested to see if it is a zero. If it is, the computer prints *heads*. If not, the computer drops to line 30 where it prints *tails*. Lines 50, 60 and 70 accomplish the restart when you press RETURN.

Program Listing

```
10 PRINT CHR$(147):REM SCREEN CLEAR
15 Q=RND(-TI)
20 IF (INT(2*(RND(1))))<1 THEN PRINT
   "HEADS":GOTO 40
30 PRINT "TAILS"
40 PRINT:PRINT:PRINT
50 PRINT"PRESS RETURN"
60 INPUT"TO TOSS ANOTHER COIN";K$
70 CLR
80 GOTO 10
```

2 Traditional Dice Roll

Here's a simple, brief way to roll and display results for two dice.

Lines 100-110 get a random number between 1 and 6 and store it in A. Lines 200-210 get another random number from 1 to 6 and store it in B.

Lines 300-310 print the contents of A and B along with a suitable message.

Program Listing

```
10 Q=RND(-TI)
20 PRINT CHR$(147):REM SCREEN CLEAR
100 A=INT(7*RND(1))
110 IF A<1 THEN 100
200 B=INT(7*RND(1))
210 IF B<1 THEN 200
300 PRINT"FIRST DICE: ";A
310 PRINT"SECOND DICE:";B
400 FOR L=1 TO 10:PRINT:NEXT L
410 PRINT"PRESS RETURN"
420 INPUT"TO ROLL DICE AGAIN";K$
430 CLR
440 GOTO 20
```

3 See Two Dice

Here's a quick way to add real dice to any fun program you are designing for your computer.

This program rolls two dice and lets you see the results, as with real dice. This is especially useful in those games where it is important to see the value of each.

The subroutine in lines 100-140 generates the necessary pair of random numbers. Lines 60, 70 and 80 make the display you want.

Note that lines 60 and 80 each have nine asterisks. Line 140 is RETURN and must be the last line in the program.

After you type in and RUN the program,press RETURN on your computer's keyboard to roll the dice.

Program Listing

```
10 Q=RND(-TI)
20 PRINT CHR$(147):REM SCREEN CLEAR
30 PRINT"PRESS RETURN"
```

```
40 INPUT"TO ROLL TWO DICE";K$:PRINT:PRINT
45 PRINT CHR$(147):REM SCREEN CLEAR
50 GOSUB 100
60 PRINT"**********"
70 PRINT"*";DL;"*";DR;"*"
80 PRINT"**********"
90 PRINT:PRINT:GOTO 30
100 DL=INT(7*RND(1))
110 IF DL<1 THEN 100
120 DR=INT(7*RND(1))
130 IF DR<1 THEN 120
140 RETURN
```

4 See Four Dice

Two dice not enough for your game? Here's how to see four dice after a roll!

Naturally, this program works just like the program in tip number 3 except that the FOR/NEXT loop in lines 50-140 makes the computer roll and display four times rather than two times. If you need six, eight or ten dice on display, change the number two in line 50 to three, four or five.

Program Listing

```
10 Q=RND(-TI)
20 PRINT CHR$(147):REM SCREEN CLEAR
30 PRINT"PRESS RETURN"
40 INPUT"TO ROLL FOUR DICE";K$:PRINT:PRINT
45 PRINT CHR$(147):REM SCREEN CLEAR
50 FOR L=1 TO 2
60 DL=INT(7*RND(1))
70 IF DL<1 THEN 60
80 DR=INT(7*RND(1))
90 IF DR<1 THEN 80
100 PRINT"**********"
110 PRINT"*";DL;"*";DR;"*"
```

```
120 PRINT"**********"
130 PRINT:PRINT
140 NEXT L
150 PRINT:CLR:GOTO 30
```

5 Secret Message

Secret messages can be lots of fun! They often are composed of codes in which letters of the alphabet have been replaced by numbers.

In this easy-to-use program, the computer generates a list of pseudorandom numbers and assigns one number to each letter of the alphabet. You use the numbers, in lieu of letters, to write notes to your friends.

There is very little chance of the same number being assigned to two different letters because available numbers range from zero to 999.

When typing this program into your computer, be sure to separate the alphabet letters with commas in line 100.

By the way, note the nice two-column screen printing format! Line 250 does that.

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
20 Z=RND(-TI)
100 DATA A,B,C,D,E,F,G,H,I,J,K,L,M,N,
    O,P,Q,R,S,T,U,V,W,X,Y,Z
200 FOR N=1 TO 13
210 C=INT(1000*RND(1))
220 READ L$
230 D=INT(1000*RND(1))
240 READ J$
250 PRINT L$;" ";C,J$;" ";D
260 NEXT N
300 GET K$
310 IF K$="" THEN 300
320 RESTORE
330 GOTO 10
```

Sample Run

```
A   861          B   110
C   73           D   56
E   783          F   714
G   444          H   228
I   160          J   25
K   954          L   386
M   952          N   279
O   944          P   999
Q   869          R   310
S   323          T   401
U   603          V   9
W   286          X   534
Y   352          Z   33
```

6 Sound Off

You can make your computer beep on command.

Line 10 clears the screen. The FOR/NEXT loop in lines 20 to 90 make the VIC sound its built-in beeper 10 times. You can change the number of times the sound is made by changing the number 10 at the end of line 20. Line 30 actually prints the word beepon the screen.Lines 40 and 50 make the beeper sound off.

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
20 FOR L=1 TO 10
30 PRINT"BEEP"
40 POKE 36878,10
50 POKE 36874,241
60 FOR T=1 TO 300:NEXT T
70 POKE 36878,0
80 FOR T=1 TO 300:NEXT T
90 NEXT L
```

7 R2D2 Sound Effect

The computer does its best to emulate the cuddly little robot. Now, where's C3PO?

Program Listing

```
10 PRINT CHR$(147)
20 PRINT"****************"
30 PRINT"* ROBOT SOUND *"
40 PRINT"****************"
50 POKE 36878,INT(14*RND(1))
60 POKE 36874,240+INT(16*RND(1))
70 FOR T=1 TO INT(100*RND(1)):NEXT T
80 POKE 36878,0
90 GOTO 50
```

8 Piano

Just about the world's simplest piano. Play the letters C, D, E, F, G, A, and B. This program lets you practice, up and down, up and down the musical scale, over and over again.

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
20 POKE 36878,15
30 K$=""
100 GET K$
110 IF K$="" THEN 100
120 IF K$="C" THEN N=225:GOTO 200
130 IF K$="D" THEN N=228:GOTO 200
140 IF K$="E" THEN N=231:GOTO 200
150 IF K$="F" THEN N=232:GOTO 200
160 IF K$="G" THEN N=235:GOTO 200
170 IF K$="A" THEN N=237:GOTO 200
```

```
180 IF K$="B" THEN N=239:GOTO 200
190 GOTO 100
200 PRINT TAB(8);"PIANO"
210 PRINT TAB(10);K$
220 POKE 36874,N
230 FOR T=1 TO 180:NEXT T
240 POKE 36874,0
250 GOTO 10
```

9 Mystery Clues

Want to create your own murder mystery? Figure out whodunit and write your program backwards from there. When your players make wrong guesses, give them tantalizing clues.

Here's a short program which you can load into your computer in a matter of minutes. Key it in and try it out. It shows how you can add clues to your mysteries.

For simplicity, we assume here the Butler did it. Note that, in line 20, we are making him equal to X$. At line 30, the computer stops to ask you whom you think did it. Your answer is recorded in A$.

In line 40, your answer, lodged in A$, is compared with the computer's already-certain knowledge that the Butler did it. A$ is compared with X$. If they agree, and only if they agree, the computer displays the message, "You guessed it." If you got it right, things will end right there.

If, however, you missed it, program execution (sorry about using that word in a murder mystery!) drops to line 50 where we hear the computer, "Clue: servant." After deftly dropping that clue, the computer moves back and runs through the whole affair another time. It will keep running through it until you answer, "Butler," in response to its question in line 30.

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
```

```
20 X$="BUTLER"
30 INPUT"WHODUNIT";A$
40 IF X$=A$ THEN 100
50 PRINT:PRINT"CLUE:","SERVANT":PRINT
   :GOTO 20
100 PRINT CHR$(147)
110 PRINT"YES"
120 PRINT"YOU GUESSED IT"
130 POKE 36878,10:POKE 36874,240
140 FOR T=1 TO 99:NEXT T
150 POKE 36878,0:POKE 36874,0
160 END
```

10 Original Hi/Lo Game

Here it is. Where everybody started in microcomputer programming back in the Seventies. The first game ever played was a high-low guess-the-number routine.

The computer selects a secret number. You try to guess it. The computer tells you whether or not you are too high, too low, or right on the number.

Here's how it works: the secret number can be zero to 1000. Line 100 generates a random number (the secret number) and stores it. Line 210 asks you to guess the number.

Lines 300-310 decide if you are right or wrong. Line 220 keeps track of the number of attempts.

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
20 Z=RND(-TI)
30 T=0
100 R=INT(1000*RND(1))
200 PRINT"I HAVE A SECRET NUMBER"
210 INPUT"CAN YOU GUESS IT";B
220 T=T+1
```

```
230 PRINT:PRINT:PRINT"THAT WAS TRY NO.";T
300 IF B>R THEN PRINT:INPUT"TOO HIGH,
    GUESS AGAIN";B:GOTO 220
310 IF B<R THEN PRINT:INPUT"TOO LOW,GUESS
    AGAIN";B:GOTO 220
400 PRINT CHR$(147)
410 PRINT"YES, YOU GOT IT"
420 PRINT R;"IS THE NUMBER"
430 PRINT"YOU GOT IT"
440 PRINT"IN";T;"TRIES"
450 FOR L=1 TO 10:PRINT:NEXT L
460 CLR
470 GOTO 30
```

Sample Run

```
I HAVE A SECRET NUMBER
CAN YOU GUESS IT 500
THAT WAS TRY NO. 1
TOO HIGH,GUESS AGAIN 250
THAT WAS TRY NO. 2
TOO LOW,GUESS AGAIN 400
THAT WAS TRY NO. 3
TOO HIGH,GUESS AGAIN 300
THAT WAS TRY NO. 4
TOO LOW,GUESS AGAIN 290
THAT WAS TRY NO. 5
TOO LOW,GUESS AGAIN 350
THAT WAS TRY NO. 6
TOO HIGH,GUESS AGAIN 340
THAT WAS TRY NO. 7
TOO HIGH,GUESS AGAIN 330
THAT WAS TRY NO. 8
TOO HIGH,GUESS AGAIN 320
THAT WAS TRY NO. 9
TOO LOW,GUESS AGAIN 310
THAT WAS TRY NO. 10
TOO LOW,GUESS AGAIN 325
THAT WAS TRY NO. 11
TOO LOW,GUESS AGAIN 327
THAT WAS TRY NO. 12
TOO HIGH,GUESS AGAIN 328
```

```
THAT WAS TRY NO. 13
TOO HIGH,GUESS AGAIN 326
THAT WAS TRY NO. 14

YES, YOU GOT IT
 326 IS THE NUMBER
YOU GOT IT
IN 14 TRIES
```

11 Code Groups

Need some secret codes for your latest sensitive mission? How about sets of five random letters for use in Morse code practice?

This program has the computer generate an endless string of random combinations of five letters.

Program Listing

```
10 PRINT CHR$(147):REM SCREEN CLEAR
20 FOR L=1 TO 5
30 N=INT(91*RND(1))
40 IF N<65 THEN 30
50 PRINT CHR$(N);
60 NEXT L
70 PRINT:GOTO 20
```

Sample Run

```
JXSRV    BMLHJ    QSYRI
SXYKR    USGOO    NLLML
NAEHB    YRSKS    VZQOR
TQVSM    HTSKQ    UFAPA
RPVKQ    UDVUN    YNBNR
AWLZV    RSUHF    TDWSO
ZNAWP    ICHVI    NPGGH
YTDMJ    IMLFE    VEMFT
RKEZK    BYWHN    GULFD
```

Program Listing

```
1 REM******THIS PROGRAM LISTING HAS THE
2 REM******COMPUTER PRINT ON A LINE PRINTER
3 REM******WHICH MUST BE CONNECTED TO THE
  COMPUTER
4 REM******TO USE THIS PROGRAM
5 REM*************************************************
8 OPEN 3,4
9 CMD 3
10 PRINT CHR$(147):REM SCREEN CLEAR
20 FOR L=1 TO 5
30 N=INT(91*RND(1))
40 IF N<65 THEN 30
50 PRINT CHR$(N);
60 NEXT L
70 PRINT:GOTO 20
```

Sample Run

```
AIURR
ZNCBS
RGHRF
BTHLN
XQJRT
KLZZN
GDAES
VVYJK
MCTFR
FXYMY
```

12 60-Second Timer

A one-minute timer can be very handy for fun-n-games. This easy-to-use clock "ticks" as it counts off seconds up to 60. When it reaches 60 seconds, it rings an alarm.

The number of seconds counted can be changed by

changing the number 60 in line 20.

The clock can be calibrated by changing the number 800 in line 50. Line 50 is a time-delay loop set for approximately one second.

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
15 POKE 36878,15
20 FOR T=1 TO 60
30 POKE 36874,255:POKE 36874,0
40 PRINT T,"SECONDS"
50 FOR L=1 TO 800:NEXT L
60 NEXT T
70 POKE 36874,255
80 FOR T=1 TO 500:NEXT T
90 POKE 36874,0
```

Sample Run

```
 1              SECONDS
 2              SECONDS
 3              SECONDS
 4              SECONDS
 5              SECONDS
 6              SECONDS
 7              SECONDS
 8              SECONDS
 9              SECONDS
10              SECONDS
11              SECONDS
12              SECONDS
13              SECONDS
14              SECONDS
15              SECONDS
16              SECONDS
17              SECONDS
18              SECONDS
19              SECONDS
20              SECONDS
21              SECONDS
22              SECONDS
```

13 Find Highest/Lowest

Suppose we have a list of people and each person has been assigned a number or score. This program accepts the names and scores and sorts out the persons with the highest and lowest scores. Here's how it works.

Lines 20-90 take in the info on each person. As each person's score is entered, lines 50-80 determine if it is higher or lower than all previous scores. If higher or lower, it is so noted.

To complete data entry, simply press RETURN without data. That will prompt the computer, at lines 110 - 140, to print the lowest score and the highest score.

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
20 INPUT"NAME: ";N$
30 IF N$="" THEN 100
40 INPUT"SCORE: ";S
50 X=X+1
60 IF X=1 THEN LS=S:LN$=N$:HS=S:HN$=N$
70 IF S<LS THEN LS=S:LN$=N$
80 IF S>HS THEN HS=S:HN$=N$
90 N$="":GOTO 20
100 PRINT:PRINT:PRINT
110 PRINT LN$;" WAS LOW"
120 PRINT"AT";LS
130 PRINT HN$;" WAS HIGH"
140 PRINT"AT";HS
150 PRINT:PRINT:PRINT:PRINT
160 INPUT"PRESS RETURN FOR MORE";K$
170 CLR:GOTO 10
```

14 Sorting Scores

Here's how to sort a set of scores. Any numbers can be used. Zero is assumed to be lower than any positive

number and a negative number is lower than zero.

Key in as many numbers as you like. Then key a zero when you want your computer to compute final results. Obviously, a zero cannot be in the set of numbers you are sorting since we use zero to get out of the input loop.

At the end of the RUN, the computer will tell you which number is lowest and which is highest.

Program Listing

```
10 PRINT CHR$(147):REM SCREEN CLEAR
20 INPUT"SCORE:";N
30 IF N=0 THEN 100
40 S=S+1
50 IF S=1 THEN LN=N:HN=N
60 IF N<LN THEN LN=N
70 IF N>HN THEN HN=N
80 GOTO 20
100 PRINT CHR$(147)
110 PRINT"LOW NUMBER: ";LN
120 PRINT"HIGH NUMBER:";HN
130 FOR L=1 TO 10:PRINT:NEXT L
140 PRINT"FOR A DIFFERENT"
150 PRINT"SET OF NUMBERS"
160 PRINT"PRESS ANY KEY"
170 GET K$
180 IF K$="" THEN 170
190 CLR:GOTO 10
```

Sample Run

```
SCORE: 789
SCORE: 654
SCORE: 123
SCORE: 468
SCORE: 642
SCORE: 999
SCORE: 111
SCORE: 195
SCORE: 473
SCORE: 0
```

```
LOW NUMBER:  111
HIGH NUMBER: 999

FOR A DIFFERENT
SET OF NUMBERS
PRESS ANY KEY
```

15 Keeping Game Scores

Writing a computer football game? Spelling bee? Cave adventure? No matter what kind of fun you are preparing, you'll need a way to keep score. Here's how.

The wealthy English duke has just been killed in our little mystery game. In lines 10 through 160 of our program listing, below, you play the game, attempting to find out whodunit.

The trick here is in the scorekeeping. Note line 150. If you guessed correctly in response to the query in line 140, at line 150 the computer will give you credit by adding one point to your score stored in memory location R. It does that by comparing your line 140 answer stored in P$ with the correct answer stored in A$.

If you blew it and guessed wrong, the program drops below line 150 to line 160 where it increases your "wrong score" by adding one point to W.

If you got a W+1 at line 160, the program moves back to line 90 and gets you to try again. If you scored a victory and got an R+1 at line 150, the program jumps to line 200 where it stops to display your total right and wrong score. After that, it's back to line 20 for a complete new run-through.

Program Listing

```
10 PRINT CHR$(147):Z=RND(-TI)
20 S=INT(7*RND(1))
30 IF S<1 THEN 40
40 FOR L=1 TO S
```

```
50 READ A$
60 NEXT L
70 RESTORE
80 PRINT"**********************"
90 PRINT"WHO KILLED THE DUKE?"
100 READ B$,C$,D$,E$,F$,G$
110 RESTORE
120 PRINT:PRINT"WAS IT THE..."
130 PRINT B$,C$,D$,E$,F$,G$
140 PRINT:INPUT"WHODUNIT";P$
150 IF A$=P$ THEN R=R+1:PRINT
    :PRINT"OKAY, YOU ARE RIGHT"
    :PRINT"IT WAS THE ";A$:GOTO 200
160 PRINT CHR$(147):PRINT"NO, NOT THE "
    ;P$:W=W+1:PRINT:GOTO 90
200 PRINT:PRINT"YOUR SCORE IS..."
210 PRINT R;"RIGHT ";W;"WRONG"
220 FOR L=1 TO 8:PRINT:NEXT L
230 PRINT"LET'S PLAY AGAIN"
240 CLR
250 GOTO 20
300 DATA BUTLER, NANNY, GARDNER,BURGLAR
    ,SON,WIFE
```

16 Batting Average

Once you know the number of times you were right and wrong in a game, as in Tip Number 15, it's fun to convert those raw numbers to a batting average. Numbers right and numbers wrong take on a new meaning when changed to a batting average. Folks seem to be able to understand a batting average better.

Our program, starting at line 900, is a partial listing designed to be tacked onto the end of your longer game program to display the final results of play. It will show the number of tries, number of right answers, percentage right, and batting average.

You'll want to test load this program so add lines 10,

800,810 and 820 as shown. Lines 800, 810 and 820 will give you the R and T values you'll need going into the program at line 900.

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
800 INPUT"NUMBER RIGHT: ";R
810 INPUT"NUMBER OF TRIES:";T
820 PRINT CHR$(147)
900 PRINT R;"RIGHT"
910 PRINT" IN";T;"TRIES"
920 D=R/T:P=100*D:B=INT(10*P)
930 PRINT
940 PRINT P;"PERCENT"
950 PRINT
960 PRINT" YOU ARE BATTING";B
```

17 Computer Rating Service

Of course, once you know a player's batting average it still might need some interpretation. In this program, the computer takes a look at a batting average and makes a comment.

Remember that this listing, starting here with line 800, is a partial program to be tacked on the end of a longer game. Note that, at 800, you already have values for G (number right) and E (number of tries). Line 810 converts those raw numbers to a batting average (H).

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
700 INPUT"NUMBER RIGHT: ";G
710 INPUT"NUMBER TRIES: ";E
720 PRINT CHR$(147)
800 PRINT G;"RIGHT",,"IN";E;"TRIES"
810 H=INT((G/E)*1000)
```

```
820 PRINT"YOU ARE BATTING";H
830 PRINT"YOU ARE ";
840 IF H<100 THEN PRINT"THE PITS":GOTO 910
850 IF H<300 THEN PRINT"POOR":GOTO 910
860 IF H<500 THEN PRINT"AVERAGE":GOTO 910
870 IF H<700 THEN PRINT"TOP NOTCH":GOTO 910
880 IF H<900 THEN PRINT"DAMN NEAR PERFECT"
    :GOTO 910
890 PRINT"HALL OF FAME"
900 PRINT"MATERIAL"
910 FOR L=1 TO 10:PRINT:NEXT L
920 INPUT"FOR MORE PRESS RETURN";K$
930 CLR:GOTO 10
```

18 Box Score

To dress up scores during and at the end of a game program, use this method of putting those scores in a box. The box around the score will highlight it and jazz up your video display.

Program Listing

```
 10 PRINT CHR$(147)
 20 PRINT"TYPE A NAME AND SCORE"
 30 PRINT"OF EXACTLY TEN"
 40 PRINT"CHARACTERS, INCLUDING SPACES"
 50 PRINT:INPUT T$
 60 IF LEN(T$)>10 THEN PRINT"TOO LONG"
    :GOTO 50
 70 IF LEN(T$)<10 THEN PRINT"TOO SHORT"
    :GOTO 50
900 PRINT CHR$(147);CHR$(28);CHR$(128)
910 PRINT TAB(5);CHR$(111);
920 FOR L=1 TO 10:PRINT CHR$(163);:NEXT L
930 PRINT CHR$(112)
940 PRINT TAB(5);CHR$(165);
950 PRINT TAB(4);T$;
```

```
960 PRINT TAB(16);CHR$(167)
970 PRINT TAB(5);CHR$(108);
980 FOR L=1 TO 10:PRINT CHR$(164);:NEXT L
990 PRINT TAB(16);CHR$(186)
1000 PRINT CHR$(144)
1010 PRINT CHR$(144)
1100 GET K$
1110 IF K$="" THEN 1100
1120 CLR
1130 GOTO 10
```

Sample Run

```
TYPE A NAME AND SCORE
OF EXACTLY TEN
CHARACTERS, INCLUDING SPACES

WALLY 4321


TYPE A NAME AND SCORE
OF EXACTLY TEN
CHARACTERS, INCLUDING SPACES

ED 7654321


TYPE A NAME AND SCORE
OF EXACTLY TEN
CHARACTERS, INCLUDING SPACES

ANDREW 987
```

Text On Text

19 Create a Quiz

One of the most fascinating uses for your computer is in having it carry on a video conversation with your friends, relatives and neighbors. One useful way to promote such conversation is through a quiz. An instructional, educational quiz, such as we have here.

Quiz data—the computer's storehouse of knowledge—is in lines 20 to 70. Be careful, when you type them into your computer, to include the commas separating the two halfs of each data line. Spelling and spacing must be exact.

Lines 90 and 100 obtain a random number in the range of 1 to 11. Line 110 selects the data line for a question. Lines 120 through 140 get the appropriate word FIRST, SECOND, THIRD, FOURTH, FIFTH or SIXTH from the selected data line. Lines 160 to 180 print the quiz question on the screen, while line 190 reads the DATA line to learn the correct answer. You provide your response when the computer asks for it at line 200. Lines 220-240 decide whether you are right or wrong.

Of course, the quiz can be made much longer. In this example, it could be expanded to encompass all past U.S. presidents.

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
20 DATA FIRST,GEORGE WASHINGTON
30 DATA SECOND,JOHN ADAMS
40 DATA THIRD,THOMAS JEFFERSON
50 DATA FOURTH,JAMES MADISON
60 DATA FIFTH,JAMES MONROE
70 DATA SIXTH,JOHN QUINCY ADAMS
75 PRINT"HOW MANY"
80 PRINT"U.S. PRESIDENTS"
85 PRINT"CAN YOU NAME?"
90 R=INT(12*RND(1))
100 IF R<1 THEN 90
110 IF INT(R/2)=R/2 THEN R=R+1
```

```
120 FOR L=1 TO R
130 READ S$
140 NEXT L
150 PRINT:PRINT
160 PRINT"WHO WAS THE"
170 PRINT S$;" PRESIDENT"
180 PRINT"OF THE U.S."
190 READ C$
200 INPUT D$
210 FOR L=1 TO 11:PRINT:NEXT L
220 IF D$=C$ THEN PRINT"THAT'S CORRECT"
    :GOTO 240
230 PRINT"THAT'S WRONG"
240 PRINT"THE ";S$
250 PRINT"PRESIDENT WAS"
260 PRINT C$
270 RESTORE
280 PRINT:PRINT
290 GOTO 90
```

20 Killing Time

Sometimes, it may seem to you as if the computer will never get to the result of a job. You understand the processing delay but your non-computer friends may not. They could be confused by the wait and think the computer is "broken."

To keep their minds off the slowness, give them something to look at while the computer is "thinking."

The added, extra lines, numbered 50, 60, 70 and 80, take up more processing time but make for less confusion. Computing may take a bit longer but your fun will be increased.

If you delete lines 50-80 you'll see how the program runs faster but the blank screen is confusing.

Program Listing

```
10 PRINT CHR$(147):REM SCREEN CLEAR
```

```
20 INPUT"GIVE ME A NUMBER:";N
30 FOR L=1 TO N
40 X=X+L
50 FOR T=1 TO 100:NEXT T
60 PRINT CHR$(147):REM CLEAR SCREEN
70 FOR T=1 TO 100:NEXT T
80 PRINT"I AM THINKING"
90 NEXT L
100 PRINT CHR$(147):REM SCREEN CLEAR
110 PRINT"I HAVE THE ANSWER"
120 PRINT
130 PRINT"THE TOTAL OF"
140 PRINT"ALL NUMBERS FROM"
150 PRINT"1 TO";N
160 PRINT"IS";X
170 PRINT:PRINT:PRINT
180 CLR
190 GOTO 20
```

21 Word-Error Trapping

Error trapping is available for strings. Suppose the program, as in this example, asks at line 10 for a word. It is looking for YES or NO. If it gets a YES, then line 20 sees that it got what it wanted and moves operations along to line 100.

If it gets a NO, then line 20 hasn't received what it wants so program execution moves on to line 30. Here, at line 30, the program finds something useful and shoots operations down to line 200.

If, however, neither YES nor NO were entered at line 10, then neither lines 20 nor 30 would be satisfied so action would drop to line 40. Here, the error is trapped by commanding the operator to give one of the two correct answers. Then, at line 80, the operation is returned to line 10 for a new try at the correct input.

Program Listing

```
5 PRINT CHR$(147):REM CLEAR SCREEN
10 INPUT"WANNA PLAY AGAIN";A$
20 IF A$="YES" THEN 100
30 IF A$="NO" THEN 200
40 PRINT
50 PRINT"PLEASE ANSWER"
60 PRINT"ONLY YES OR NO"
70 PRINT
80 GOTO 10
100 PRINT"THANK YOU FOR THAT ";A$
110 PRINT:PRINT:PRINT
120 GOTO 10
200 PRINT
210 PRINT"THANK YOU FOR THAT ";A$
220 PRINT"THIS RUN IS NOW AT AN END"
230 END
```

22 Character Numbers

This brief program displays the ASCII value for each keyboard character, side-by-side with the character it stands for. You will be able quickly to tell what each number prints.

Line 40 is a timing loop to slow down the presentation so you can digest the information. To make it even slower, increase the number 400 in line 40. To make it faster, decrease the number 400 in line 40.

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
20 FOR N=33 TO 127
30 PRINTN,CHR$(N):PRINT
40 FOR T=1 TO 400:NEXT T
50 IF N=255 THEN 20
60 NEXT N
70 FOR N=161 TO 255
80 GOTO 30
```

23 One-Time Password

If you don't want unauthorized use of your programs, insert a requirement that a user know a password. This particular routine allows only one try at entering a correct password.

For our password, we have selected "elephant" and stored it in line 30. You can change the password to whatever you like.

If a correct attempt at entering the password is made, program action will progress to line 100. Otherwise, action drops to line 40 and action ends.

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
20 INPUT"WHAT IS THE PASSWORD";A$
30 IF A$="ELEPHANT" THEN 100
40 PRINT"WRONG"
50 END
100 PRINT"YOU GOT IT RIGHT"
110 PRINT"NOW THE PROGRAM WOULD RUN"
```

24 Three-Tries Password

Here the software lets you try three times to enter the correct password. You don't get to go forward with the program if you don't get it right in three tries.

Again the password is "elephant" and is stored in line 30. You can change the password to whatever suits you.

Lines 40 to 60 allow the three attempts. If no good after three tries, then END.

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
20 INPUT"WHAT IS THE PASSWORD";A$
30 IF A$="ELEPHANT" THEN 100
40 B=B+1
50 IF B=3 THEN PRINT"WRONG":END
60 GOTO 20
100 PRINT"YOU GOT IT RIGHT"
110 PRINT"NOW THE PROGRAM WOULD RUN"
```

25 Multiple Passwords

Here's a really complex password entry system. It has a unique "account number" and a password for each person. This will allow several different persons access to the program but each person will have a different combination to the lock!

account number	password
12345	zebra
23456	goose
34567	trout
45678	snake

Each individual user must correctly enter his unique account number and then his own personal password. If account number is wrong, then the password never can be right. If account number is okay but password doesn't match, the user gets no run.

You can add users to this program by adding lines to the 300-340 subroutine.

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
20 INPUT"YOUR ACCOUNT NUMBER";UA
30 GOSUB 300
```

```
40 FOR L=1 TO 3
50 INPUT"PASSWORD";PS$
60 IF PS$="" THEN 50
70 IF PS$=PW$ THEN 110
80 NEXT L
90 PRINT"YOU WERE INCORRECT"
100 END
110 PRINT"YOU GOT IT ALL RIGHT"
120 PRINT"NOW THE PROGRAM WOULD RUN"
130 END
300 IF UA=12345 THEN PW$="ZEBRA"
310 IF UA=23456 THEN PW$="GOOSE"
320 IF UA=34567 THEN PW$="TROUT"
330 IF UA=45678 THEN PW$="SNAKE"
340 RETURN
```

26 Name In A Box

Put your name up in lights! Or, at least, on the video display screen of your computer.

This short program creates a box on the screen and puts a name you have specified into that box. The name is highlighted.

You can change what the box is composed of by changing the asterisks in lines 70, 100 and 120.

Program Listing

```
10 PRINT CHR$(147)
15 PRINT"WHAT IS YOUR NAME"
20 INPUT N$
30 LN=LEN(N$)
40 LT=LN+4
50 PRINT CHR$(147)
60 FOR L=1 TO LT
70 PRINT "*";
80 NEXT L
90 PRINT
```

```
100 PRINT"* ";N$;" *"
110 FOR L=1 TO LT
120 PRINT "*";
130 NEXT L
140 FOR L=1 TO 10:PRINT:NEXT L
150 CLR
160 GOTO 15
```

Sample Run

```
WHAT IS YOUR NAME
ANDREW

************
* ANDREW *
************


WHAT IS YOUR NAME
ED

********
* ED *
********
```

27 Entering: Letter Stop

One way to conclude an input series, and get out of its entry loop, is to use a key letter to promote a jump. In this brief example, we input numbers, at line 100, as string values. If we give the computer an X rather than a number, it will jump down to line 200 for new action.

Numbers keyed in are stored first as strings. Then line 120 changes them to number values for the addition in line 130.

Program Listing

```
10 PRINT CHR$(147)
```

```
100 PRINT"GIVE ME A NUMBER"
105 INPUT A$
110 IF A$="X" THEN 200
120 B=VAL(A$)
130 C=C+B
140 GOTO 100
200 PRINT CHR$(147)
210 PRINT"THE TOTAL OF"
220 PRINT"THOSE NUMBERS IS"
230 PRINT C
300 PRINT:PRINT:PRINT
310 C=0
320 GOTO 100
```

28 Entering: Zero Stop

Here's another way to conclude an entry loop: have the computer be on the lookout for a plain zero. When a zero is entered, the computer will jump out of the entry cycle and on to further action.

This program totals numbers as they are added and accumulates them in memory location B. If one of the numbers entered is a zero alone, then line 110 will spot it and send the computer on down to line 200, breaking the entry cycle.

Naturally, you can't use a zero in a string of numbers to be added since zero causes the computer to quit entering and get on with displaying.

Program Listing

```
10 PRINT CHR$(147)
20 B=0
100 PRINT"GIVE ME A NUMBER"
105 INPUT A
110 IF A=0 THEN 200
120 B=B+A
130 PRINT
```

```
140 GOTO 100
200 PRINT CHR$(147)
210 PRINT"THE TOTAL OF"
220 PRINT"THOSE NUMBERS IS"
230 PRINT B
300 PRINT:PRINT:PRINT
310 CLR
320 GOTO 20
```

29 Super Reverser

Enter any word or number and find it reversed on the display!

After a run, the computer awaits your press of any key to do another.

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
20 INPUT"WORD/NUMBER:";N$
30 IF N$="" THEN 20
35 PRINT CHR$(147)
40 L=LEN(N$)
45 PRINT"ORIGINAL:";N$
50 PRINT"REVERSED:";
60 FOR Y=L TO 1 STEP-1
70 B$=MID$(N$,Y,1)
80 PRINT B$;
90 NEXT Y
100 FOR K=1 TO 10:PRINT:NEXT K
110 PRINT"PRESS ANY KEY"
120 PRINT"TO DO ANOTHER"
130 GET KY$
140 IF KY$="" THEN 130
150 B$=""
160 GOTO 10
```

30 Marching Numbers

This little program does a big job! It creates the unusual display of numbers from one to nine marching across the screen. Try it; you'll like it.

Program Listing

```
10 PRINT CHR$(147)
20 X=0
30 X=X+1
40 IF X>9 THEN 20
50 PRINT X;
60 GOTO 30
```

31 Superior Decision Maker

Remember that YES/NO Executive Decision Maker which was so popular for the computer? Well, the power in the Commodore computer makes a much-improved decision maker possible.

In this superior edition, a choice of eight replies is possible.

After a run, the computer awaits your press of any key to do another.

Program Listing

```
10 DATA FIRE SOMEONE
20 DATA PASS THE BUCK
30 DATA YES
40 DATA MAYBE
50 DATA REORGANIZE
60 DATA SIT ON IT
70 DATA NO
80 DATA SEE YOUR ANALYST
90 Q=RND(-TI)
```

```
100 PRINT CHR$(147)
110 N=INT(9*RND(1))
120 IF N<1 THEN 110
130 FOR L=1 TO N
140 READ Z$
150 NEXT L
160 PRINT Z$
170 GET K$
180 IF K$="" THEN 170
190 RESTORE
200 GOTO 100
```

32 Sentence Writer

Practice your English!

Exhibit your knowledge of nouns and verbs. This program leads the computer to solicit individual words from you and use those words to create sentences.

Besides helping you better understand verbs, nouns and simple declarative sentence structures, the program demonstrates the computer's ability to simulate conversation and communication.

Lines 20, 30 and 40 take in the words.

You may modify the program to suit your own interests or needs.

Program Listing

```
10 PRINT CHR$(147)
20 INPUT"A PLURAL NOUN";PN$
30 INPUT"A VERB";VB$
40 INPUT"A SINGULAR NOUN";SN$
50 PRINT CHR$(147)
60 PRINT"THE ";PN$;" ";VB$;" ";SN$;"."
70 FOR L=1 TO 10:PRINT:NEXT L
80 PRINT"PRESS RETURN"
90 INPUT"TO DO ANOTHER";K$
100 GOTO 10
```

33 Categorizing

A large quantity of numbers can be categorized and thereby cut down into a smaller quantity of numbers. See our example: it takes test scores and divides them into ranges labeled A, B, C, D, and F.

The program assumes exam or test scores in a range of zero to 100. The letter grades include zero to 59, F; 60-69, D; 71-79, C; 80-89, B; 90-100, A.

Key in as many scores as you like and then enter the letter X to stop the entry cycle.

Lines 100-140 sort all scores into the A through F categories. Lines 150-170 sort highest and lowest scores. Line 200 finds mid-range and average scores.

Program Listing

```
10 PRINT CHR$(147)
20 PRINT"ENTER GROUP OF SCORES"
30 PRINT"FROM 0 TO 100,"
40 PRINT"ONE AT A TIME"
50 PRINT:PRINT"ENTER X"
60 PRINT"AFTER LAST SCORE"
70 PRINT:PRINT
80 INPUT"SCORE";G$
85 IF G$="" THEN 80
90 IF G$="X" THEN 200
100 G=VAL(G$)
110 N=N+1
120 IF G<60 THEN F=F+1:GOTO 150
125 IF G<70 THEN D=D+1:GOTO 150
130 IF G<80 THEN C=C+1:GOTO 150
135 IF G<90 THEN B=B+1:GOTO 150
140 A=A+1
150 IF N=1 THEN L=G:H=G
160 IF G<L THEN L=G
170 IF G>H THEN H=G
180 S=S+G
185 G$=""
190 GOTO 80
```

```
200 P=S/N:M=L+((H-L)/2)
210 PRINT CHR$(147)
220 PRINT"THERE WERE";N;"SCORES"
230 PRINT"RANGING";L;"TO";H
240 PRINT"MID-RANGE:";M
250 PRINT"AVERAGE:  ";P
260 PRINT
270 PRINT"TOTALS EACH LETTER"
280 PRINT
290 PRINT "A",A
300 PRINT "B",B
310 PRINT "C",C
320 PRINT "D",D
330 PRINT "F",F
400 PRINT:PRINT:PRINT
410 PRINT"FOR MORE"
420 INPUT"PRESS RETURN";K$
430 CLR
440 GOTO 10
```

Sample Run

```
ENTER GROUP OF SCORES
FROM 0 TO 100,
ONE AT A TIME

ENTER X
AFTER LAST SCORE


SCORE55
SCORE66
SCORE77
SCORE88
SCORE99
SCOREX

THERE WERE 5 SCORES
RANGING 55 TO 99
MID-RANGE: 77
AVERAGE:   77
```

```
TOTALS EACH LETTER

A               1
B               1
C               1
D               1
F               1


FOR MORE
PRESS RETURN


ENTER GROUP OF SCORES
FROM 0 TO 100.
ONE AT A TIME

ENTER X
AFTER LAST SCORE


SCORE76
SCORE54
SCORE89
SCORE92
SCORE79
SCORE69
SCORE48
SCORE81
SCORE73
SCORE75
SCORE94
SCORE87
SCORE88
SCOREX

THERE WERE 13 SCORES
RANGING 48 TO 94
MID-RANGE:  71
AVERAGE:    77.3076923
```

```
TOTALS EACH LETTER

A               2
B               4
C               4
D               1
F               2


FOR MORE
PRESS RETURN
```

34 Alphabet Soup

Sure, everybody knows there are 26 letters in the alphabet. But, do you know which letter is number 20? Number 5? Number 17? Well, your Computer knows!

Type in this short ready-to-run program. RUN it. The computer will spit out number-and-letter combinations all day long. The number on the left is the position in the alphabet of the letter on the right.

It's a fun way to demonstrate to your friends just how "smart" the computer is!

Program Listing

```
10 PRINT CHR$(147)
20 Z=RND(-TI)
30 P=INT(91*RND(1))
40 IF P<65 THEN 30
50 X$=CHR$(P)
60 PRINT P-64,X$
70 GOTO 30
```

Sample Run

```
17              Q
4               D
```

2	B
23	W
5	E
1	A
3	C
4	D
3	C
16	P
7	G
9	I
13	M
19	S
5	E
14	N
12	L
21	U
9	I
25	Y
14	N
14	N
5	E
12	L

35 Create A Table

This program generates a table of values, as a demonstration on how to set up a table on the video display.

Subroutine lines 900 and 910 generate random numbers in the range of zero to 99. Lines 20 and 30 find how many times through the random number generator it takes to get a number greater than 50. The answer is stored in A.

Lines 40 and 50 do it again and store the answer in B. Lines 60 and 70 do it and store in C.

Line 10 prints the table heading and line 100 displays the results. Line 110 causes the whole operation to repeat until you have a table of 20 lines on the screen.

Program Listing

```
5 PRINT CHR$(147):REM CLEAR SCREEN
7 Q=RND(-TI)
10 PRINT TAB(5);"A";TAB(10);"B";TAB(15);"C"
15 PRINT
20 GOSUB 900
30 IF X>50 THEN A=A+1:GOTO 20
40 GOSUB 900
50 IF X>50 THEN B=B+1:GOTO 20
60 GOSUB 900
70 IF X>50 THEN C=C+1:GOTO 20
100 PRINT TAB(4);A;TAB(9);B;TAB(14);C
110 IF T=19 THEN 200
120 A=0:B=0:C=0:T=T+1:GOTO 20
200 GET K$
210 IF K$="" THEN 200
220 CLR
230 GOTO 5
900 X=INT(100*RND(1))
910 RETURN
```

36 Question & Answer

Here's how to use the DATA statement, and the computer's ability to search for data, to create a Q&A.

We put DATA in lines 20-130. It could be anywhere in the program. For instance, at the end at lines 400-510.

The computer sees two items in each data line. Program lines 140 and 160 force the machine to take only odd-numbered data from the list. That is, S$ in line 180 is always the first piece of data in a data line. And C$ in line 210 is always the second item in a data line. Line 230 checks to see if you answered the line 220 question correctly.

Program Listing

```
10 PRINT CHR$(147)
20 DATA JANUARY,31
30 DATA FEBRUARY,28
40 DATA MARCH,31
50 DATA APRIL,30
60 DATA MAY,31
70 DATA JUNE,30
80 DATA JULY,31
90 DATA AUGUST,31
100 DATA SEPTEMBER,30
110 DATA OCTOBER,31
120 DATA NOVEMBER,30
130 DATA DECEMBER,31
140 R=INT(25*RND(1))
150 IF R<1 THEN 140
160 IF INT(R/2)=R/2 THEN R=R-1
170 FOR L=1 TO R
180 READ S$
190 NEXT L
200 PRINT "MONTH IS ";S$
210 READ C$
220 INPUT"HOW MANY DAYS";D$
230 IF D$=C$ THEN PRINT:PRINT"CORRECT"
    :GOTO 300
240 PRINT:PRINT"WRONG"
300 PRINT"NUMBER OF DAYS IS ";C$
310 RESTORE
350 FOR L=1 TO 5:PRINT:NEXT L
360 CLR
370 GOTO 140
```

Gee Whiz

37 Gee Whiz I: Smart Adder

These six programs, in this section of the book, make up our *Gee Whiz* series. One of the fun ways to use your computer is in wowing your friends. Next time they ask, "But, what can it do?", show them its uncanny abilities at adding, spelling, writing upside down, even cracking jokes. Try these six *Gee Whiz* programs on your friends. You'll love their reactions.

Smart Adder is the first in the series. When your neighbor drops in for a cup of coffee, bring out the computer for a demonstration of its lightning speed.

This program adds long strings of numbers in a flash. You give the computer a number. It starts at 1 and adds all numbers up to and including your number. For instance, if you give it a five, it will add 1 plus 2 plus 3 plus 4 plus 5 and display the result.

Ask your neighbor how fast he or she can add all the numbers to 100. It should take several minutes. While he's working on it, let your computer do it in a split second. Your neighbor's reaction is bound to be, "Gee whiz!"

Program Listing

```
10 PRINT CHR$(147)
20 INPUT"GIVE ME A NUMBER";N
30 IF N<1 THEN 20
40 FOR L=1 TO N:X=X+L:NEXT L
50 PRINT:PRINT"THE TOTAL OF"
60 PRINT"ALL NUMBERS"
70 PRINT"FROM 1 TO";N
80 PRINT"IS";X
90 PRINT:PRINT:PRINT:PRINT
100 CLR:GOTO 20
```

38 Gee Whiz II: Three-Digit Mystery

Have your neighbor secretly select any three-digit number in which all three digits are the same. Then have him tell the computer only the *sum* of those three digits.

The computer will identify his secret number!

Program Listing

```
10 PRINT CHR$(147)
20 PRINT"SELECT A"
30 PRINT"THREE-DIGIT NUMBER"
40 PRINT"WITH ALL THREE"
50 PRINT"DIGITS THE SAME."
60 PRINT
70 PRINT"ADD THE THREE"
80 PRINT"DIGITS TOGETHER"
90 PRINT
100 PRINT"WHAT IS THE SUM OF"
110 INPUT"THE THREE DIGITS";N
120 IF N<3 OR N>27 THEN 100
130 Q=37*N
140 PRINT:PRINT
150 PRINT"YOUR NUMBER IS";Q
160 FOR L=1 TO 7:PRINT:NEXT L
170 CLR:GOTO 20
```

39 Gee Whiz III: Yes/No Decision Maker

This is handy for the busy executive who doesn't have time for decisions.

Line 10 clears the screen. Line 20 generates a random number from zero to 100. Line 20 selects a *yes*

answer if the random number is greater than 49. Otherwise, line 30 chooses a *no* answer. Press any key to repeat the run.

Program Listing

```
10 PRINT CHR$(147):Z=RND(-TI)
20 IF (100*RND(1))>49 THEN PRINT"YES"
   :GOTO 40
30 PRINT"NO"
40 GET K$:IF K$="" THEN 40
50 GOTO 10
```

40 Gee Whiz IV: First Alphabet Spotter

There are 26 letters in the alphabet. Each has a number. For instance, number 1 is A. Number 20 is T. This *Gee Whiz* program has the computer ask you for a number from 1 to 26 and then, faster than a jackrabbit, tell you what letter it goes with.

Naturally, you'll know how it works but to your non-computer friends it will seem like the computer is a genius!

Program Listing

```
10 PRINT CHR$(147)
20 PRINT"GIVE ME THE NUMBER"
30 PRINT"OF A LETTER"
40 PRINT"FROM THE ALPHABET"
50 INPUT"FROM 1 TO 26";N
55 IF N<1 OR N>26 THEN 50
60 X=N+64
70 PRINT:PRINT:PRINT
80 PRINT"LETTER NUMBER";N;"IS ";CHR$(X)
90 FOR L=1 TO 7:PRINT:NEXT L
100 GOTO 20
```

41 Gee Whiz V: Second Alphabet Spotter

This is a variation on the previous program. This *Gee Whiz* program has the computer ask you for a number from 1 to 26 and then, faster than a jackrabbit, tell you what letter it goes with.

Program Listing

```
10 PRINT CHR$(147)
20 PRINT"GIVE ME THE NUMBER"
30 PRINT"OF A LETTER"
40 PRINT"FROM THE ALPHABET"
50 INPUT"FROM 1 TO 26";N
60 IF N<1 OR N>26 THEN 50
70 FOR L=1 TO N
80 READ A$
90 NEXT L
100 PRINT:PRINT:PRINT
110 PRINT"LETTER NUMBER";N;"IS ";A$
200 FOR L=1 TO 7:PRINT:NEXT L
210 RESTORE
220 GOTO 20
300 DATA A,B,C,D,E,F,G,H,I,J,K,L,M
310 DATA N,O,P,Q,R,S,T,U,V,W,X,Y,Z
```

42 Gee Whiz VI: Guess The Number

Here it is! The world's oldest, longest running, most popular game: Guess The Number.

When you start the program running, the computer thinks of a number and stores that away. You try to guess the number. If your number is too high, the computer says, "TOO HIGH."

If you are too low, the computer will report "TOO LOW." The possible numbers range from zero to 100.

Program Listing

```
10 PRINT CHR$(147):Z=RND(-TI)
20 FOR L=1 TO 20:PRINT"*";:NEXT L:PRINT
30 N=INT(101*RND(1))
40 INPUT"GUESS THE NUMBER";G
50 IF G>N THEN PRINT"TOO HIGH":GOTO 40
60 IF G<N THEN PRINT"TOO LOW":GOTO 40
70 PRINT:PRINT"RIGHT !"
80 PRINT"LET'S GO AGAIN"
90 GOTO 20
```

Sample Run

```
**********
GUESS THE NUMBER 100
TOO HIGH
GUESS THE NUMBER 1
TOO LOW
GUESS THE NUMBER 33
TOO HIGH
GUESS THE NUMBER 9
TOO LOW
GUESS THE NUMBER 26
TOO HIGH
GUESS THE NUMBER 16
TOO HIGH
GUESS THE NUMBER 12
TOO LOW
GUESS THE NUMBER 13

RIGHT !
LET'S GO AGAIN
**********
```

Number Crunching

43 Memory Tester

Most everybody can remember numbers. At least short numbers with few digits. But how long a number can you recall in a flash?

The computer will briefly display a number. It then will remove the number from your view and ask you to repeat what it was. If you miss three times, the computer will tell you to FORGET IT, give you your score and end the game. Then it will start over.

On the other hand, if you recall correctly, the computer will say so and then give you a new number. The new number will have more digits than the previous number. Each time you guess correctly, the number gets longer.

No matter how good you are, at some point you won't be able to recall *all* the digits in proper sequence.

How many digits can you quickly recall?

Program Listing

```
 10 PRINT CHR$(147):REM CLEAR SCREEN
 20 Q=RND(-TI):Z=1
 30 S=10*RND(1)
 40 N=INT(S*Z)
 50 PRINT"REMEMBER----->";N
 60 IF W=3 THEN PRINT"FORGET IT !":GOTO 140
 70 FOR T=1 TO 1000:NEXT T
 80 PRINT CHR$(147):REM SCREEN CLEAR
 90 INPUT"WHAT WAS IT";S
100 IF S<>N THEN PRINT"***YOU ARE WRONG***"
    :W=W+1:GOTO 60
110 PRINT"***YOU ARE RIGHT***":R=R+1:Z=Z*10
120 PRINT R;"RIGHT SO FAR"
125 FOR T=1 TO 1000:NEXT T
130 PRINT:GOTO 30
140 PRINT:PRINT"YOU HAD";R;"RIGHT"
150 PRINT:PRINT"LET'S START OVER"
160 PRINT"PRESS ANY KEY"
170 GET K$
180 IF K$="" THEN 170
190 CLR:GOTO 10
```

44 Number Reverser

Give your computer any three-digit number and, as a result of this particular programming trick, it will reverse the original number. For example, 789 will be transformed into 987. Or 123 into 321. It takes your three-digit number apart and reassembles it in reverse order.

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
20 PRINT"TYPE A THREE-DIGIT"
30 INPUT"NUMBER";N
40 IF N<100 OR N>999 THEN PRINT:GOTO 20
50 A=INT(N/100)
60 B=INT(10*((N/100)-A))
70 C=VAL(RIGHT$(STR$(N),1))
80 PRINT:PRINT RIGHT$(STR$(C),1)
   ;RIGHT$(STR$(B),1);RIGHT$(STR$(A),1)
90 PRINT:PRINT:PRINT
100 CLR:GOTO 20
```

45 Exam Score Sorter

A quick way to sort and count a book full of letter grades, this program permits one-key entry of a mixed series of data.

We use the familiar letter grades A, B, C, D, and F. You may substitute any other set of five characters you wish in the IF statements in lines 50 to 90.

The letter X is used to conclude the series and lead the computer to display final results.

You press the appropriate key and the computer knows immediately what grade you have indicated. Key in as many grades as you like in any mixed order.

When you have completed entering all grades, type in the letter X. The computer will report the total of A's, B's, C's, D's, and F's.

We use exam-score sorting as our example here but this same program would be good for data collection in the field in many professions. And you can stretch out the possible categories to 25 or more.

To make the program run again, press the RETURN key on the computer's keyboard.

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
15 PRINT"ENTER LETTER GRADES"
20 PRINT"A,B,C,D,F";
25 GET G$
30 IF G$="" THEN 25
40 IF G$="X" THEN 110
50 IF G$="A" THEN A=A+1:GOTO 10
60 IF G$="B" THEN B=B+1:GOTO 10
70 IF G$="C" THEN C=C+1:GOTO 10
80 IF G$="D" THEN D=D+1:GOTO 10
90 IF G$="F" THEN F=F+1:GOTO 10
100 GOTO 10
110 PRINT CHR$(147):REM SCREEN CLEAR
120 PRINT "A",A
130 PRINT "B",B
140 PRINT "C",C
150 PRINT "D",D
160 PRINT "F",F
170 PRINT:PRINT:PRINT
180 INPUT"FOR MORE PRESS RETURN";K$
190 CLR:GOTO 10
```

46 Number-Error Trapping

Good programs, those which are well written, need *error trapping*. It's a technique for making sure persons communicating with the computer don't key in inappropriate data or make mistakes which would cause computation problems for the computer.

For instance, see the example program here. In line 10 the computer asks for a number. In line 20, if the number is too low, it says so and goes back to line 10 to repeat its request.

At line 30, if the number received at line 10 is too large, it says so and goes back to line 10 for a better choice.

The result is only printed at line 40 when a satisfactory number has been keyed in back at line 10.

You can set your own limits by changing the 10 in line 20 and the 100 in line 30.

Program Listing

```
5 PRINT CHR$(147):REM CLEAR SCREEN
10 INPUT"GIVE ME A NUMBER";A
20 IF A<10 THEN PRINT"TOO LOW":GOTO 10
30 IF A>100 THEN PRINT"TOO HIGH":GOTO 10
40 PRINT A
```

47 Standard Deviation

Here's a way to determine mean and standard deviation. In this particular program, you exit the entry cycle by entering the large number 999999999 (nine 9's) so you can't use 999999999 as one of your data points.

This is a great opportunity to experiment with standard deviation computations. Try a series of data points such as 3, 5, 3, 7, and 4. They should result in

```
DATA POINTS TOTAL 22
MEAN 4.4
VARIANCE 2.23999998
STD DEVIATION 1.49666295
```

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
20 INPUT"DATA POINT:";X
30 IF X=999999999 THEN 60
```

```
40 T=T+X:S=S+X↑2:N=N+1
50 GOTO 20
60 A=T/N:V=S/N-A↑2:D=SQR(V)
70 PRINT:PRINT:PRINT
80 PRINT"DATA POINTS TOTAL";T
90 PRINT"MEAN";A
100 PRINT"VARIANCE";V
110 PRINT"STD DEVIATION";D
120 PRINT:PRINT:PRINT
130 INPUT"FOR MORE PRESS RETURN";K$
140 CLR:GOTO 10
```

Sample Run

```
DATA POINT: 9
DATA POINT: 8
DATA POINT: 7
DATA POINT: 6
DATA POINT: 5
DATA POINT: 4
DATA POINT: 3
DATA POINT: 2
DATA POINT: 999999999

DATA POINTS TOTAL 44
MEAN 5.5
VARIANCE 5.25
STD DEVIATION 2.29128785

FOR MORE PRESS RETURN
```

48 Percentages

Usually it's more convenient to enter percentages as percent rather than having to convert to decimals in your head first. Of course, the computer needs that converted decimal value to do its work. How to get it?

This program does the trick. You give it a percentage

and it converts that to a decimal. The computer does the hard work for you!

Line 30 makes the actual conversion. Use this idea as part of a larger check-balancing, accounting or bookkeeping program and save lots of mental effort.

Program Listing

```
10 PRINT CHR$(147):REM SCREEN CLEAR
20 INPUT"PERCENTAGE";P
30 D=0.01*P
40 PRINT"DECIMAL";D
50 PRINT:PRINT:PRINT
60 CLR:GOTO 20
```

49 Logic Functions

You can make your computer do things based on its decision that something exists. That is, in the first program listing here, it only will print the value of C if it finds that B has an existing value. If B is found to have no value, does not exist, C will not be printed.

The decision is in line 40. The machine only prints C if B does not equal zero. Since, in line 20, we set B = 10, the computer will find that something exists in B and, thus, go ahead and do the work assigned in the last half of line 40. If nothing had been stored in B, the last half of line 40 would have been ignored.

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
20 B=10
30 C=10*B
40 IF B THEN PRINT C
```

In the second program here, the computer only displays the results of the tests in lines 40 and 50 if the results of one or both is "true."

By doing the simple math in your head, you can see that the information in the right-hand side of line 20 is true. The information in the right-hand side of line 30 is false.

Line 20 says that 6 + 8 is greater than 3 times 4. That is, 14 is greater than 12. That is true.

Line 30 says that 5 + 2 is greater than 9 + 2. That is, 7 is greater than 11. That is false.

After reading line 20, the computer will store a 1 in B since the statement is true. Upon reading line 30, the computer will store a zero in C since the statement is false.

As action drops to line 40, the computer will find the 1 it stored in B and, thus, complete the action called for at the right-hand end of line 40. It will display the message, "B OKAY."

At line 50, however, the computer will find "nothing" (zero) in C and will not complete the right-hand end of that instruction. It only will do the right-hand end if it finds something in the left-hand end.

These logic functions are great for quick tests.

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
20 B=(6+8)>(3*4)
30 C=(5+2)>(9+2)
40 IF B THEN PRINT "B OKAY"
50 IF C THEN PRINT "C OKAY"
```

50 Above & Below a Line

Here's a way to count numbers above and below a cut-off line. The computer solicits numbers between 1 and 100. Any numbers you key in which are below 1 or above 100 are trapped out by line 40. Entering a zero ends the input cycle.

Line 50 counts the total numbers. Line 60 counts on-

ly those numbers between 1 and 50. Line 80 counts the numbers from 51 to 100. Lines 90 to 130 present results.

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
20 INPUT"GIVE ME A NUMBER";Z
30 IF Z=0 THEN 80
40 IF Z<1 OR Z>100 THEN 20
50 N=N+1
60 IF Z<51 THEN B=B+1
70 GOTO 20
80 A=N-B
90 FOR L=1 TO 16:PRINT:NEXT L
100 PRINT"NUMBERS TOTAL";N
110 PRINT"1-50",B
120 PRINT"51-100",A
130 FOR L=1 TO 16:PRINT:NEXT L
140 CLR:GOTO 20
```

51 Factoring

This program finds and lists the factors of any number you specify. It can be used as a subroutine in a larger program, with appropriate attention to line numbers, variable names, and RETURN.

The number of individual factors are limited by the DIM statement in line 20.

The list will exclude the number itself divided by 1.

For a quick sample run, try the number 18. You should find factors are 9, 6, 3 and 2.

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
20 DIM Q(650)
30 INPUT"NUMBER";N
40 FOR L=2 TO N/2
```

```
50 M=N/L
60 IF M=INT(M) THEN Q(L)=M
70 NEXT L
80 PRINT:PRINT"FACTORS ARE:"
90 FOR L=1 TO N/2
100 IF Q(L)>1 THEN PRINT Q(L):GOTO 120
110 Z=Z+1
120 NEXT L
130 IF N=1 THEN PRINT"NONE":GOTO 150
140 IF Z=INT(N/2) THEN PRINT"NONE"
```

52 Which is Smallest?

How can the computer tell which number is smaller or larger? Here's how.

Type in the program and RUN it. It will ask for, and accept a continuous string of numbers until you end the input routine by keying in a zero.

Lines 40 to 60 make the decision as to which number is lowest.

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
20 INPUT"GIVE ME A NUMBER";Z
30 IF Z=0 THEN 80
40 N=N+1
50 IF N=1 THEN D=Z
60 IF Z<D THEN D=Z
70 GOTO 20
80 PRINT:PRINT:PRINT
90 PRINT"THE SMALLEST"
100 PRINT"NUMBER IS";D
110 PRINT:PRINT:PRINT
120 INPUT"FOR MORE PRESS RETURN";K$
130 CLR:GOTO 10
```

53 Which is Largest?

Suppose you have a group of numbers and you would like to know which number is largest within the group? Here's a software routine for your computer so it can locate the largest number.

You can key in as many numbers as you wish. To end that entry cycle, type in a zero. The computer will see that zero as its cue to leave the entering routine and get on with computing.

Line 40 tests each new number as you enter it. If a new number is larger, that new number is stored in memory location D. At the end of the entry cycle, the largest number is left stored in D. Line 70 recalls that largest number and prints it.

Program Listing

```
10 PRINT CHR$(147):REM SCREEN CLEAR
20 INPUT"GIVE ME A NUMBER";Z
30 IF Z=0 THEN 60
40 IF Z>D THEN D=Z
50 GOTO 20
60 PRINT:PRINT:PRINT
70 PRINT"THE LARGEST"
80 PRINT"NUMBER IS";D
90 PRINT:PRINT:PRINT
100 INPUT"FOR MORE PRESS RETURN";K$
110 CLR:GOTO 10
```

54 Reciprocals

Key in any number. The computer will display its reciprocal. The actual conversion is done here at line 30.

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
```

```
15 PRINT"NUMBER TO BE CONVERTED"
20 INPUT"TO ITS RECIPROCAL";N
25 IF N=0 THEN PRINT:GOTO 15
30 R=1/N
40 PRINT:PRINT:PRINT
50 PRINT"RECIPROCAL OF";N
60 PRINT"IS";R
70 PRINT:PRINT:PRINT
80 INPUT"FOR MORE PRESS RETURN";K$
90 CLR:GOTO 10
```

55 Dump the Integer

Look at the number 123.456 with an eye toward how to get rid of the portion left of the decimal point. Keep only .456 and dump 123. Here's a short program tc accomplish that.

Try 5.67. It will come out .67. Or 500.5 which will come out .5.

Program Listing

```
10 PRINT CHR$(147):REM SCREEN CLEAR
20 PRINT"GIVE ME A NUMBER"
30 INPUT"WITH A DECIMAL";N
40 X=N-INT(N)
50 PRINT
60 PRINT"THE FRACTIONAL"
70 PRINT"PORTION OF"
80 PRINT N;" IS";X
90 PRINT:PRINT:PRINT
100 INPUT"FOR MORE PRESS RETURN";K$
110 CLR:GOTO 10
```

56 Averages

Key in numbers in any order. A zero will end entry. The computer will tell you the average number of all numbers you entered.

Line 40 finds the total number of all numbers entered. Line 50 finds the total of entered numbers. Line 70 computes the average.

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
20 INPUT"GIVE ME A NUMBER";Z
30 IF Z=0 THEN 70
40 N=N+1
50 T=T+Z
60 GOTO 20
70 A=T/N
100 PRINT:PRINT:PRINT
110 PRINT"THE AVERAGE IS";A
120 PRINT:PRINT:PRINT
130 INPUT"FOR MORE PRESS RETURN";K$
140 CLR:GOTO 10
```

57 Mid-Range Number

Here's how to find the middle of a range of numbers. You key in as many numbers in a series as you wish. After the last number, key in a zero to move the program out of the entry cycle.

Lines 40 to 70 select the highest and lowest numbers in the range. They actually define the range. Then line 90 finds the middle point of that range.

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
```

```
20 INPUT"GIVE ME A NUMBER";Z
30 IF Z=0 THEN 90
40 N=N+1
50 IF N=1 THEN H=Z:L=Z
60 IF Z<L THEN L=Z
70 IF Z>H THEN H=Z
80 GOTO 20
90 M=L+((H-L)/2)
100 PRINT:PRINT:PRINT
110 PRINT"MID-RANGE NUMBER:";M
120 PRINT:PRINT:PRINT
130 INPUT"FOR MORE PRESS RETURN";K$
140 CLR:GOTO 10
```

58 Rounding Off

The technique for rounding off numbers is easy. This program, which can stand alone or be worked into a larger program as a subroutine, rounds a decimal to the nearest whole number.

There are two views on how to round off. One holds that "if the number is more than five, you round up." Which means that exactly 0.5 rounds down.

Another view is that "any number less than five rounds down." In that case exactly 0.5 rounds up.

The first set of program lines below is for the fellow with the "more than five rounds up" idea.

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
20 PRINT"GIVE ME A NUMBER"
30 PRINT"TO BE ROUNDED OFF"
40 INPUT N
50 IF N>INT(N) THEN 80
60 R=N
70 GOTO 130
80 D=N-INT(N)
90 IF D>0.5 THEN 120
```

```
100 R=INT(N)
110 GOTO 130
120 R=INT(N)+1
130 PRINT CHR$(147)
140 PRINT N
150 PRINT"ROUNDS OFF TO"
160 PRINT R
170 PRINT:PRINT:PRINT
180 CLR:GOTO 20
```

The second set of program lines rounds off on the "less than five rounds down" theory.

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
20 PRINT"GIVE ME A NUMBER"
30 PRINT"TO BE ROUNDED OFF"
40 INPUT N
50 IF N>INT(N) THEN 80
60 R=N
70 GOTO 130
80 D=N-INT(N)
90 IF D<0.5 THEN 120
100 R=INT(N)+1
110 GOTO 130
120 R=INT(N)
130 PRINT CHR$(147)
140 PRINT N
150 PRINT"ROUNDS OFF TO"
160 PRINT R
170 PRINT:PRINT:PRINT
180 CLR:GOTO 20
```

59 Two-Digit Round Off

It is possible to round off to the nearest hundredths place. That is, to two digits after the decimal point. Here's how:

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
20 PRINT"GIVE ME A NUMBER"
30 PRINT"TO MORE THAN"
40 INPUT"TWO DECIMAL PLACES";N
50 N$=STR$(N)
60 N$=RIGHT$(N$,(LEN(N$)-1))
70 R=INT(100*N+0.5)/100
80 R$=STR$(R)
90 R$=RIGHT$(R$,(LEN(R$)-1))
100 PRINT CHR$(147)
110 PRINT N;"ROUNDS TO"
120 PRINT R
130 PRINT:PRINT"       OR"
140 PRINT:PRINT"$";N$;" BECOMES"
150 PRINT"$";R$
160 PRINT:PRINT:PRINT
170 INPUT"PRESS RETURN FOR MORE";K$
180 CLR:GOTO 10
```

60 Percent to Decimal

Checking, interest, sales tax, and other financial programs are more "user friendly" if you don't have to make manual conversions in your head. For example, if you know your savings account earns 8 percent interest, and you need to multiply by the decimal value for 8 percent (which is 0.08), it is easier to be able to enter 8 and let the computer figure out the decimal value.

Here's another way to change percentages to decimals inside a program to simplify entry by permitting percents to be entered as simple numbers.

For some examples, try entering a price of 2.50 and a sales tax percentage of 6. Your computer will find the bill totals $2.65. Or try $7.80 and 5 percent tax. The bill will be $8.19. Try $123.75 at 8 percent tax. The bill will total $133.65.

Program Listing

```
10 PRINT CHR$(147):REM SCREEN CLEAR
20 INPUT"PRICE $";P
30 INPUT"SALES TAX %";R
40 T=0.01*R
50 S=T*P:B=P+S
60 S$=STR$(S):B$=STR$(B)
70 S$=RIGHT$(S$,(LEN(S$)-1))
   :B$=RIGHT$(B$,(LEN(B$)-1))
80 PRINT CHR$(147)
90 PRINT"SALES TAX","$";S$
100 PRINT"TOTAL BILL","$";B$
110 PRINT:PRINT:PRINT
120 INPUT"FOR MORE PRESS RETURN";K$
130 CLR:GOTO 10
```

61 Every 10th Answer

This program generates a random number in the range of zero to 999. However, it has a difference. It only shows you every tenth number it generates.

Line 20 generates the numbers. Line 40 selects the tenth number from each set.

Program Listing

```
10 PRINT CHR$(147):X=RND(-TI)
20 T=INT(1000*RND(1))
30 V=V+1
40 IF 0.1*V=INT(0.1*V) THEN PRINT V,T
50 GOTO 20
```

62 Random Sampler

This program strengthens your confidence in the ran-

dom number generator built into your computer.

It generates 100 numbers between zero and 100 and tells you how many of those are above 49 and how many are below 50. See the sample RUN for several sets of results in our recent test.

Program Listing

```
10 PRINT CHR$(147):X=RND(-TI)
20 FOR L=1 TO 100
30 X=INT(100*RND(1))
40 IF X<50 THEN Y=Y+1
50 IF X>49 THEN N=N+1
60 NEXT L
70 PRINT Y;"YES",N;"NO"
80 PRINT:PRINT:PRINT
90 CLR:GOTO 20
```

Sample Run

```
49 YES          51 NO
54 YES          46 NO
52 YES          48 NO
```

63 Random Numbers: Zero To Nine

Although you see four program lines below, what we really have here is a very convenient single-line program for you to insert in a larger game or educational-testing program.

Line 20 is the winner here. It prints a random number from zero to nine every time. For your use here, we print that number on the screen. You could just as easily have the computer store that random number in a memory location for later recall and use.

We have added lines to make your computer

show you a whole series of random numbers from zero to nine. Remember, line 20 is the important single-line program element here.

If you would like random numbers in the range from zero to 99, make it 100* in line 20. For zero to 999, use 1000* in line 20.

Program Listing

```
10 PRINT CHR$(147):X=RND(-TI)
20 PRINT INT(10*RND(1))
30 FOR T=1 TO 500:NEXT T
40 GOTO 20
```

64 Random Numbers: Distribution

Ever wonder how "random" are the numbers generated by the random-number generator in your computer when you use the RND instruction? Try this program.

It generates 100 random numbers in a range from zero to nine and counts how many there are of each number between zero and nine.

By the way, while it is doing that it will display the message "counting" so you can tell it is working.

At the end of its run, the computer prints a neat chart, on the video display, of results.

Program Listing

```
10 PRINT CHR$(147):X=RND(-TI)
20 FOR L=1 TO 100
30 N=INT(10*RND(1))
40 IF N=0 THEN A=A+1
50 IF N=1 THEN B=B+1
60 IF N=2 THEN C=C+1
70 IF N=3 THEN D=D+1
```

```
80 IF N=4 THEN E=E+1
90 IF N=5 THEN F=F+1
100 IF N=6 THEN G=G+1
110 IF N=7 THEN H=H+1
120 IF N=8 THEN I=I+1
130 IF N=9 THEN J=J+1
140 PRINT"COUNTING"
150 NEXT L
160 PRINT CHR$(147):REM SCREEN CLEAR
200 PRINT "0",A
210 PRINT "1",B
220 PRINT "2",C
230 PRINT "3",D
240 PRINT "4",E
250 PRINT "5",F
260 PRINT "6",G
270 PRINT "7",H
280 PRINT "8",I
290 PRINT "9",J
300 PRINT:PRINT:PRINT
310 INPUT"PRESS RETURN FOR MORE";K$
320 CLR:GOTO 10
```

Sample Run

```
COUNTING
COUNTING
COUNTING
COUNTING

0             5
1             14
2             6
3             12
4             11
5             11
6             9
7             13
8             9
9             10

PRESS RETURN FOR MORE
```

65 Random Numbers: Averages

This program generates 100 random numbers and totals them. Then it finds the average of all 100 numbers.

In fact, the average number itself is a useful new random number.

To make the program run again, press the RETURN key on the computer's keyboard.

Program Listing

```
10 PRINT CHR$(147):X=RND(-TI)
20 FOR L=0 TO 99
30 N=INT(10*RND(1))
40 NT=NT+N
50 PRINT"AVERAGING"
60 NEXT L
70 PRINT CHR$(147):REM CLEAR SCREEN
80 AV=NT/100
90 PRINT"TOTAL OF 100"
100 PRINT"RANDOM NUMBERS"
110 PRINT"FROM ZERO TO 9"
120 PRINT"IS";NT
130 PRINT
140 PRINT"AVERAGE IS";AV
150 PRINT:PRINT:PRINT:PRINT
160 INPUT"FOR MORE PRESS RETURN";K$
170 CLR:GOTO 10
```

66 Random Numbers: Sorting High/Low

It's important to be able to sort a group of numbers to see what the highest and lowest values are. This program does that.

The random number generator is in line 30. It gives numbers in a range of zero to 999. Line 50 determines the lowest number in the set and line 60 finds the highest number.

Program Listing

```
10 PRINT CHR$(147):X=RND(-TI)
20 FOR L=0 TO 99
30 N=INT(1000*RND(1))
40 IF L=0 THEN LN=N:HN=N
50 IF N<LN THEN LN=N
60 IF N>HN THEN HN=N
70 PRINT"SORTING"
80 NEXT L
90 PRINT CHR$(147):REM CLEAR SCREEN
100 PRINT"LOW NUMBER:",LN
110 PRINT"HIGH NUMBER:",HN
200 PRINT:PRINT:PRINT:PRINT
210 PRINT"TO DO ANOTHER"
220 PRINT"PRESS ANY KEY"
230 GET K$
240 IF K$="" THEN 230
250 CLR
260 GOTO 10
```

Sample Run

```
SORTING
SORTING
SORTING
SORTING
SORTING
SORTING
SORTING
SORTING
SORTING
SORTING

LOW NUMBER:              41
HIGH NUMBER:             997

TO DO ANOTHER
PRESS ANY KEY
```

Money Matters

67 Money Grows

This section of the book includes a number of programs relating to household money management and to small-business applications. This first program shows you how your money grows when deposited in a savings account at a certain annual interest rate, compounded monthly.

The program will have the computer ask for the initial amount of principal saved by depositing in the account. Then the annual interest rate and the number of months to be displayed. The result of the run is a display of the changing principal as months pass and interest is added on.

Line 10 clears the text screen. Lines 20 to 40 take in data from you. Lines 50 to 90 put out the results. Very handy!

Line 85 is a timing loop to slow down the presentation so you can digest the information. To make it even slower, increase the number 400 in line 85. To make it faster, decrease the number 400 in line 85.

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
20 INPUT"PRINCIPAL $";P
30 INPUT"ANNUAL INTEREST %";R
40 INPUT"NUMBER MONTHS";M
45 PRINT CHR$(147):REM CLEAR SCREEN
50 PRINT"MONTH";SPC(7);"BALANCE"
55 FOR Q=1 TO M
60 I=(P*(0.01*R))/12
70 P=P+I
75 PP=INT(100*P+0.5)/100
80 PRINT Q,PP
85 FOR T=1 TO 400:NEXT T
90 NEXT Q
100 PRINT:PRINT
110 INPUT"FOR MORE PRESS RETURN";K$
120 CLR:GOTO 10
```

Sample Run

```
PRINCIPAL $ 1000
ANNUAL INTEREST % 10
NUMBER MONTHS 12
MONTH          BALANCE
 1                1008.33
 2                1016.74
 3                1025.21
 4                1033.75
 5                1042.37
 6                1051.05
 7                1059.81
 8                1068.64
 9                1077.55
10                1086.53
11                1095.58
12                1104.71

FOR MORE PRESS RETURN
```

68 Shopper's Friend

This program finds the computer asking for certain information and then telling you which product brand name is the best buy.

The computer wil ask for the brand name of a product, the quantity in the product package, and the price of the package. Then it will ask for the name, quantity and price for a second product.

After digesting all this information, it will tell you the brand name of the best-buy product and show you the unit prices for both brand names so you can agree with the computer's judgment.

For example, suppose you were looking at corn flakes in boxes, one by Post and one by Kellogg. Suppose the Post box contained 24 ounces of flakes and was priced on the grocery shelf at $1.98 while the Kellogg box held 18 ounces

and was priced at $1.59. Which would be the better buy based on unit price per ounce of flakes?

Run the data through your computer and you'll find it computes the Post corn flakes to be the best buy with a unit price of 8¢ vs. the Kellogg unit price of 9¢.

By the way, if the unit prices turn out to be equal, the computer will say they are equal.

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
20 PRINT SPC(3);"SHOPPER'S FRIEND"
30 FOR L=1 TO 22:PRINT"*";:NEXT L:PRINT
40 INPUT"FIRST BRAND";X$
50 INPUT"QUANTITY";M
55 IF M=0 THEN 50
60 INPUT"PRICE";N
70 INPUT"SECOND BRAND";Y$
80 INPUT"QUANTITY";Q
85 IF Q=0 THEN 80
90 INPUT"PRICE";R
100 PRINT CHR$(147):REM CLEAR SCREEN
110 IF N/M=R/Q THEN 500
120 IF N/M<R/Q THEN 300
200 PRINT Y$;" IS BEST BUY"
210 GOTO 400
300 PRINT X$;" IS BEST BUY"
400 PRINT
410 PRINT X$;" UNIT=$";INT(100*(N/M)
    +0.5)/100
420 PRINT Y$;" UNIT=$";INT(100*(R/Q)
    +0.5)/100
430 FOR L=1 TO 8:PRINT:NEXT L
440 INPUT"FOR MORE PRESS RETURN";K$
450 CLR
460 GOTO 10
500 PRINT
510 PRINT X$;" = ";Y$
520 GOTO 400
```

69 Car Payments

Shopping for a new car? Use your computer to compute quickly the potential monthly car payment on various models.

Imagine you want an $8000 car and are prepared to put up $1000 against the purchase. You want to arrange to finance the car for 36 months. You know the current annual interest rate on car loans is 15 percent.

Key in those few numbers and the computer instantly tells you the car payment will be $242.66 per month.

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
20 GOSUB 300
30 PRINT"AUTOMOBILE PAYMENT"
40 GOSUB 300
50 PRINT
60 PRINT"PURCHASE PRICE $"
70 INPUT T
80 PRINT"DOWN PAYMENT $"
90 INPUT R
100 PRINT"NUMBER OF MONTHS"
110 INPUT N
120 PRINT"ANNUAL INTEREST %"
130 INPUT I
140 I=(0.01*I)/12
150 P=(T-R)*I/(1-1/(1+I)↑N)
160 PP=INT(100*P+0.5)/100
170 PRINT
180 PRINT"PAYMENT $",PP
190 PRINT
200 GOSUB 300
210 PRINT
220 INPUT"FOR MORE PRESS RETURN";K$
230 CLR
240 GOTO 10
300 FOR L=1 TO 22
310 PRINT CHR$(115);
```

```
320 NEXT L
330 RETURN
```

Sample Run

```
**********************
AUTOMOBILE PAYMENT
**********************

PURCHASE PRICE $
 8000
DOWN PAYMENT $
 1000
NUMBER OF MONTHS
 36
ANNUAL INTEREST %
 15

PAYMENT $              242.66

**********************

FOR MORE PRESS RETURN
```

70 To Nearest 95 Cents

Many companies like to price their goods at a figure ending in 95 cents. For instance, a ten dollar item might be marked $9.95 or $10.95.

Here's a program which demonstrates how to make all prices come out to the nearest 95 cents. See line 40. It merely takes the integer portion of the dollars number and adds 0.95 to it.

Program Listing

```
10 PRINT CHR$(147):REM SCREEN CLEAR
20 PRINT"MANUFACTURING COST $"
```

```
25 INPUT C
30 PRINT"PRICING MULTIPLIER"
35 INPUT M
40 P=INT(C*M)+0.95
50 PRINT:PRINT
60 PRINT"RETAIL PRICE $";P
70 PRINT:PRINT:PRINT
80 CLR:GOTO 20
```

Sample Run

```
MANUFACTURING COST $
 1.23
PRICING MULTIPLIER
 5


RETAIL PRICE $ 6.95




MANUFACTURING COST $
 5.67
PRICING MULTIPLIER
 3


RETAIL PRICE $ 17.95




MANUFACTURING COST $
 .46
PRICING MULTIPLIER
 10


RETAIL PRICE $ 4.95
```

71 To the Nearest Penny

This program is useful when you have a dollar-and-cents figure with more than two decimal places. For example, $151.6972. You need to transform $151.6972 to the more common $151.70

This small program would make a good subroutine in a larger set of instructions. To do so, insert GOSUB at the appropriate place in the larger set of program lines. Modify the line numbers of this small program so the subroutine will be located in an unused position in the larger listing. Change the last line of this small program to RETURN. Delete the first line.

Program Listing

```
  10 PRINT CHR$(147):REM CLEAR SCREEN
  20 PRINT"TYPE IN A"
  30 PRINT"DOLLAR FIGURE"
  40 PRINT"TO MORE THAN"
  50 PRINT"TWO DECIMAL PLACES"
  55 INPUT N
  60 R=INT(100*N+0.5)/100
  70 PRINT
  80 N$="$"+RIGHT$((STR$(N)),(LEN(STR$
     (N))-1))
  90 R$="$"+RIGHT$((STR$(R)),(LEN(STR$
     (R))-1))
100 PRINT CHR$(147):REM CLEAR SCREN
105 PRINT N$
110 PRINT"TO NEAREST PENNY IS"
115 PRINT R$
120 PRINT:PRINT:PRINT:PRINT
130 PRINT"PRESS ANY KEY"
140 PRINT"TO DO ANOTHER"
150 GET K$
160 IF K$="" THEN 150
170 CLR
180 GOTO 10
```

72 Mark Up

Mr. Storekeeper, here's just what you have needed to compute mark ups. This program causes your computer to find the retail price for which your percentage off would give the wholesale cost.

For instance, if you got 40 percent off on an item and paid $60, how much was it priced at, at retail? The answer is $100. To put that another way, if retail price or suggested retail price is $100 and you got 40 percent off at wholesale, what is the wholesale price? The answer is $60.

Try $40 wholesale which is 60 percent off. The answer is $100 retail. Or try $10 wholesale at 90 percent off. Retail would be $100. Or $75 wholesale at 25 percent off gives $100 retail.

Here's a toughie! Try $19.95 wholesale cost. Mark-up percentage is 40. The correct retail answer is $33.25.

Program Listing

```
10 PRINT CHR$(147):REM SCREEN CLEAR
20 PRINT"WHOLESALE COST $"
30 INPUT W
40 PRINT"MARK-UP PERCENTAGE %"
50 INPUT P
60 D=1-0.01*P
70 R=W/D
75 PRINT
80 PRINT"RETAIL PRICE $"
85 PRINT R
90 PRINT:PRINT:PRINT
100 CLR:GOTO 20
```

73 Percentage Off

From earlier tips in this book, you know how to make

your computer convert percentages to decimals. But what if you want to know "percentage off?"

For example, how much is 40 percent off? This program can be used to interpret 40 percent off and compute the decimal value needed. Try 40 percent off $100. The computer will change 40 percent off into decimal value 0.60. If you multiply 0.60 times $100 you find $60 is 40 percent off $100.

Line 50 makes the important translation.

Program Listing

```
10 PRINT CHR$(147):REM SCREEN CLEAR
20 INPUT"LIST PRICE";L
30 INPUT"PERCENT OFF";P
40 PRINT CHR$(147):REM CLEAR SCREEN
50 D=1-0.01*P
60 PRINT"TO COMPUTE WITH"
70 PRINT P;"% OFF"
80 PRINT"THE DECIMAL WILL BE"
90 PRINT D
100 PRINT
110 PRINT P;"% OFF";L
120 PRINT"RESULTS IN A COST OF"
130 PRINT "$";D*L
140 PRINT:PRINT:PRINT
150 CLR:GOTO 20
```

74 Dollars & Cents

If the result of your computation is a "money" answer, and you don't know whether to display it in dollars or cents, let the computer decide.

This program decides whether to display the output in dollars or cents. Line 50 in the program makes the decision.

Program Listing

```
10 PRINT CHR$(147)
20 INPUT"QUANTITY";P
30 INPUT"TOTAL COST $ ";C
40 T=C/P
50 IF T<1 THEN T=100*T:GOTO 80
60 PRINT"EACH COST $";T
70 PRINT:PRINT:CLR:GOTO 20
80 PRINT"EACH COST";T;"CENTS"
90 PRINT:PRINT:CLR:GOTO 20
```

75 Wages & Hours

These useful lines compute total hours worked at regular pay and number of hours worked at time-and-a-half overtime. The computer then finds gross pay and rounds off to the nearest cent.

The program knows that overtime starts after 40 hours. It makes payroll bookkeeping quick and simple.

Program Listing

```
10 PRINT CHR$(147)
20 PRINT"HOURLY PAY RATE"
25 INPUT"$ ";P
30 PRINT"NUMBER HOURS WORKED"
35 INPUT H
40 IF H>40 THEN OT=H-40:GOTO 100
50 W=H*P
60 PRINT"GROSS WAGES"
70 PRINT"$";W
90 END
100 W=(40*P)+(OT*P*1.5)
110 GOTO 60
```

76 Invoicing

There's a lot of repetitious math work to be done before you mail invoices to your customers. This software has the computer collect a few pertinent bits of data from you and then present all the various totals you need to plug into an invoice.

It gives you a total retail price for all goods sold on the invoice, total sales tax if applicable, shipping charges and the grand total amount due you from your customer.

Program Listing

```
10 PRINT CHR$(147)
20 INPUT"QUANTITY SOLD";Q
30 INPUT"UNIT PRICE $ ";P
40 INPUT"SALES TAX PERCENT";S
50 INPUT"SHIPPING CHARGES $ ";H
60 S=0.01*S:C=Q*P:T=C*S:F=C+T+H
70 CC=INT(100*C+0.5)/100
80 TT=INT(100*T+0.5)/100
90 FF=INT(100*F+0.5)/100
100 PRINT CHR$(147):REM CLEAR SCREEN
110 PRINT"PRICE $",CC
120 PRINT"TAX $",TT
130 PRINT"SHIPPING $",H
140 PRINT
150 PRINT"TOTAL"
160 PRINT"INVOICE $",FF
200 GET K$
210 IF K$="" THEN 200
220 CLR:GOTO 10
```

Sample Run

```
QUANTITY SOLD 100
UNIT PRICE $  2
SALES TAX PERCENT 5
SHIPPING CHARGES $  6

PRICE $              200
```

```
TAX $                10
SHIPPING $            6

TOTAL
INVOICE $           216
```

77 Unit Price

Suppose you find 895 green Widgets and buy them for $695. How much did each green Widget cost? Rounded off, 78 cents.

Unit price is total price divided by quantity. The quantity can be expressed in weight, total numbers, etc. It works the same whether you are talking about pounds of coffee, yards of concrete, gallons of ice cream, boxes of books, or units of Widgets.

This program asks for the name of the item, quantity purchased and total price paid. It then displays quantity, name, total and unit price.

Program Listing

```
10 PRINT CHR$(147)
20 PRINT"ITEM NAME:"
25 INPUT N$
30 PRINT
35 PRINT"QUANTITY OF ITEMS:"
40 INPUT Q
45 PRINT
50 PRINT"TOTAL PRICE"
55 PRINT"PAID FOR ITEMS:"
60 INPUT"$ ";P
70 U=P/Q
80 UU=INT(100*U+0.5)/100
90 PRINT
100 PRINT N$;" UNIT PRICE"
110 PRINT"IS $";UU
120 PRINT:PRINT:PRINT
```

```
130 INPUT"FOR MORE PRESS RETURN";K$
140 CLR:GOTO 10
```

Sample Run

```
ITEM NAME:
WIDGET

QUANTITY OF ITEMS:
 999

TOTAL PRICE
PAID FOR ITEMS:
$  14653

WIDGET UNIT PRICE
IS $ 14.67
```

78 Inventory Counter

The computer makes it very convenient to tally the number of items in your inventory.

This workaholic program is set up for up to ten different inventory items. You code them with the numbers 1 through 10.

You may enter any quantity for any item code in any mixed sequence. You may repeat item codes and add to quantities as often as you like.

When you finish entering quantities, enter a zero in response to the "item code" query. The computer will respond with a display of grand totals for each of the ten item codes.

Here's the routine:

Line 25 asks for the item code number from one through 10. If you enter a zero, action jumps to line 70 for a display of grand totals. Otherwise, the computer proceeds to line 40.

You are limited to item-code responses from zero through ten. If you try to enter a number larger than 10, the

computer will discover that via the test in line 40 and ship the whole operation back to line 20 where it asks you again for a valid number. If, at line 40, it finds a valid item code number from one to ten it will allow the execution to go on to line 50.

At line 50, the computer asks for the quantity of the item. Line 60 causes the computer to jump to the appropriate line to add that quantity to the various item totals.

Each of the lines 201 through 210 end with a jump to line 20. This allows you to continue to enter item codes and quantities as long as you like. When you finish, enter zero for item code and line 30 will push action to line 70.

At line 70, the computer finds instructions, through line 90, to display the grand totals.

Lines 100 to 130 ask if you want to do more.

To make the program run again, press the RETURN key on the computer's keyboard.

Program Listing

```
20 PRINT CHR$(147):REM CLEAR SCREEN
25 INPUT"ITEM CODE";C
30 IF C=0 THEN 70
40 IF C>10 THEN 20
50 INPUT"QUANTITY";Q
60 IF C=1 THEN 201
61 IF C=2 THEN 202
62 IF C=3 THEN 203
63 IF C=4 THEN 204
64 IF C=5 THEN 205
65 IF C=6 THEN 206
66 IF C=7 THEN 207
67 IF C=8 THEN 208
68 IF C=9 THEN 209
69 IF C=10 THEN 210
70 PRINT CHR$(147):REM CLEAR SCREEN
90 GOSUB 250
100 PRINT:PRINT:PRINT
110 INPUT"FOR MORE PRESS RETURN";K$
120 CLR
130 GOTO 20
201 J=J+Q:GOTO 20
202 K=K+Q:GOTO 20
```

```
203 L=L+Q:GOTO 20
204 M=M+Q:GOTO 20
205 N=N+Q:GOTO 20
206 R=R+Q:GOTO 20
207 S=S+Q:GOTO 20
208 T=T+Q:GOTO 20
209 U=U+Q:GOTO 20
210 V=V+Q:GOTO 20
220 END
250 PRINT"ITEM"
260 PRINT"CODE","QUANTITY"
270 PRINT
300 PRINT "1",J
310 PRINT "2",K
320 PRINT "3",L
330 PRINT "4",M
340 PRINT "5",N
350 PRINT "6",R
360 PRINT "7",S
370 PRINT "8",T
380 PRINT "9",U
390 PRINT "10",V
400 RETURN
```

Sample Run

```
ITEM CODE 5
QUANTITY 12

ITEM CODE 8
QUANTITY 2

ITEM CODE 3
QUANTITY 67

ITEM CODE 4
QUANTITY 1

ITEM CODE 9
QUANTITY 9

ITEM CODE 7
```

```
QUANTITY 21

ITEM CODE 0

ITEM
CODE              QUANTITY

 1                0
 2                0
 3                67
 4                1
 5                12
 6                0
 7                21
 8                2
 9                9
10                0


FOR MORE PRESS RETURN
```

79 Daily Code

Need to have a secret code each day of the year? This software generates a list of code numbers. Of course, you can change the list every day if you wish.

Program Listing

```
10 PRINT CHR$(147):Z=RND(-TI)
20 GOSUB 200
100 PRINT"SUNDAY",C:GOSUB 200
110 PRINT"MONDAY",C:GOSUB 200
120 PRINT"TUESDAY",C:GOSUB 200
130 PRINT"WEDNESDAY",C:GOSUB 200
140 PRINT"THURSDAY",C:GOSUB 200
150 PRINT"FRIDAY",C:GOSUB 200
```

```
160 PRINT"SATURDAY",C:END
200 C=INT(10000*RND(1))
210 IF C<1000 THEN 200
220 RETURN
```

Sample Run

```
SUNDAY                  9722
MONDAY                  1496
TUESDAY                 7648
WEDNESDAY               1930
THURSDAY                5119
FRIDAY                  3417
SATURDAY                2745
```

80 Daily Receipts Adder

This program allows a businessman to quickly add up his day's receipts, from both wholesale and retail orders as desired.

The machine first collects wholesale dollars from you.You key in "nothing" by pressing the RETURN key, without any data, to exit the wholesale entry loop.

Next the machine will go to retail and ask for those dollar figures from you.

Then it prints a summary of results including total wholesale dollars, total number of wholesale entries, total retail dollars, total number of retail entries, and then the grand total of all dollars and grand total of all entries.

Program Listing

```
10 PRINT CHR$(147)
20 PRINT"DAILY RECEIPTS ADDER"
30 PRINT"◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆"
40 INPUT"WHOLESALE $ ";WH$
50 IF WH$="" THEN 100
60 WT=WT+1
```

```
70 WD=WD+VAL(WH$)
80 WH$=""
90 GOTO 40
100 INPUT"RETAIL $ ";RL$
110 IF RL$="" THEN 160
120 RT=RT+1
130 RD=RD+VAL(RL$)
140 RL$=""
150 GOTO 100
160 PRINT CHR$(147):REM CLEAR SCREEN
180 PRINT"WHOLESALE $";WD
190 PRINT WT;"ENTRIES"
200 PRINT
210 PRINT"RETAIL $";RD
220 PRINT RT;"ENTRIES"
230 PRINT
240 PRINT"TOTAL $";WD+RD
250 PRINT WT+RT;"ENTRIES"
260 PRINT:PRINT:PRINT
270 INPUT"FOR MORE PRESS RETURN";K$
280 CLR:GOTO 10
```

Sample Run

```
DAILY RECEIPTS ADDER
**********************
WHOLESALE $ 23
WHOLESALE $ 31
WHOLESALE $ 56
WHOLESALE $ 47
WHOLESALE $ 93
WHOLESALE $ 11
WHOLESALE $ 67
WHOLESALE $ 88
WHOLESALE $ 70
WHOLESALE $
RETAIL $ 102
RETAIL $ 211
RETAIL $ 854
RETAIL $ 235
RETAIL $
```

```
WHOLESALE $ 486
 9 ENTRIES

RETAIL $ 1402
 4 ENTRIES

TOTAL $ 1888
 13 ENTRIES

FOR MORE PRESS RETURN
```

81 Advertising Cost-per-Thousand

Suppose your local radio station time salesman told you he could deliver 51,000 listeners for each $133 ad run on his station. And your local newspaper space salesman said he could deliver 160,000 readers for each $330 ad run in his paper. Which would be the better quantity buy for you?

This program gives you the answers in black and white. The newspaper would cost you about $2.06 for each 1000 readers while the radio station would cost almost $2.61 per thousand listeners. Now all you need to decide is which audience you prefer.

By the way, the cost-per-thousand comparison applies to magazines, TV, or any medium.

Program Listing

```
10 PRINT CHR$(147)
20 INPUT"AD COST $ ";A
30 INPUT"CIRCULATION";C
40 M=1000*(A/C)
45 PRINT
50 PRINT"$";M
55 PRINT"COST PER THOUSAND"
60 FOR Q=1 TO 10:PRINT:NEXT Q
70 GOTO 20
```

Sample Run

```
AD COST $  133
CIRCULATION 51000

$ 2.60784314
COST PER THOUSAND


AD COST $  330
CIRCULATION 160000

$ 2.0625
COST PER THOUSAND


AD COST $  5000
CIRCULATION 500000

$ 10
COST PER THOUSAND
```

82 Advertising
Cost-per-Unit Sold

Your favorite newspaper had the lowest cost-per-thousand so you ran an ad. The ad cost you $330. Lots of customers came by to check out your merchandise and you actually sold 77 pieces. What'd it cost you to sell each item?

With this quickie program you'll know it cost you $4.28 in ad money to sell each unit.

Program Listing

```
10 PRINT CHR$(147)
20 INPUT"AD COST $ ";A
30 INPUT"UNITS SOLD";U
```

```
40 C=A/U
50 PRINT
60 PRINT"THE AD COST"
70 PRINT"$";C
80 PRINT"PER UNIT SOLD"
90 FOR P=1 TO 11:PRINT:NEXT P
100 CLR
110 GOTO 20
```

Sample Run

```
AD COST $  330
UNITS SOLD 77

THE AD COST
$ 4.28571429
PER UNIT SOLD

AD COST $  5000
UNITS SOLD 500000

THE AD COST
$ .01
PER UNIT SOLD

AD COST $  1000
UNITS SOLD 279876

THE AD COST
$ 3.5730109E-03
PER UNIT SOLD
```

83 Making Change

A penny saved is a penny earned. Every businessman is aware of pennies, nickels, dimes, even quarters lost by sales people who can't make correct change. If you have sales people out front accepting cash

away from your register, or if you're too small to have a cash register, use this program to make *correct* change.

Key in the amount of the sale and the amount of money tendered by the customer and this software will tell you exactly how many quarters, dimes, nickels and pennies to hand back to the customer.

Program Listing

```
10 PRINT CHR$(147)
20 INPUT"SALE AMOUNT CENTS";X
30 INPUT"CENTS TENDERED";T
40 A=T-X
50 IF A<25 THEN 90
60 Q=Q+1
70 A=A-25
80 GOTO 50
90 IF A<10 THEN 130
100 D=D+1
110 A=A-10
120 GOTO 90
130 IF A<5 THEN 170
140 N=N+1
150 A=A-5
160 GOTO 130
170 P=A
180 PRINT
190 PRINT Q,"QUARTERS"
200 PRINT D,"DIMES"
210 PRINT N,"NICKLES"
220 PRINT P,"PENNIES"
230 FOR Z=1 TO 10:PRINT:NEXT Z
240 CLR
250 GOTO 20
```

Sample Run

```
SALE AMOUNT CENTS 33
CENTS TENDERED 100

 2              QUARTERS
 1              DIMES
```

```
 1              NICKLES
 2              PENNIES

SALE AMOUNT CENTS 1
CENTS TENDERED 100

 3              QUARTERS
 2              DIMES
 0              NICKLES
 4              PENNIES


SALE AMOUNT CENTS 27
CENTS TENDERED 50

 0              QUARTERS
 2              DIMES
 0              NICKLES
 3              PENNIES
```

84 Single-Digit Adder

Here's a bean counter for use in those jobs where you don't want to expend any mental energy on overseeing calculations. This program watches the keyboard for your single-digit number and immediately displays the total. You do *not* have to press RETURN after keying in a single-digit number.

Suppose you were counting beans in a jar. You could keep your finger near the number one key and repeatedly press it. The total of the number of times you pressed one would be displayed and immediately updated after each new press of one.

Of course, you could do the same for any of the keyboard numbers from zero through nine. Pressing any other key will cause your results to go bonkers.

Maybe you have sorted a pile of pennies into stacks of

five each. Press a five for each stack and you'll have the total number of pennies.

Program Listing

```
10 PRINT CHR$(147)
20 GOSUB 200
30 GET K$
40 IF K$="" THEN 30
50 IF K$="C" THEN T=0:GOTO 90
60 IF ASC(K$)<48 OR ASC(K$)>57 THEN 30
70 N=VAL(K$)
80 T=T+N
90 GOSUB 200
100 PRINT"TOTAL";T
110 GOTO 30
200 PRINT CHR$(147)
210 PRINT TAB(4);"ADDING MACHINE"
220 FOR L=1 TO 22:PRINT"*";:NEXT L
230 RETURN
```

85 Currency Converter

A customer in England sends you payment in Pounds. How many dollars is it worth?

This program converts foreign currency into U.S. dollars.

Call your bank for today's exchange rate. Suppose you have 100 English Pounds you wish to exchange for U.S. dollars. Your bank tells you the current rate is $2.50 per Pound. Your computer tells you that you have $40 in U.S. funds. Similarly, if you had $40, the computer would have computed the exchange to 100 Pounds.

Program Listing

```
10 PRINT CHR$(147)
20 PRINT"TYPE OF CURRENCY"
30 INPUT C$
```

```
40 PRINT
50 PRINT"CONVERT FROM DOLLARS"
60 PRINT"OR TO DOLLARS?"
70 PRINT:PRINT"PRESS F(FROM) OR T(TO)"
80 GET K$
90 IF K$="" THEN 80
100 IF K$="F" OR K$="T" THEN 200
110 GOTO 80
200 PRINT CHR$(147):REM SCREEN CLEAR
210 INPUT"EXCHANGE RATE";R
215 IF K$="T" THEN 300
220 INPUT"DOLLARS";D
230 X=D*R
240 PRINT
250 PRINT D;"DOLLARS"
260 PRINT "=";X;C$;"S"
270 PRINT:PRINT:PRINT:PRINT
280 INPUT"FOR MORE PRESS RETURN";Q$
290 CLR:K$="":GOTO 10
300 PRINT
310 PRINT"HOW MANY"
320 PRINT C$;"S";
330 INPUT D
340 X=D/R
350 PRINT
360 PRINT D;C$;"S"
370 PRINT"=";X;"DOLLARS"
380 GOTO 270
```

Sample Run

```
TYPE OF CURRENCY
POUND

CONVERT FROM DOLLARS
OR TO DOLLARS?

PRESS F(FROM) OR T(TO)
F

EXCHANGE RATE 2.5
DOLLARS 40
```

```
 40 DOLLARS
= 100 POUNDS

FOR MORE PRESS RETURN

TYPE OF CURRENCY
POUND

CONVERT FROM DOLLARS
OR TO DOLLARS?

PRESS F(FROM) OR T(TO)
T

EXCHANGE RATE 2.5

HOW MANY
POUNDS 100

 100 POUNDS
= 40 DOLLARS

FOR MORE PRESS RETURN
```

86 Fractional feet

You are measuring a box and the computation comes out to 14.5 feet. How do you change 14.5 feet into 14 feet 6 inches? Here's how:

Program Listing

```
10 PRINT CHR$(147)
20 PRINT"TYPE FRACTIONAL FEET"
30 PRINT"(FEET TO A DECIMAL)"
40 INPUT F
45 W=INT(F)
50 B=F-W
60 D=12*B
```

```
62 IF (D-INT(D))<0.5 THEN I=INT(D):GOTO 80
70 I=INT(D)+1
80 PRINT
90 PRINT F;"FEET ="
100 PRINT W;"FEET";I;"INCHES"
200 FOR L=1 TO 10:PRINT:NEXT L
210 INPUT"PRESS RETURN FOR MORE";K$
220 CLR:GOTO 10
```

Sample Run

```
TYPE FRACTIONAL FEET
(FEET TO A DECIMAL)

 14.5 FEET =
 14 FEET 6 INCHES

PRESS RETURN FOR MORE
```

87 Simulated RPN Calculator

The computer can be made to act very much like a calculator with Reverse Polish Notation (RPN).

To start a run , key in a number followed immediately by a math operator symbol. Your choices are limited to REM , -, / and *. Don't use other keys except BREAK. BREAK ends the run.

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
20 N$="":X$=""
25 PRINT T$
30 GET N$
35 IF N$="" THEN 30
40 IF ASC(N$)<42 OR ASC(N$)>57 THEN 30
50 IF ASC(N$)=46 OR ASC(N$)>47
   THEN X$=X$+N$:GOTO 30
60 IF ASC(N$)=42 THEN 1100
```

```
70 IF ASC(N$)=43 THEN 1200
80 IF ASC(N$)=45 THEN 1300
90 IF ASC(N$)=47 THEN 1400
95 GOTO 10
1100 X=VAL(X$)
1110 T=T*X
1120 GOTO 1420
1200 X=VAL(X$)
1210 T=T+X
1220 GOTO 1420
1300 X=VAL(X$)
1310 T=T-X
1320 GOTO 1420
1400 X=VAL(X$)
1410 T=T/X
1420 T$=STR$(T)
1430 GOTO 10
```

88 60-Second Timer

A one-minute timer can be very handy for fun-n-games. This easy-to-use clock counts off seconds up to 60.

The number of seconds counted can be changed by changing the number 60 in lines 20 and 80.

The clock can be calibrated by changing the number 725 in line 40. Line 40 is a time-delay loop set for approximately one second.

Program Listing

```
10 PRINT CHR$(147)
20 FOR T=1 TO 60
30 PRINT TAB(10);T
40 FOR TT=1 TO 725:NEXT TT
50 PRINT CHR$(147)
60 NEXT T
70 PRINT CHR$(147)
```

```
80 PRINT"60 SECONDS ELAPSED"
90 FOR A=1 TO 5
100 POKE 36878,15
110 POKE 36874,255
120 FOR Q=1 TO 250:NEXT Q
130 POKE 36878,0
140 FOR Q=1 TO 250:NEXT Q
150 NEXT A
160 FOR G=1 TO 10:PRINT:NEXT G
170 INPUT"PRESS RETURN FOR MORE";K$
180 CLR:GOTO 10
```

89 Stopwatch

Now you can leave that chrome-plated stopwatch at home next time you travel to your favorite auto or horse race. This program turns your computer into a handy stopwatch timer using the TV display.

When you RUN the program, the stopwatch will start counting seconds.

You can adjust the accuracy of the seconds count by changing the number 725 in line 120. We show it set at 725. To slow down the timer, increase that number. To speed up the clock, decrease the number.

Program Listing

```
10 GOSUB 400
20 PRINT"NUMBER OF SECONDS"
30 PRINT"TO BE COUNTED DOWN"
40 INPUT S
50 GOSUB 400
60 PRINT"PRESS ANY KEY"
70 PRINT"TO START TIMING"
80 GET K$
90 IF K$="" THEN 80
100 GOSUB 400
110 FOR T=S TO 1 STEP -1
120 FOR W=1 TO 725:NEXT W
130 PRINT T
```

```
140 NEXT T
200 GOSUB 400
210 PRINT"TIME IS UP"
220 PRINT S;"SECONDS"
230 PRINT"HAVE ELAPSED"
240 FOR B=1 TO 5
250 POKE 36878,15
260 POKE 36874,240
270 FOR W=1 TO 250:NEXT W
280 POKE 36878,0
290 FOR W=1 TO 250:NEXT W
300 NEXT B
310 FOR L=1 TO 10:PRINT:NEXT L
320 INPUT"PRESS RETURN FOR MORE";K$
330 CLR:GOTO 10
400 PRINT CHR$(147)
410 PRINT TAB(6);"STOPWATCH"
420 FOR L=1 TO 22:PRINT CHR$(166);:NEXT L
430 PRINT
440 RETURN
```

90 Shipments

A bar graph displaying number of shipments per month can be a useful way to see the trend in the movement of inventory from your warehouse.

This program collects data from you for one calendar year, January through December, in lines 100 to 210. That data is printed on the computer's video display by lines 410 to 750.

There are 17 graph squares left to right.

By the way, so the graph won't overflow the screen, the graph has 17 available positions from left to right.

Program Listing

```
10 PRINT CHR$(147)
100 INPUT"JANUARY";JA
110 INPUT"FEBRUARY";FE
120 INPUT"MARCH";MA
```

```
130 INPUT"APRIL";AP
140 INPUT"MAY";MY
150 INPUT"JUNE";JU
160 INPUT"JULY";JL
170 INPUT"AUGUST";AU
180 INPUT"SEPTEMBER";SE
190 INPUT"OCTOBER";OC
200 INPUT"NOVEMBER";NO
210 INPUT"DECEMBER";DE
410 PRINT CHR$(147):REM CLEAR SCREEN
500 PRINT TAB(2);"MONTHLY SHIPMENTS"
510 PRINT TAB(3);"(RANGE:0 TO 17)"
520 PRINT"JAN";
530 Z=JA:GOSUB 900
540 PRINT"FEB";
550 Z=FE:GOSUB 900
560 PRINT"MAR";
570 Z=MA:GOSUB 900
580 PRINT"APR";
590 Z=AP:GOSUB 900
600 PRINT"MAY";
610 Z=MY:GOSUB 900
620 PRINT"JUN";
630 Z=JU:GOSUB 900
640 PRINT"JUL";
650 Z=JL:GOSUB 900
660 PRINT"AUG";
670 Z=AU:GOSUB 900
680 PRINT"SEP";
690 Z=SE:GOSUB 900
700 PRINT"OCT";
710 Z=OC:GOSUB 900
720 PRINT"NOV";
730 Z=NO:GOSUB 900
740 PRINT"DEC";
750 Z=DE:GOSUB 900
800 FOR L=1 TO 5:PRINT:NEXT L
810 INPUT"FOR MORE PRESS RETURN";K$
820 CLR:GOTO 10
900 FOR L=1 TO Z
910 PRINT CHR$(162);
920 NEXT L
```

```
930 PRINT
940 RETURN
```

Sample Run

```
JANUARY 17
FEBRUARY 10
MARCH 6
APRIL 1
MAY 3
JUNE 9
JULY 12
AUGUST 15
SEPTEMBER 1
OCTOBER 17
NOVEMBER 13
DECEMBER 6

  MONTHLY SHIPMENTS
   (RANGE:0 TO 17)
JAN
FEB
MAR
APR
MAY
JUN
JUL
AUG
SEP
OCT
NOV
DEC

FOR MORE PRESS RETURN
```

91 Marketing Performance

Ever stared at a page of numbers for hours without really seeing their relationships? Well, that doesn't have

to happen when you put those numbers on a visual display—a graph.

This program is set up to take in data on each quarter of three years and then display that data on a color graph on your computer's video display.

The graph has 17 available positions from left to right so the graph won't overflow the screen. You could make it dollars or millions or whatever you need.

Anyway you cut it, it's a short, easy-to-type-in program which can give you a quick look at how marketing performance in one quarter-year relates to marketing performance in other quarter years.

Program Listing

```
10 PRINT CHR$(147)
100 INPUT"QUARTER 1 ";A
110 INPUT"QUARTER 2 ";B
120 INPUT"QUARTER 3 ";C
130 INPUT"QUARTER 4 ";D
140 INPUT"QUARTER 5 ";E
150 INPUT"QUARTER 6 ";F
160 INPUT"QUARTER 7 ";G
170 INPUT"QUARTER 8 ";H
180 INPUT"QUARTER 9 ";I
190 INPUT"QUARTER 10";J
200 INPUT"QUARTER 11";K
210 INPUT"QUARTER 12";L
400 PRINT CHR$(147):REM CLEAR SCREEN
410 PRINT CHR$(156)
450 PRINT "MARKETING PERFORMANCE"
460 PRINT TAB(3);"(RANGE:0 TO 17)"
500 PRINT CHR$(28)
510 PRINT
520 PRINT"Q 1";
530 Z=A:GOSUB 900
540 PRINT"Q 2";
550 Z=B:GOSUB 900
560 PRINT"Q 3";
570 Z=C:GOSUB 900
580 PRINT"Q 4";
590 Z=D:GOSUB 900
```

```
600 PRINT"Q 5";
610 Z=E:GOSUB 900
620 PRINT"Q 6";
630 Z=F:GOSUB 900
640 PRINT"Q 7";
650 Z=G:GOSUB 900
660 PRINT"Q 8";
670 Z=H:GOSUB 900
680 PRINT"Q 9";
690 Z=I:GOSUB 900
700 PRINT"Q10";
710 Z=J:GOSUB 900
720 PRINT"Q11";
730 Z=K:GOSUB 900
740 PRINT"Q12";
750 Z=L:GOSUB 900
800 FOR W=1 TO 3:PRINT:NEXT W
810 PRINT CHR$(31)
820 INPUT"FOR MORE PRESS RETURN";Y$
830 CLR:GOTO 10
900 IF Z>17 THEN PRINT" MORE THAN 17"
    :RETURN
905 FOR T=1 TO Z
910 IF Z>0 THEN PRINT CHR$(162);
920 NEXT T
930 PRINT
940 RETURN
```

Sample Run

```
QUARTER 1   17
QUARTER 2   1
QUARTER 3   16
QUARTER 4   2
QUARTER 5   15
QUARTER 6   3
QUARTER 7   14
QUARTER 8   4
QUARTER 9   13
QUARTER 10  5
QUARTER 11  12
QUARTER 12  6
```

```
MARKETING PERFORMANCE
    (RANGE:0 TO 17)


Q 1
Q 2
Q 3
Q 4
Q 5
Q 6
Q 7
Q 8
Q 9
Q10
Q11
Q12

FOR MORE PRESS RETURN
```

92 Cash Receipts Comparer

Business been up? Or, dropping off? How have your cash receipts looked over the last six months?

This short, easy-to-key-in piece of software accepts data from you about each month's receipts and then displays that data in an easy-to-read graph. The graph shows exactly how one month's receipts compare with another.

Lines 100 to 160 ask for the month's total receipts.

Lines 300 to 360 create the graph.

By the way, so the graph won't overflow the screen, the graph has 17 available positions from left to right.

Program Listing

```
10 PRINT CHR$(147)
20 DIM CR(6)
100 FOR L=1 TO 6
110 PRINT"CASH RECEIPTS"
120 PRINT"FOR MONTH";L
```

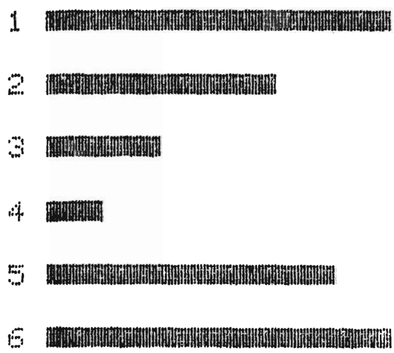
```
130 INPUT CR(L)
140 IF CR(L)>18 THEN 130
150 PRINT
160 NEXT L
200 PRINT CHR$(147):REM CLEAR SCREEN
210 PRINT CHR$(144)
220 PRINT TAB(4);"CASH RECEIPTS"
230 PRINT TAB(3);"(RANGE:0 TO 18)"
240 PRINT CHR$(28)
250 PRINT
300 FOR T=1 TO 6
305 PRINT T;
310 IF CR(T)>18 THEN PRINT" MORE THAN 18"
    :NEXT T
320 FOR X=1 TO CR(T)
330 IF CR(T)>0 THEN PRINT CHR$(18)
    ;CHR$(160);
340 NEXT X
350 PRINT CHR$(146):PRINT
360 NEXT T
400 FOR W=1 TO 3:PRINT:NEXT W
410 PRINT CHR$(31)
420 INPUT"FOR MORE PRESS RETURN";Y$
430 CLR:GOTO 10
```

Sample Run

```
CASH RECEIPTS
FOR MONTH 1
 18

CASH RECEIPTS
FOR MONTH 2
 12

CASH RECEIPTS
FOR MONTH 3
 6

CASH RECEIPTS
FOR MONTH 4
 3
```

```
CASH RECEIPTS
FOR MONTH 5
 15

CASH RECEIPTS
FOR MONTH 6
 18
```

```
    CASH RECEIPTS
   (RANGE:0 TO 18)
```




```
FOR MORE PRESS RETURN
```

Colorful Graphics

93 Box

In this *Colorful Graphics* section of the book, you will find a number of interesting new and different applications for the graphics capabilities of the computer. These can be modified, combined or otherwise changed to suit your own needs. Our titles represent only the thoughts we had when we watched these programs run. You might like to dream up new and different titles for your own creations made by modifying these programs.

Colors can be changed. Screen locations can be changed. Movement can be reversed. Try all of these programs. You'll like them!

Here's how to draw a box on the graphics display-screen area on your TV monitor.

Program Listing

```
10 PRINT CHR$(147)
20 PRINT:PRINT:PRINT
30 PRINT SPC(5);CHR$(111);
40 FOR L=1 TO 10:PRINT CHR$(163);:NEXT L
50 PRINT CHR$(112)
60 FOR L=1 TO 13
70 PRINT SPC(5);CHR$(165);SPC(10);CHR$(167)
80 NEXT L
90 PRINT SPC(5);CHR$(108);
100 FOR L=1 TO 10:PRINT CHR$(164);:NEXT L
110 PRINT CHR$(186)
```

94 Graphic-Screen Familiarization Tour

To make your color graphics as good as you can, you must know where you are drawing on the video display. And you need to know what you can draw. This program makes graphics screen locations familiar.

Lines 20 and 30 use text screen to ask you to select a particular graphics-screen location for use. The possible range of numbers here is 7680 to 8185.

Lines 40 to 60 set up the graphics screen. Then, the loop in lines 70 to 90 runs through and repeats all possible color-graphics characters available for you to command.

At line 80, the character numbers are POKEd into the screen location you selected back at line 30.

Try running this program several times, using different screen locations from 7680 to 8185 each time. You'll soon get a feel for the large number of locations available on the graphics screen. The more locations, the higher the resolution of the graphics.

Press BREAK to end.

Program Listing

```
10 PRINT CHR$(147)
20 PRINT"SELECT SCREEN LOCATION"
30 INPUT"7680 TO 8185";SL
40 PRINT CHR$(147)
60 POKE SL+30720,0
70 FOR C=0 TO 255
80 POKE SL,C
90 NEXT C
100 GOTO 70
```

95 Graphic-Screen Character Spotter

Just the opposite of the previous program, this set of program lines will let you select the graphic-screen character you want to see.

Lines 20 and 30 ask for your choice of a character number between zero and 255.

Lines 70-90 place your selected character in all locations on the face of the video display.

Press any key to repeat the run.

Program Listing

```
10 PRINT CHR$(147)
20 PRINT"SELECT CHARACTER"
30 INPUT"NUMBER 0 TO 255";C
40 PRINT CHR$(147)
70 FOR SL=7680 TO 8185
75 POKE 30720+SL,0
80 POKE SL,C
90 NEXT SL
100 GET K$
110 IF K$="" THEN 100
120 CLR:GOTO 10
```

96 Dancing Spot

You can make any one spot on the face of your television set, or video-display tube, dance or glitter with color using this program.

We generate random character numbers in line 20. Then we plug those numbers into our POKE statement in line 30.

Change the location of the spot on the screen by changing the number 7889 in lines 30 and 40 to any number between 7680 and 8185.

Program Listing

```
10 PRINT CHR$(147)
20 C=INT(8*RND(1))
30 POKE 7889+30720,C
40 POKE 7889,81
50 GOTO 20
```

97 Beautiful Braided Rug

Just goes to show that some of the most attractive

computer graphics require some of the shortest programs.

Program Listing

```
10 PRINT CHR$(147)
20 SL=INT(8186*RND(1))
30 IF SL<7680 THEN 20
40 POKE SL+30720,(INT(8*RND(1)))
50 POKE SL,102
60 GOTO 20
```

98 Screen Filler

This program fills the screen, side to side, top to bottom, with the character specified by the number at the end of line 30.

In this case, 128 is the number for a "reversed" white-on-black @ character. The program puts a reversed @ at every one of the 506 text-screen video locations.

Program Listing

```
10 PRINT CHR$(147)
20 FOR L=7680 TO 8185
25 POKE L+30720,0
30 POKE L,128
40 NEXT L
50 GOTO 50
```

Some say it looks like Outer Space. Maybe a view of Earth from out there? Whatever, it makes a fun, colorful display.

A random number from one to 255 is generated by line 30. It is used in the POKE statement in line 40 to create the myriad characters on the text-screen video display.

For a variation, change line 60 to:

```
60 GOTO 20
```

which will provide a continuously-changing screen pattern.

Program Listing

```
10 PRINT CHR$(147):REM CLEAR SCREEN
20 FOR V=7680 TO 8185
30 R=INT(256*RND(1))
35 POKE 30720+V,(INT(8*RND(1)))
40 POKE V,R
50 NEXT V
60 GOTO 60
```

99 Snowfall

White flakes sprinkle down the screen, over and over—until you press the * key. It may be useless but it's a lot of fun to watch!

Program Listing

```
10 POKE 36879,110
20 SL=INT(8186*RND(1))
30 IF SL<7680 THEN 20
40 POKE SL,42
50 GET K$
60 IF K$="*" THEN 80
70 GOTO 20
80 PRINT CHR$(147)
90 POKE 36879,27
100 PRINT"PRESS Q TO QUIT"
110 PRINT"PRESS R TO RUN AGAIN"
120 GET KY$
130 IF KY$="" THEN 120
140 IF KY$="R" THEN 10
150 IF KY$="Q" THEN 170
160 GOTO 120
170 END
```

100 Making Things Move

Movement on the computer display screen is an illusion. As in any television picture, the turning on and turning off of dots in a pattern across a screen can seem to provide motion to an object drawn on the face of the tube.

There are a number of ways to get the look of motion. Let's send a dot across the screen:

Program Listing

```
10 PRINT CHR$(147)
20 POKE 36879,46
30 FOR SL=7900 TO 7921
40 POKE SL+30720,1
50 POKE SL,81
60 FOR T=1 TO 50:NEXT T
70 POKE SL+30720,2
80 POKE SL,81
90 NEXT SL
100 GOSUB 200
110 FOR SL=7921 TO 7900 STEP -1
120 POKE SL+30720,1
130 POKE SL,81
140 FOR T=1 TO 50:NEXT T
150 POKE SL+30720,2
160 POKE SL,81
170 NEXT SL
180 GOSUB 200
190 GOTO 30
200 POKE 36878,15
210 POKE 36874,240
220 FOR T=1 TO 10:NEXT T
230 POKE 36878,0
240 POKE 36874,0
250 RETURN
```

101 Drawing Sketches

Now you can draw lines, rules, diagrams, maps, charts,

boxes—anything you can imagine--on the face of your color TV set. Use the Computer keyboard as your pen and its video output as your ink.

Lines 40 to340accept your up, down, right, or left commands, as U, D, R, or L. No other letters will work. Lines 400 and 410 draw your lines.

Program Listing

```
10 PRINT CHR$(147)
20 POKE 36879,254
30 X=7680
40 GET K$
50 IF K$="" THEN 40
60 IF K$="U" THEN 80
70 GOTO 100
80 IF X-22>7679 THEN X=X-22
90 GOTO 400
100 IF K$="D" THEN 120
110 GOTO 200
120 IF X+22<8186 THEN X=X+22
130 GOTO 400
200 IF K$="L" THEN 220
210 GOTO 300
220 IF ((X-2)/22)=INT((X-2)/22) THEN 40
230 IF X-1<7680 THEN 40
240 X=X-1
250 GOTO 400
300 IF K$="R" THEN 320
310 GOTO 40
320 IF ((X-1)/22)=INT((X-1)/22) THEN 40
330 IF X+1>8185 THEN 40
340 X=X+1
400 POKE X+30720,2
410 POKE X,81
420 GOTO 40
```

Commodore Computer Book II

101 Programming Tips & Tricks for the VIC-20 and Commodore 64

by Howard Adler

Here's a giant collection of practical, useful, efficient programming techniques and operating shortcuts for the *Commodore* VIC-20 Commodore Model 64, and other personal computers using the Commodore BASIC programming language, right out of a master programmer's notebook.

Loaded with hints, secrets and easy-to-follow instructions, this book shows you how to handle routine programming chores on the Commodore computers more quickly, do special effects, make your computer work for you—faster and more efficiently.

Each of the 101 computer programming tips in this book features a complete, tested, ready to type and run program. Each will run immediately, as you find it in this book, or it easily can be included in a larger set of instructions to your computer. All 101 programs in this book have been thoroughly tested on the VIC-20 and are ready to type in and run.

These are ideal programs for beginners, newcomers, novices and first-timers in the world of computers, as well as those old-hands needing new and different ideas for ways to use their *Commodore* computers.

Learn insider's how-to secrets for using the exciting BASIC words CLR, DATA, DIM, FOR, TO, STEP, END, GOSUB, GOTO, IF, THEN, INPUT, NEXT, PRINT, READ, REM, RESTORE, RETURN, CHR$, INT, LEFT$, MID$, RIGHT$, RND, SQR, STR$, TAB, TIME, VAL, AND, OR, and many more.

Sections in this book include *Introduction, Fun & Games, Text On Text, Gee Whiz!, Number Crunching, Money Matters, Colorful Graphics,* and a handy *Appendix*.

ARCsoft Publishers
Woodsboro, Maryland

ISBN 0-86668-030-6