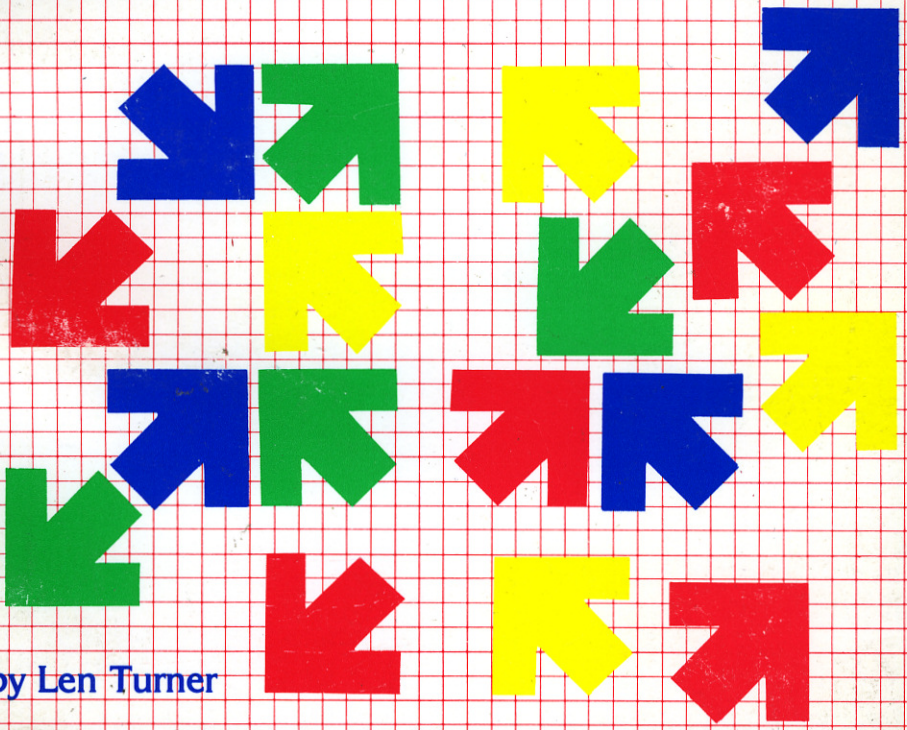
$8.95

# 101 Programming Tips & Tricks for the Texas Instruments TI-99/4A Home Computer

The exciting software ideabook, overflowing with pro hints, secrets, shortcuts and techniques for using the TI-99/2, Compact 40, TI-99/4A and other Texas Instruments personal, home and business microcomputers...with 101 complete programs.



by Len Turner

# 101 Programming Tips & Tricks for the Texas Instruments TI-99/4A Home Computer

**Texas Instruments books by Len Turner**

101 Programming Tips & Tricks for the Texas Instruments TI-99/4A Home Computer

Texas Instruments Computer Program Writing Workbook

36 Texas Instruments TI-99/4A Programs for Home, School & Office

# 101 Programming Tips & Tricks for the Texas Instruments TI-99/4A Home Computer

by Len Turner

ARCsoft Publishers

WOODSBORO, MARYLAND

FIRST EDITION
FIRST PRINTING



Library of Congress (LC) number: 83-5988



ISBN 0-86668-025-X

# Preface

The Texas Instruments microcomputers are among the world's most popular systems for use in the home, classroom and small-business office. In fact, the TI-99/4A probably is the all-time best-selling home computer to date.

The lightweight desktop design of the TI-99/4A, the convenient portability of the Compact 40, the powerful BASIC language capability of all the TI microcomputers place them in the forefront of the new wave of personal computers for hobbyists, students, teachers, professionals, business persons and all who want to learn the new technology.

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

The total number of applications to which the Texas

Instruments home, personal and business microcomputers can be put is limited only by the scope of the imagination. In this book, we have created 101 practical new sets of applications programs for your use. It is hoped that you will, by using these 101 programs, learn how to make your TI computer work for you. You will be able to gain an understanding of how programs work in the computer and how to build on these 101 easy-to-use pieces of software to make your computer do even more work as your understanding grows.

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-in-and-run programs for the Texas Instruments TI-99/4A, TI-99/2, Compact 40, and other TI personal, home and business microcomputers. You type these programs into your computer and it does the rest. You do not have to be a program writer to use this book!

This volume is a companion book to *36 Texas Instruments TI-99/4A Programs for Home, School & Office* and the *Texas Instruments Computer Program Writing Workbook*.

—Len Turner

# Table of Contents

## Text on Text



## Gee Whiz



## Number Crunching

**Money Matters**

## Colorful Graphics



## Appendix

# Introduction

There is a great need for practical, useful software for the new generation of popular personal computers. The Texas Instruments models TI-99/4A Home Computer, TI-99/2 computer, Compact 40 portable computer, and other TI personal/home/business microcomputers, for instance, are among the world's most popular gear. The TI computers are powerful and 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 101 complete easy-to-type-in ready-to-run new and different sets of program listings for you to use in getting ideas about how to make your own TI work for you.

These programs are very 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. You can learn a great deal about how BASIC programs are organized and how they work simply by typing in these programs. Use these fun and practical

programs and, then, modify them and expand them to suit your needs as your interests grow.

These programs are designed to be typed into your computer, via its keyboard console, just as you find the programs printed here in this book. No other programming is needed. We assume you have read the owner's manuals and instructional pamphlets which came with your computer and accessories. You know how to hook up the console to the TV modulator/connector and to any other accessories you may have purchased. You know how to type the programs into your TI computer. 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!

These programs do not require tape or disk, unless you choose to save them on those media. These programs are so easy to type in you can save this book and retype them whenever you wish to rerun a program.

## Computer printouts

To make sure there are no errors in these programs, we have written and tested each and every program on our own TI-99/4A *and* printed every one on a TI-99 line printer. The hardcopy printout from that line printer is reproduced directly in this book!

The TI 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 these programs run exactly as reproduced here.

If, after typing in a program from this book, you get an error message from your TI computer, check the handy list of TI BASIC words and other TI info in the Appendix at the back of this book. Then compare your typed program lines with the program lines reproduced in this book.

Undoubtedly, you will find you have made a typing error in entering the program lines into your TI. However, should you find an error in a program 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.

## Home, school and office

This book has been organized into six sections plus an Appendix. The first section includes programs which might be useful in creating new and different games to play on your TI. The second section helps you learn how the computer uses words, letters, numbers, symbols and alphanumerics. The third section is a collection of programs you can use to wow your friends. The fourth section includes tips and tricks on how to work with, and manipulate, numbers in the computer. The fifth section holds practical-use programs for around the home, especially in money matters. The sixth sections shows the way to video graphics on the TV screen. The Appendix holds lots of useful reference information, including a list of BASIC words as used in the TI.

The sections are: *Fun & Games, Text on Text, Gee Whiz, Number Crunching, Money Matters, Colorful Graphics, and Appendix.*

The sections include a total of 101 complete ready-to-run programs in the BASIC language. Each is made brief so you can quickly type it in and get it running fast. Each stands alone and runs all by itself. Or each could be incorporated into a longer set of program lines as a subroutine.

Try them all. They're great fun to run. And they are especially designed to be short so you won't have to spend hours typing in one program.

## Endless running

Many of the programs in this book will continue to run until you command them off manually via the CLEAR function. You may stop any run, at any time, by use of the CLEAR function.

The function key is in the lower right corner of the console keyboard and is labeled FCTN. Press and hold FCTN and press the number 4 key in the upper row of keys. The combination of FCTN and 4 creates the CLEAR instruction to the computer.

This CLEAR function is the same as what is called BREAK in other microcomputers.

Here is an example of how the CLEAR function works in the TI computer. Type in this brief two-line pro-

gram. Type in line 10 and press ENTER. Then type in line 20 and press ENTER. This will lodge the complete program in program memory. Here is the program:

```
10 PRINT "XYZ"
20 GOTO 10
```

After you have the program stored in program memory, type in RUN and press ENTER to start the operation. The computer will do as instructed. It will print the letters XYZ repeatedly. In fact, it will go on forever until you stop the action.

To stop the run, press and hold the FCTN key. While holding FCTN down, press the number 4 key. This is the CLEAR function. It will stop the computer run. Try it.

## REMarks

As you read through all of the programs in this book, you will notice few REM, or remarks, statements. The author's training in writing BASIC-language computer programs included an emphasis on brevity and saving of 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 *Texas Instrument Program Writing Workbook* published by *ARCsoft Publishers*. Your objective always should be to make the most efficient use of available memory.

Always 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 various programs into your TI computer, one at a time, you'll make modifications to suit your personal needs and interests. For instance, exact wording of PRINT statements can be changed. Or two or more programs can be combined into one grand scheme. Your applications may vary.

If you want to load more than one of these programs into your TI computer at the same time, be sure to use different sets of line numbers for different programs.

Computer programmers today generally mix the use of the two words, ENTER and RETURN. They are used to

mean the same thing. In this case, we mean the ENTER key on the right side of the console keyboard.

## Other computers

These programs will run on any computer which is set up to be programmed in BASIC. However, to run these on machines other than ones using TI BASIC as found in the TI-99/4A, you may have to make slight modifications to program lines. Graphic commands, especially, will differ elsewhere. Also use of multiple-statement lines, using the colon (:), is quite different in most other forms of BASIC.

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

Also, if you use a non-TI microcomputer, such things as line numbering, spacing, logical tests, multiplication symbols, print statements and other instructions may differ.

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

## Standalone vs. subroutine

All of the programs in this book can be used as portions of larger lists of instructions to your computer. That is, 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 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 each 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.

If you want to load more than one of these programs

into your TI computer at the same time, be sure to use different sets of line numbers.

## Learning programming

These programs are written to be typed into your TI computer just as you find them here with no 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 much of the programming advice in this book will, in fact, also be of interest to old-timers in the program-writing game since we have presented many powerful new twists aimed at making your computer do more work more quickly.

Amidst the 101 tips, tricks & programs in this book, you will find countless secrets, shortcuts, hints, techniques and make-it-easier instructions for using your computer. Each tip, trick and program is intended to make you a more-versatile programmer and make your programming chores lighter.

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 writer, modify these programs and 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 ENTER 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 ENTER.

## Program Listing

```
10 CALL CLEAR
15 RANDOMIZE
20 IF INT(3*RND)<1 THEN 100
30 PRINT "TAILS"
40 PRINT
45 PRINT
50 INPUT "FOR MORE, PRESS ENTER":KY$
60 CALL CLEAR
70 GOTO 20
100 PRINT "HEADS"
110 GOTO 40
```

# 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 RANDOMIZE
20 CALL CLEAR
100 A=INT(7*RND)
110 IF A<1 THEN 100
200 B=INT(7*RND)
210 IF B<1 THEN 200
300 PRINT "FIRST DICE:",A
310 PRINT "SECOND DICE:",B
400 FOR L=1 TO 10
410 PRINT
420 NEXT L
430 PRINT "TO ROLL DICE AGAIN,"
440 INPUT "PRESS ENTER ":KY$
450 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 ENTER on your computer 's keyboard to roll the dice.

## Program Listing

```
10 RANDOMIZE
15 CALL CLEAR
```

```
20 PRINT "TO ROLL TWO DICE,"
30 INPUT "PRESS ENTER":KY$
40 PRINT
50 GOSUB 100
60 PRINT "*************"
70 PRINT "* ";DL;" * ";DR;" *"
80 PRINT "*************"
90 PRINT
95 GOTO 20
100 DL=INT(7*RND)
110 IF DL<1 THEN 100
120 DR=INT(7*RND)
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 RANDOMIZE
15 CALL CLEAR
20 PRINT "TO ROLL TWO DICE,"
30 INPUT "PRESS ENTER":KY$
40 CALL CLEAR
50 FOR L=1 TO 2
60 DQ=INT(7*RND)
70 IF DQ<1 THEN 60
80 DR=INT(7*RND)
90 IF DR<1 THEN 80
```

```
100 PRINT "**************"
110 PRINT "* ";DQ;" * ";DR;" *"
120 PRINT "**************"
130 PRINT
140 NEXT L
150 PRINT
160 PRINT
170 GOTO 20
```

# 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 RANDOMIZE
20 CALL CLEAR
100 DATA A,B,C,D,E,F,G,H,I,J,K,L,M
110 DATA N,O,P,Q,R,S,T,U,V,W,X,Y,Z
200 FOR N=1 TO 13
210 C=INT(1000*RND)
220 READ L$
230 D=INT(1000*RND)
240 READ J$
250 PRINT L$;" ";C,J$;" ";D
260 NEXT N
```

## 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 |

## Sample Run

| | | | |
|---|---|---|---|
| A | 307 | B | 328 |
| C | 445 | D | 503 |
| E | 278 | F | 21 |
| G | 460 | H | 888 |
| I | 220 | J | 130 |
| K | 169 | L | 869 |
| M | 920 | N | 346 |
| O | 381 | P | 158 |
| Q | 45 | R | 366 |
| S | 800 | T | 430 |
| U | 424 | V | 516 |
| W | 199 | X | 780 |
| Y | 224 | Z | 964 |

# 6 Sound Off

You can make your computer beep on command.

Line 10 clears the screen. The FOR/NEXT loop in lines 20 to 50 make the TI 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. Line 40 makes the beeper sound off.

### Program Listing

```
10 CALL CLEAR
20 FOR L=1 TO 10
30 PRINT "BEEP"
40 CALL SOUND(1,1000,1)
50 NEXT L
```

# 7 R2D2 Sound Effect

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

### Program Listing

```
10 CALL CLEAR
20 PRINT "****************"
30 PRINT "* ROBOT SOUND *"
40 PRINT "****************"
50 PRINT
60 PRINT
70 PRINT
80 PRINT
90 RANDOMIZE
100 D=INT(10*RND)
110 IF D<1 THEN 100
120 F=INT(5000*RND)
130 IF F<110 THEN 120
140 V=INT(20*RND)
200 CALL SOUND(D,F,V)
210 GOTO 100
```

# 8 Sound Off II

Here's another, slightly different version of our beeper program.

## Program Listing

```
10 CALL CLEAR
20 D=5
30 V=5
40 FOR F=110 TO 1100 STEP 10
50 CALL SOUND(D,F,V)
60 NEXT F
```

# 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 CALL CLEAR
20 X$="BUTLER"
30 INPUT "WHODUNIT? ":A$
40 IF X$=A$ THEN 100
50 PRINT
60 PRINT "CLUE: ","SERVANT"
70 PRINT
80 GOTO 30
100 PRINT "YES, YOU GUESSED IT"
110 PRINT "THE ";X$;" DID IT"
```

# 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 200 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 RANDOMIZE
20 CALL CLEAR
30 T=0
100 R=INT(1001*RND)
200 INPUT "GUESS THE NUMBER ":B
210 PRINT
220 T=T+1
230 PRINT "THAT WAS TRY NUMBER ";T
```

```
300 IF B>R THEN 350
310 IF B<R THEN 330
320 GOTO 400
330 PRINT "TOO LOW"
340 GOTO 360
350 PRINT "TOO HIGH"
360 INPUT "GUESS AGAIN ":B
370 GOTO 210
400 CALL SOUND(99,440,1)
410 PRINT "YES, YOU GOT IT !"
420 PRINT R;" IS THE NUMBER"
430 PRINT "YOU GOT IT IN ";T;" TRIES"
440 PRINT
450 PRINT
460 PRINT
470 GOTO 30
```

# 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 CALL CLEAR
20 FOR L=1 TO 5
30 N=INT(91*RND)
40 IF N<65 THEN 30
50 PRINT CHR$(N);
60 NEXT L
70 PRINT
80 GOTO 20
```

### Sample Run

```
CYGQH     MWHOJ     KAFDH
XMIAJ     CTWRQ     BTPNC
BZPDO     VFEZK     QDFGN
REMSF     HALVN     ANBWO
NTVEA     NXECR     RUNSX
IDPLG     PODAA     TPKII
```

# 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 200 in line 50. Line 50 is a time-delay loop set for approximately one second.

Lines 70-90 provide a rapid burst of five beeps when the clock reaches 60 seconds. To change the length of this alarm, change the number 5 in line 70.

### Program Listing

```
10 CALL CLEAR
20 FOR T=1 TO 60
30 CALL SOUND(1,5000,1)
40 PRINT T,"SECONDS"
50 FOR L=1 TO 200
55 NEXT L
60 NEXT T
70 FOR E=1 TO 5
80 CALL SOUND(99,3000,1)
90 NEXT E
```

# 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 20-210 take in the info on each person. As each person's score is entered, lines 60 - 200 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 ENTER without data. That will prompt the computer, at lines 230 and 240, to print the lowest score and the highest score.

## Program Listing

```
10 CALL CLEAR
20 INPUT "NAME: ":N$
30 IF N$="" THEN 220
40 INPUT "SCORE: ":S
50 X=X+1
60 IF X=1 THEN 80
70 GOTO 130
80 LS=S
90 LN$=N$
100 HS=S
110 HN$=N$
120 GOTO 20
130 IF S<LS THEN 150
140 GOTO 170
150 LS=S
160 LN$=N$
170 IF S>HS THEN 190
180 GOTO 20
190 HS=S
200 HN$=N$
210 GOTO 20
220 CALL CLEAR
230 PRINT LN$;" LOWEST AT ";LS
240 PRINT HN$;" HIGHEST AT ";HS
```

```
250 FOR Z=1 TO 10
260 PRINT
270 NEXT Z
280 INPUT "TO DO MORE, PRESS ENTER":KY$
290 LS=0
300 HS=0
310 LN$=""
320 HN$=""
330 X=0
340 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 CALL CLEAR
20 INPUT "TYPE A SCORE: ":N
30 IF N=0 THEN 170
40 S=S+1
50 IF S=1 THEN 70
60 GOTO 100
70 LN=N
80 HN=N
90 GOTO 20
100 IF N<LN THEN 120
110 GOTO 130
120 LN=N
130 IF N>HN THEN 150
140 GOTO 20
```

```
150 HN=N
160 GOTO 20
170 CALL CLEAR
180 PRINT "LOW NUMBER IS ";LN
190 PRINT "HIGH NUMBER IS ";HN
200 FOR Z=1 TO 10
210 PRINT
220 NEXT Z
230 PRINT "FOR A DIFFERENT"
240 PRINT "SET OF NUMBERS,"
250 INPUT "PRESS ENTER":KY$
260 S=0
270 LN=0
280 HN=0
290 GOTO 10
```

## Sample Run

```
TYPE A SCORE:  321
TYPE A SCORE:  789
TYPE A SCORE:  654
TYPE A SCORE:  135
TYPE A SCORE:  864
TYPE A SCORE:  285
TYPE A SCORE:  432
TYPE A SCORE:  915
TYPE A SCORE:  0
LOW NUMBER IS  135
HIGH NUMBER IS  915

FOR A DIFFERENT
SET OF NUMBERS,
PRESS ENTER

TYPE A SCORE:  8
TYPE A SCORE:  2
TYPE A SCORE:  5
TYPE A SCORE:  9
TYPE A SCORE:  1
TYPE A SCORE:  3
```

```
TYPE A SCORE:  7
TYPE A SCORE:  4
TYPE A SCORE:  6
TYPE A SCORE:  0
LOW NUMBER IS  1
HIGH NUMBER IS  9

FOR A DIFFERENT
SET OF NUMBERS,
PRESS ENTER
```

# 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 170. If you guessed correctly in response to the query in line 160, at line 175 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 160 answer stored in P$ with the correct answer stored in A$.

If you blew it and guessed wrong, the program drops below line 170 to line 184 where it increases your "wrong score" by adding one point to W.

If you got a W+1 at line 184, the program moves back to line 100 and gets you to try again. If you scored a victory and got an R+1 at line 175, the program jumps to line 200 where it stops to display your total right and wrong score. After that, it's back to line 40 for a complete new run-through.

## Program Listing

```
10 RANDOMIZE
20 CALL CLEAR
```

```
40 S=INT(7*RND)
50 IF S<1 THEN 40
60 FOR L=1 TO S
70 READ A$
80 NEXT L
90 RESTORE
95 PRINT "**********************"
100 PRINT "WHO KILLED THE DUKE ?"
110 READ B$,C$,D$,E$,F$,G$
120 RESTORE
130 PRINT
135 PRINT "WAS IT THE..."
140 PRINT B$,C$,D$,E$,F$,G$
150 PRINT
160 INPUT "WHODUNIT ";P$
170 IF A$=P$ THEN 175
172 GOTO 180
175 R=R+1
176 PRINT
177 PRINT "OKAY, YOU ARE RIGHT"
178 PRINT "IT WAS THE ";A$
179 GOTO 200
180 PRINT
182 PRINT "NO, NOT THE ";P$
184 W=W+1
186 PRINT
188 GOTO 100
200 PRINT
210 PRINT "YOUR SCORE SO FAR IS..."
220 PRINT R;" RIGHT ",W;" WRONG"
230 PRINT "THAT IS ";100*(R/(R+W));"PERCENT"
240 PRINT
250 GOTO 40
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 and 800 as shown. Line 800 will give you the R and T values you'll need going into the program at line 900.

### Program Listing

```
10 CALL CLEAR
800 INPUT "NUMBER RIGHT: ":R
810 INPUT "NUMBER TRIES: ":T
820 PRINT
900 PRINT R;" RIGHT"
910 PRINT "IN ";T;" TRIES"
920 D=R/T
930 P=100*D
940 B=INT(10*P)
950 PRINT "THAT'S ";P;" PERCENT"
960 PRINT "YOU ARE BATTING ";B
1000 FOR Z=1 TO 10
1010 PRINT
1020 NEXT Z
1030 GOTO 800
```

# 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 CALL CLEAR
700 INPUT "NUMBER RIGHT: ":G
720 INPUT "NUMBER OF TRIES: ":E
730 PRINT
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 910
850 IF H<300 THEN 920
860 IF H<500 THEN 930
870 IF H<700 THEN 940
880 IF H<900 THEN 950
900 PRINT "HALL OF FAME MATERIAL"
905 GOTO 960
910 PRINT "THE PITS"
915 GOTO 960
920 PRINT "POOR"
925 GOTO 960
930 PRINT "AVERAGE"
935 GOTO 960
940 PRINT "TOP NOTCH"
945 GOTO 960
950 PRINT "DAMN NEAR PERFECT"
960 PRINT
970 PRINT "YOUR BATTING AVERAGE IS ";H
1000 FOR Z=1 TO 10
1010 PRINT
1020 NEXT Z
1030 GOTO 700
```

# 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 CALL CLEAR
20 INPUT "PLAYER'S NAME:  ":N$
30 INPUT "PLAYER'S SCORE: ":S
40 PRINT
50 S$=STR$(S)
60 LN=LEN(N$)+LEN(S$)
70 LT=LN+14
80 FOR Z=1 TO LT
90 PRINT "*";
100 NEXT Z
110 PRINT
120 PRINT "* ";N$;"'S SCORE: ";S$;" *"
130 FOR Z=1 TO LT
140 PRINT "*";
150 NEXT Z
160 FOR Z=1 TO 10
170 PRINT
180 NEXT Z
190 GOTO 20
```

## Sample Run

```
PLAYER'S NAME:  HELEN
PLAYER'S SCORE:  88

**********************
* HELEN'S SCORE: 88 *
**********************

PLAYER'S NAME:  SAM
PLAYER'S SCORE:  98765

************************
* SAM'S SCORE: 98765 *
************************
```

FGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijk
GHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijkl
HIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklm
IJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmn
JKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmno
KLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnop
LMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopq
MNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqr
NOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrs
OPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrst
PQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstu
QRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuv
RSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvw
STUVWXYZ[\]^_`abcdefghijklmnopqrstuvwx
TUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxy
UVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz
VWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{
WXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|
XYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}
YZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
Z[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
[\]^_`abcdefghijklmnopqrstuvwxyz{|}~ !
\]^_`abcdefghijklmnopqrstuvwxyz{|}~ !"
]^_`abcdefghijklmnopqrstuvwxyz{|}~ !"#
^_`abcdefghijklmnopqrstuvwxyz{|}~ !"#$
_`abcdefghijklmnopqrstuvwxyz{|}~ !"#$%
`abcdefghijklmnopqrstuvwxyz{|}~ !"#$%&
abcdefghijklmnopqrstuvwxyz{|}~ !"#$%&'
bcdefghijklmnopqrstuvwxyz{|}~ !"#$%&'(
cdefghijklmnopqrstuvwxyz{|}~ !"#$%&'()
defghijklmnopqrstuvwxyz{|}~ !"#$%&'()*
efghijklmnopqrstuvwxyz{|}~ !"#$%&'()*+
fghijklmnopqrstuvwxyz{|}~ !"#$%&'()*+,
ghijklmnopqrstuvwxyz{|}~ !"#$%&'()*+,-
hijklmnopqrstuvwxyz{|}~ !"#$%&'()*+,-.
ijklmnopqrstuvwxyz{|}~ !"#$%&'()*+,-./
jklmnopqrstuvwxyz{|}~ !"#$%&'()*+,-./0
klmnopqrstuvwxyz{|}~ !"#$%&'()*+,-./01
lmnopqrstuvwxyz{|}~ !"#$%&'()*+,-./012
mnopqrstuvwxyz{|}~ !"#$%&'()*+,-./0123
nopqrstuvwxyz{|}~ !"#$%&'()*+,-./01234
opqrstuvwxyz{|}~ !"#$%&'()*+,-./012345

# 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 CALL CLEAR
15 RANDOMIZE
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
80 PRINT "HOW MANY","U.S. PRESIDENTS",
   "CAN YOU NAME ?"
90 R=INT(12*RND)
100 IF R<1 THEN 90
110 IF INT(R/2)=(R/2)THEN 116
114 GOTO 120
```

```
116 R=R+1
120 FOR L=1 TO R
130 READ S$
140 NEXT L
150 PRINT
160 PRINT "WHO WAS THE"
170 PRINT S$
180 PRINT "PRESIDENT OF THE U.S."
190 READ C$
200 INPUT D$
210 PRINT
220 IF D$=C$ THEN 235
225 PRINT "THAT'S WRONG"
230 GOTO 240
235 PRINT "THAT'S CORRECT"
240 PRINT "THE ";S$;" PRESIDENT WAS"
250 PRINT C$
260 RESTORE
270 PRINT
280 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 CALL CLEAR
20 INPUT "GIVE ME A NUMBER  ":N
30 FOR L=1 TO N
40 X=X+L
50 FOR T=1 TO 100
55 NEXT T
60 CALL CLEAR
70 FOR T=1 TO 100
75 NEXT T
80 PRINT "I AM THINKING"
90 NEXT L
100 CALL CLEAR
110 PRINT "I HAVE THE ANSWER NOW"
115 PRINT
120 PRINT "THE TOTAL OF ALL NUMBERS"
130 PRINT "FROM 1 TO ";N;" IS ";X
140 PRINT
150 PRINT
160 PRINT
170 X=0
180 GOTO 20
```

# 21 Word-Error Trapping

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 50, the operation is returned to line 10 for a new try at the correct input.

### Program Listing

```
10 INPUT "WANT TO PLAY AGAIN?  ";A$
20 IF A$="YES" THEN 100
30 IF A$="NO" THEN 200
40 PRINT "PLEASE ANSWER ONLY YES OR NO"
50 PRINT
60 GOTO 10
100 PRINT "THANK YOU FOR THAT ";A$
110 PRINT
120 PRINT
130 GOTO 10
200 GOTO 100
```

# 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 CALL CLEAR
20 FOR N=32 TO 126
30 PRINT N,CHR$(N)
40 FOR T=1 TO 400
50 NEXT T
60 NEXT N
70 GOTO 20
```

# 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 CALL CLEAR
20 INPUT "WHAT IS THE PASSWORD?  ":A$
30 IF A$="ELEPHANT" THEN 100
40 PRINT A$;" ISN'T IT"
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 CALL CLEAR
20 INPUT "WHAT IS THE PASSWORD?  ":A$
30 IF A$="ELEPHANT" THEN 100
40 B=B+1
50 PRINT A$;" ISN'T IT"
60 IF B=3 THEN 80
70 GOTO 20
80 END
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 CALL CLEAR
20 INPUT "YOUR ACCOUNT NUMBER?  ":UA
```

```
30 GOSUB 300
40 FOR L=1 TO 3
50 INPUT "PASSWORD?  ":PS$
55 IF PS$="" THEN 50
60 IF PS$=PW$ THEN 100
70 NEXT L
80 PRINT "YOU WERE INCORRECT"
90 END
100 PRINT "YOU GOT IT ALL RIGHT"
110 PRINT "NOW THE PROGRAM WOULD RUN"
120 END
300 IF UA=12345 THEN 306
304 GOTO 310
306 PW$="ZEBRA"
310 IF UA=23456 THEN 316
314 GOTO 320
316 PW$="GOOSE"
320 IF UA=34567 THEN 326
324 GOTO 330
326 PW$="TROUT"
330 IF UA=45678 THEN 336
334 GOTO 340
336 PW$="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 CALL CLEAR
20 INPUT "WHAT IS YOUR NAME?  ":N$
```

```
30 LN=LEN(N$)
40 LT=LN+4
50 CALL CLEAR
60 FOR L=1 TO LT
70 PRINT "*";
80 NEXT L
90 PRINT
100 PRINT "* ";N$;" *"
110 FOR L=1 TO LT
120 PRINT "*";
130 NEXT L
150 FOR L=1 TO 10
160 PRINT
170 NEXT L
180 GOTO 20
```

### Sample Run

```
WHAT IS YOUR NAME?  ED

******
* ED *
******

WHAT IS YOUR NAME?  NAME

********
* NAME *
********
```

# 27 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 CALL CLEAR
20 B=0
100 INPUT "GIVE ME A NUMBER:  ":A
110 IF A=0 THEN 200
120 B=B+A
130 PRINT
140 GOTO 100
200 PRINT
210 PRINT "THE TOTAL OF"
220 PRINT "THOSE NUMBERS IS ";B
300 PRINT
310 PRINT
320 PRINT
330 GOTO 20
```

# 28 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 CALL CLEAR
100 INPUT "GIVE ME A NUMBER:  ":A$
110 IF A$="X" THEN 200
120 B=VAL(A$)
```

```
130 C=C+B
140 GOTO 100
200 PRINT
210 PRINT "THE TOTAL OF"
220 PRINT "THOSE NUMBERS IS ";C
300 PRINT
310 PRINT
320 PRINT
330 C=0
340 GOTO 100
```

# 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 CALL CLEAR
20 INPUT "WORD/NUMBER:    ":N$
30 IF N$="" THEN 20
40 L=LEN(N$)
50 PRINT "REVERSED:       ";
60 FOR Y=L TO 1 STEP -1
70 B$=SEG$(N$,Y,1)
80 PRINT B$;
90 NEXT Y
100 PRINT
110 FOR K=1 TO 10
120 PRINT
130 NEXT K
140 PRINT "PRESS ANY KEY TO DO MORE"
150 CALL KEY(0,Z,X)
160 IF X=0 THEN 150
170 B$=""
180 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 CALL CLEAR
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 Texas Instruments 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 RANDOMIZE
```

```
100 CALL CLEAR
110 N=INT(9*RND)
120 IF N<1 THEN 110
130 FOR L=1 TO N
140 READ DM$
150 NEXT L
160 PRINT DM$
170 CALL KEY(0,Z,X)
180 IF X=0 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 CALL CLEAR
20 INPUT "A PLURAL NOUN:  ":PN$
30 INPUT "A VERB:  ":VB$
40 INPUT "A SINGULAR NOUN:  ":SN$
50 CALL CLEAR
60 PRINT "THE ";PN$;" ";VB$;" ";SN$;"."
70 FOR L=1 TO 10
80 PRINT
90 NEXT L
100 INPUT "TO DO ANOTHER, PRESS ENTER":KY$
110 GOTO 10
```

### Sample Run

```
A PLURAL NOUN   ?
DOGS
A VERB   ?
LOVE
A SINGULAR NOUN   ?
FOOD

THE DOGS LOVE FOOD.

A PLURAL NOUN   ?
BOXES
A VERB   ?
HOLD
A SINGULAR NOUN   ?
WATER

THE BOXES HOLD WATER
```

# 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 CALL CLEAR
20 PRINT "TYPE IN A GROUP OF SCORES"
30 PRINT "FROM 0 TO 100, ONE AT A TIME"
```

```
40 PRINT
50 PRINT "ENTER AN X AFTER LAST SCORE"
60 PRINT
70 INPUT "SCORE:  ":G$
80 IF G$="X" THEN 200
85 G=VAL(G$)
90 N=N+1
100 IF G<60 THEN 104
102 GOTO 110
104 F=F+1
106 GOTO 150
110 IF G<70 THEN 114
112 GOTO 120
114 D=D+1
116 GOTO 150
120 IF G<80 THEN 124
122 GOTO 130
124 C=C+1
126 GOTO 150
130 IF G<90 THEN 134
132 GOTO 140
134 B=B+1
136 GOTO 150
140 A=A+1
150 IF N=1 THEN 154
152 GOTO 160
154 L=G
156 H=G
158 GOTO 180
160 IF G<L THEN 164
162 GOTO 170
164 L=G
170 IF G>H THEN 174
172 GOTO 180
174 H=G
180 S=S+G
190 GOTO 70
200 P=S/N
210 M=L+((H-L)/2)
220 CALL CLEAR
230 PRINT "THERE WERE ";N;" SCORES"
240 PRINT "RANGING FROM ";L;" TO ";H
```

```
250 PRINT
260 PRINT "MID-RANGE:",M
270 PRINT "AVERAGE:",P
280 PRINT
290 PRINT "TOTALS"
300 PRINT "A:",A
310 PRINT "B:",B
320 PRINT "C:",C
330 PRINT "D:",D
340 PRINT "F:",F
400 PRINT
410 PRINT
420 PRINT
430 PRINT "FOR MORE, PRESS ANY KEY"
440 CALL KEY(0,Z,X)
450 IF X=0 THEN 440
460 A=0
470 B=0
480 C=0
490 D=0
500 F=0
510 H=0
520 L=0
530 N=0
540 S=0
550 GOTO 20
```

# 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 CALL CLEAR
20 RANDOMIZE
30 P=INT(91*RND)
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
12              L
6               F
5               E
1               A
2               B
5               E
```

# 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 CALL CLEAR
6 RANDOMIZE
10 PRINT TAB(2);"A";TAB(6);"B";TAB(11);"C"
15 PRINT
20 GOSUB 900
30 IF X>50 THEN 34
32 GOTO 40
34 A=A+1
36 GOTO 20
40 GOSUB 900
50 IF X>50 THEN 54
52 GOTO 60
54 B=B+1
56 GOTO 20
60 GOSUB 900
70 IF X>50 THEN 74
72 GOTO 100
74 C=C+1
76 GOTO 20
100 PRINT TAB(1);A;TAB(5);B;TAB(10);C
110 IF T=19 THEN 200
120 A=0
```

```
130 B=0
140 C=0
150 T=T+1
160 GOTO 20
200 GOTO 200
900 X=INT(100*RND)
910 RETURN
```

**Sample Run**

```
A    B    C
6    4    4
6    1    4
6    0    0
0    0    0
5    2    0
0    0    0
2    6    1
11   4    3
0    0    0
2    0    0
7    1    0
2    1    0
1    0    0
0    1    0
1    1    0
0    1    0
25   13   5
5    5    2
6    0    1
11   2    0
```

# 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 CALL CLEAR
15 RANDOMIZE
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)
150 IF R<1 THEN 140
160 IF INT(R/2)=(R/2)THEN 164
162 GOTO 170
164 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 260
240 PRINT "WRONG"
250 GOTO 270
260 PRINT "CORRECT"
270 PRINT "NUMBER OF DAYS IS ";C$
280 RESTORE
290 FOR Z=1 TO 10
300 PRINT
320 NEXT Z
330 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 CALL CLEAR
20 INPUT "GIVE ME A NUMBER:  ":N
30 IF N<1 THEN 20
40 FOR L=1 TO N
50 X=X+L
60 NEXT L
70 CALL SOUND(100,440,1)
80 PRINT "THE TOTAL OF ALL"
90 PRINT "NUMBERS 1 THROUGH ";N
100 PRINT "IS ";X
110 FOR Z=1 TO 11
120 PRINT
130 NEXT Z
140 X=0
150 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 CALL CLEAR
20 PRINT "SELECT A THREE-DIGIT NUMBER"
30 PRINT "ALL THREE DIGITS THE SAME"
40 PRINT
50 PRINT "ADD THE DIGITS TOGETHER"
60 PRINT
70 PRINT "WHAT IS THE SUM"
80 INPUT "OF THE THREE DIGITS? ":N
90 IF N<3 THEN 80
100 IF N>27 THEN 80
110 Q=37*N
120 PRINT
130 PRINT
140 CALL SOUND(100,1000,1)
150 PRINT "YOUR NUMBER IS ";Q
160 PRINT
170 PRINT
180 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 30 selects a *yes*

answer if the random number is greater than 49. Otherwise, line 40 chooses a *no* answer.

### Program Listing

```
10 CALL CLEAR
15 RANDOMIZE
20 X=100*RND
30 IF X>49 THEN 60
40 PRINT "NO"
50 END
60 PRINT "YES"
70 END
```

# 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 CALL CLEAR
20 PRINT "GIVE ME THE NUMBER OF"
30 PRINT "A LETTER FROM THE ALPHABET"
40 INPUT "FROM 1 TO 26  ":N
60 X=N+64
70 PRINT
75 PRINT
80 PRINT "LETTER NUMBER ";N;" IS ";CHR$(X)
90 PRINT
100 PRINT
110 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 CALL CLEAR
20 DATA A,B,C,D,E,F,G,H,I,J,K,L,M
30 DATA N,O,P,Q,R,S,T,U,V,W,X,Y,Z
40 PRINT "GIVE ME THE NUMBER OF"
50 PRINT "A LETTER FROM THE ALPHABET"
60 INPUT "FROM 1 TO 26 ";N
70 FOR L=1 TO N
80 READ A$
90 NEXT L
100 PRINT
110 PRINT
120 PRINT "LETTER NUMBER ";N;" IS ";A$
130 RESTORE
140 PRINT
150 PRINT
160 GOTO 40
```

# 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 CALL CLEAR
20 RANDOMIZE
30 Q$="*"
40 GOTO 190
50 N=INT(101*RND)
60 INPUT "GUESS MY SECRET NUMBER?  ":G
70 IF G>N THEN 90
80 GOTO 120
90 PRINT "TOO HIGH"
100 PRINT
110 GOTO 60
120 IF G<N THEN 140
130 GOTO 160
140 PRINT "TOO LOW"
150 GOTO 100
160 PRINT "RIGHT !"
170 PRINT
180 PRINT
190 FOR L=1 TO 23
200 PRINT Q$;
210 NEXT L
220 GOTO 50
```

## Sample Run

```
**********GUESS MY SECRET NUMBER?   37
TOO LOW
GUESS MY SECRET NUMBER?   67
TOO HIGH
GUESS MY SECRET NUMBER?   47
TOO LOW
GUESS MY SECRET NUMBER?   57
TOO HIGH
GUESS MY SECRET NUMBER?   53
TOO LOW
GUESS MY SECRET NUMBER?   55
TOO HIGH
GUESS MY SECRET NUMBER?   54
RIGHT !
```

# 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 RANDOMIZE
20 CALL CLEAR
30 Z=1
40 S=10*RND
50 N=INT(S*Z)
60 PRINT
70 PRINT "REMEMBER---->",N
80 IF W=3 THEN 100
90 GOTO 120
100 PRINT "FORGET IT"
110 GOTO 310
120 GOSUB 410
130 CALL CLEAR
140 INPUT "WHAT WAS IT?  ":S
150 IF S<>N THEN 170
160 GOTO 210
170 PRINT "***YOU ARE WRONG***"
180 W=W+1
190 GOSUB 410
200 GOTO 60
210 PRINT "+++YOU ARE RIGHT+++"
```

```
220 R=R+1
230 Z=Z*10
240 W=0
250 GOSUB 410
260 PRINT
270 PRINT R;" RIGHT SO FAR"
280 GOSUB 410
290 PRINT
300 GOTO 40
310 PRINT
320 PRINT "YOU HAD ";R;" RIGHT"
330 GOSUB 410
340 PRINT
350 PRINT "LET'S START OVER"
360 INPUT "PRESS ENTER  ":KY$
370 R=0
380 W=0
390 Z=0
400 GOTO 20
410 FOR T=1 TO 500
420 NEXT T
430 RETURN
```

# 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 CALL CLEAR
20 INPUT "ENTER A THREE-DIGIT NUMBER:  ":N
30 IF N<100 THEN 20
40 IF N>999 THEN 20
50 A=INT(N/100)
60 B=INT(10*((N/100)-A))
70 C=INT(10*((N/10)-(INT(N/10))))
```

```
80 A$=STR$(A)
90 B$=STR$(B)
100 C$=STR$(C)
110 CALL CLEAR
120 PRINT "THE OLD NUMBER WAS ";N
130 PRINT "THE NEW NUMBER IS   ";C$;B$;A$
140 PRINT
150 PRINT
160 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.

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 any key on the computer's keyboard.

## Program Listing

```
10 CALL CLEAR
20 PRINT
30 PRINT "ENTER LETTER GRADE:"
40 CALL KEY(0,Z,X)
50 IF X=0 THEN 40
```

```
60 G$=CHR$(Z)
70 IF G$="X" THEN 280
80 IF G$="A" THEN 100
90 GOTO 120
100 A=A+1
110 GOTO 20
120 IF G$="B" THEN 140
130 GOTO 160
140 B=B+1
150 GOTO 20
160 IF G$="C" THEN 180
170 GOTO 200
180 C=C+1
190 GOTO 20
200 IF G$="D" THEN 220
210 GOTO 240
220 D=D+1
230 GOTO 20
240 IF G$="F" THEN 260
250 GOTO 270
260 F=F+1
270 GOTO 20
280 CALL CLEAR
290 PRINT "A:  ";A
300 PRINT "B:  ";B
310 PRINT "C:  ";C
320 PRINT "D:  ";D
330 PRINT "F:  ";F
340 PRINT
350 PRINT "TO DO ANOTHER SET,"
360 PRINT "PRESS ANY KEY"
370 CALL KEY(0,Z,X)
380 IF X=0 THEN 370
390 A=0
400 B=0
410 C=0
420 D=0
430 F=0
440 X=0
450 Z=0
460 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

```
10 INPUT "GIVE ME A NUMBER  ";A
20 IF A<10 THEN 24
22 GOTO 30
24 PRINT "TOO LOW"
26 GOTO 10
30 IF A>100 THEN 34
32 GOTO 40
34 PRINT "TOO HIGH"
36 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 TOTAL:      22
MEAN:            4.4
VARIANCE:        2.24
STD DEVIATN:     1.496662955
```

## Program Listing

```
10 CALL CLEAR
20 GOSUB 260
30 PRINT
40 INPUT "DATA POINT:  ":X
50 IF X=999999999 THEN 100
60 T=T+X
70 S=S+X^2
80 N=N+1
90 GOTO 40
100 A=T/N
110 V=S/N-A^2
120 D=SQR(V)
130 CALL CLEAR
140 GOSUB 260
150 PRINT
160 PRINT "DATA TOTAL:",T
170 PRINT "MEAN:",A
180 PRINT "VARIANCE:",V
190 PRINT "STD DEVIATN:",D
200 PRINT
210 PRINT
220 N=0
230 S=0
240 T=0
250 GOTO 40
260 PRINT "STANDARD DEVIATION"
270 PRINT "******************"
280 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 CALL CLEAR
20 INPUT "PERCENTAGE:  ":P
30 D=0.01*P
40 PRINT "DECIMAL:    ";D
50 PRINT
60 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 CALL CLEAR
```

```
20 B=10
30 C=10*B
40 IF B THEN 100
50 END
100 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 CALL CLEAR
20 B=(6+8)>(3*4)
30 C=(5+2)>(9+2)
40 IF B THEN 100
50 IF C THEN 200
60 END
100 PRINT "B OKAY"
110 GOTO 50
200 PRINT "C OKAY"
210 END
```

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

## Program Listing

```
10 CALL CLEAR
20 INPUT "GIVE ME A NUMBER:  ":Z
30 IF Z=0 THEN 80
40 IF Z<1 THEN 20
45 IF Z>100 THEN 20
50 N=N+1
60 IF Z<51 THEN 64
62 GOTO 70
64 B=B+1
70 GOTO 20
80 A=N-B
90 PRINT
95 PRINT
100 PRINT "TOTAL NO.'S",N
110 PRINT "1 TO 50",B
120 PRINT "51 TO 100",A
130 PRINT
140 PRINT
150 PRINT
160 B=0
170 N=0
180 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 CALL CLEAR
20 DIM Q(1700)
30 FOR L=0 TO 999
40 Q(L)=0
50 NEXT L
60 INPUT "NUMBER:  ":N
70 FOR L=2 TO N/2
80 M=N/L
90 IF M=INT(M)THEN 110
100 GOTO 120
110 Q(L)=M
120 NEXT L
130 PRINT
140 PRINT "FACTORS ARE:"
150 FOR L=1 TO N/2
160 IF Q(L)THEN 180
170 GOTO 200
180 PRINT Q(L)
190 GOTO 210
200 Z=Z+1
210 NEXT L
220 IF N=1 THEN 240
230 GOTO 260
240 PRINT "NONE"
250 GOTO 290
260 IF Z=INT(N/2)THEN 280
```

```
270 GOTO 290
280 PRINT "NONE"
290 PRINT
300 PRINT
310 Z=0
320 GOTO 30
```

# 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 CALL CLEAR
20 INPUT "GIVE ME A NUMBER:  ":Z
30 IF Z=0 THEN 80
40 N=N+1
50 IF N=1 THEN 54
52 GOTO 60
54 D=Z
60 IF Z<D THEN 64
62 GOTO 70
64 D=Z
70 GOTO 20
80 PRINT
85 PRINT
90 PRINT "THE SMALLEST NUMBER IS ";D
100 PRINT
110 PRINT
120 PRINT
130 N=0
140 GOTO 20
```

# 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 CALL CLEAR
20 INPUT "GIVE ME A NUMBER:  ":Z
30 IF Z=0 THEN 60
40 IF Z>D THEN 44
42 GOTO 20
44 D=Z
50 GOTO 20
60 PRINT
65 PRINT
70 PRINT "THE LARGEST NUMBER IS ";D
100 PRINT
110 PRINT
120 PRINT
130 GOTO 20
```

# 54 Reciprocals

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

## Program Listing

```
10 CALL CLEAR
20 PRINT "NUMBER TO BE CONVERTED"
25 INPUT "TO ITS RECIPROCAL?  ":N
30 R=1/N
35 PRINT
40 PRINT "RECIPROCAL OF ";N;" IS ";R
45 PRINT
50 PRINT
55 GOTO 20
```

# 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 CALL CLEAR
20 PRINT "GIVE ME A NUMBER"
30 INPUT "WITH A DECIMAL:  ":N
40 X=N-INT(N)
50 PRINT
60 PRINT "THE FRACTIONAL PORTION"
70 PRINT "OF ";N;" IS ";X
80 FOR L=1 TO 10
90 PRINT
100 NEXT L
110 GOTO 20
```

# 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 CALL CLEAR
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
80 PRINT
90 PRINT
100 PRINT
110 PRINT "THE AVERAGE IS ";A
120 FOR Q=1 TO 10
130 PRINT
140 NEXT Q
150 N=0
160 T=0
170 GOTO 20
```

# 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 CALL CLEAR
20 INPUT "GIVE ME A NUMBER:  ":Z
30 IF Z=0 THEN 90
40 N=N+1
50 IF N=1 THEN 54
52 GOTO 60
54 H=Z
56 L=Z
60 IF Z<L THEN 64
62 GOTO 70
64 L=Z
70 IF Z>H THEN 74
72 GOTO 20
74 H=Z
80 GOTO 20
90 M=L+((H-L)/2)
100 PRINT
110 PRINT
120 PRINT "MID-RANGE:",M
130 FOR W=1 TO 10
140 PRINT
150 NEXT W
160 H=0
170 L=0
180 M=0
190 N=0
200 GOTO 20
```

# 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 CALL CLEAR
20 PRINT "GIVE ME A NUMBER"
30 INPUT "TO BE ROUNDED OFF? ":N
40 PRINT
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 N;" ROUNDS OFF TO ";R
140 FOR E=1 TO 10
150 PRINT
160 NEXT E
170 GOTO 20
```

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

## Program Listing

```
10 CALL CLEAR
20 PRINT "GIVE ME A NUMBER"
30 INPUT "TO BE ROUNDED OFF? ":N
40 PRINT
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 N;" ROUNDS OFF TO ";R
140 FOR E=1 TO 10
150 PRINT
160 NEXT E
170 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 CALL CLEAR
20 PRINT "GIVE ME A NUMBER"
30 PRINT "TO MORE THAN"
40 INPUT "TWO DECIMAL PLACES:  ":N
50 R=INT(100*N+0.5)/100
60 CALL CLEAR
70 PRINT N;" ROUNDS TO"
80 PRINT R
90 PRINT
100 PRINT "       OR"
110 PRINT
120 PRINT "$";N;" BECOMES"
130 PRINT "$";R
140 FOR Y=1 TO 10
150 PRINT
160 NEXT Y
170 GOTO 20
```

# 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 CALL CLEAR
20 INPUT "SALE PRICE $ ":P
30 INPUT "SALES TAX  % ":R
40 T=0.01*R
50 S=T*P
60 B=P+S
70 PRINT "SALES TAX  ";"$";S
80 PRINT "TOTAL BILL $";B
90 FOR K=1 TO 10
100 PRINT
110 NEXT K
120 GOTO 20
```

# 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 RANDOMIZE
```

```
15 CALL CLEAR
20 T=INT(1000*RND)
30 V=V+1
40 IF 0.1*V=INT(0.1*V)THEN 44
42 GOTO 50
44 PRINT V,T
50 GOTO 20
```

# 62 Random Sampler

This program strengthens your confidence in the random 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 RANDOMIZE
20 CALL CLEAR
30 FOR L=1 TO 100
40 X=INT(101*RND)
50 IF X<50 THEN 70
60 GOTO 80
70 Y=Y+1
80 IF X>49 THEN 100
90 GOTO 110
100 N=N+1
110 NEXT L
120 PRINT Y;" YES"
130 PRINT N;" NO"
140 FOR B=1 TO 11
150 PRINT
160 NEXT B
170 N=0
180 Y=0
190 GOTO 30
```

### Sample Run

```
50  YES
50  NO

53  YES
47  NO

46  YES
54  NO

50  YES
50  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 RANDOMIZE
15 CALL CLEAR
20 PRINT INT(10*RND)
30 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 RANDOMIZE
20 CALL CLEAR
30 FOR L=1 TO 100
40 N=INT(10*RND)
50 IF N=0 THEN 70
60 GOTO 80
70 A=A+1
80 IF N=1 THEN 100
90 GOTO 110
100 B=B+1
110 IF N=2 THEN 130
120 GOTO 140
130 C=C+1
140 IF N=3 THEN 160
150 GOTO 170
160 D=D+1
170 IF N=4 THEN 190
180 GOTO 200
190 E=E+1
200 IF N=5 THEN 220
210 GOTO 230
220 F=F+1
230 IF N=6 THEN 250
```

```
240 GOTO 260
250 G=G+1
260 IF N=7 THEN 280
270 GOTO 290
280 H=H+1
290 IF N=8 THEN 310
300 GOTO 320
310 I=I+1
320 IF N=9 THEN 340
330 GOTO 350
340 J=J+1
350 PRINT "COUNTING",L
360 NEXT L
370 CALL CLEAR
380 K=A+B+C+D+E+F+G+H+I+J
390 PRINT "0",A
400 PRINT "1",B
410 PRINT "2",C
420 PRINT "3",D
430 PRINT "4",E
440 PRINT "5",F
450 PRINT "6",G
460 PRINT "7",H
470 PRINT "8",I
480 PRINT "9",J
490 PRINT
500 PRINT "TOTAL",K
510 PRINT
520 PRINT
530 PRINT
540 PRINT "TO DO ANOTHER SET,"
550 PRINT "PRESS ANY KEY"
560 CALL KEY(0,Z,X)
570 IF X=0 THEN 560
580 A=0
590 B=0
600 C=0
610 D=0
620 E=0
630 F=0
640 G=0
650 H=0
```

```
660 I=0
670 J=0
680 K=0
690 GOTO 20
```

# 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 ENTER key on the computer's keyboard.

## Program Listing

```
10 RANDOMIZE
20 CALL CLEAR
30 FOR L=0 TO 99
40 N=INT(10*RND)
50 NT=NT+N
60 PRINT "AVERAGING",L
70 NEXT L
80 CALL CLEAR
90 AV=NT/100
100 PRINT "TOTAL OF 100"
110 PRINT "RANDOM NOS."
120 PRINT "ZERO TO NINE",NT
130 PRINT
140 PRINT "AVERAGE IS ",AV
150 FOR Q=1 TO 11
160 PRINT
170 NEXT Q
180 INPUT "FOR MORE, PRESS ENTER":KY$
190 NT=0
200 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 RANDOMIZE
15 CALL CLEAR
20 FOR L=0 TO 99
30 N=INT(1000*RND)
40 IF L=0 THEN 44
42 GOTO 50
44 LN=N
46 HN=N
50 IF N<LN THEN 54
52 GOTO 60
54 LN=N
60 IF N>HN THEN 64
62 GOTO 70
64 HN=N
70 PRINT "SORTING",L
80 NEXT L
90 CALL CLEAR
110 PRINT "LOW NUMBER:",LN
120 PRINT "HIGH NUMBER:",HN
130 FOR Z=1 TO 11
140 PRINT
150 NEXT Z
160 PRINT "TO DO ANOTHER, PRESS ANY KEY"
170 CALL KEY(0,Z,X)
180 IF X=0 THEN 170
190 HN=0
200 LN=0
210 GOTO 20
```

UVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz
VWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{
WXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|
XYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}
YZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
Z[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
[\]^_`abcdefghijklmnopqrstuvwxyz{|}~ !
\]^_`abcdefghijklmnopqrstuvwxyz{|}~ !"
]^_`abcdefghijklmnopqrstuvwxyz{|}~ !"#
^_`abcdefghijklmnopqrstuvwxyz{|}~ !"#$
_`abcdefghijklmnopqrstuvwxyz{|}~ !"#$%
`abcdefghijklmnopqrstuvwxyz{|}~ !"#$%&
abcdefghijklmnopqrstuvwxyz{|}~ !"#$%&'
bcdefghijklmnopqrstuvwxyz{|}~ !"#$%&'(
cdefghijklmnopqrstuvwxyz{|}~ !"#$%&'()
defghijklmnopqrstuvwxyz{|}~ !"#$%&'()*
efghijklmnopqrstuvwxyz{|}~ !"#$%&'()*+
fghijklmnopqrstuvwxyz{|}~ !"#$%&'()*+,
ghijklmnopqrstuvwxyz{|}~ !"#$%&'()*+,-
hijklmnopqrstuvwxyz{|}~ !"#$%&'()*+,-.
ijklmnopqrstuvwxyz{|}~ !"#$%&'()*+,-./
jklmnopqrstuvwxyz{|}~ !"#$%&'()*+,-./0
klmnopqrstuvwxyz{|}~ !"#$%&'()*+,-./01
lmnopqrstuvwxyz{|}~ !"#$%&'()*+,-./012
mnopqrstuvwxyz{|}~ !"#$%&'()*+,-./0123
nopqrstuvwxyz{|}~ !"#$%&'()*+,-./01234
opqrstuvwxyz{|}~ !"#$%&'()*+,-./012345
pqrstuvwxyz{|}~ !"#$%&'()*+,-./0123456
qrstuvwxyz{|}~ !"#$%&'()*+,-./01234567
rstuvwxyz{|}~ !"#$%&'()*+,-./012345678
stuvwxyz{|}~ !"#$%&'()*+,-./0123456789
tuvwxyz{|}~ !"#$%&'()*+,-./0123456789:
uvwxyz{|}~ !"#$%&'()*+,-./0123456789:;
vwxyz{|}~ !"#$%&'()*+,-./0123456789:;<
wxyz{|}~ !"#$%&'()*+,-./0123456789:;<=
xyz{|}~ !"#$%&'()*+,-./0123456789:;<=>
yz{|}~ !"#$%&'()*+,-./0123456789:;<=>?
z{|}~ !"#$%&'()*+,-./0123456789:;<=>?@
{|}~ !"#$%&'()*+,-./0123456789:;<=>?@A
|}~ !"#$%&'()*+,-./0123456789:;<=>?@AB
}~ !"#$%&'()*+,-./0123456789:;<=>?@ABC
~ !"#$%&'()*+,-./0123456789:;<=>?@ABCD

# 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!

## Program Listing

```
10 CALL CLEAR
20 INPUT "PRINCIPAL         $":P
30 INPUT "ANNUAL INTEREST %":R
40 INPUT "NUMBER MONTHS      ":M
50 PRINT
60 FOR Q=1 TO M
70 I=(P*(0.01*R))/12
80 P=P+I
90 PRINT Q;" MONTH = $";P
100 NEXT Q
110 PRINT
120 INPUT "TO DO ANOTHER, PRESS ENTER":KY$
130 P=0
140 GOTO 10
```

# 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 CALL CLEAR
20 PRINT "SHOPPER'S FRIEND"
30 PRINT "****************"
40 INPUT "FIRST BRAND: ":X$
50 INPUT "QUANTITY: ":M
60 INPUT "PRICE: ":N
70 NM=N/M
80 INPUT "SECOND BRAND: ":Y$
90 INPUT "QUANTITY: ":Q
100 INPUT "PRICE: ":R
110 RQ=R/Q
120 IF NM=RQ THEN 300
130 IF NM<RQ THEN 170
140 CALL CLEAR
150 PRINT Y$;" IS THE BEST BUY"
160 GOTO 190
170 CALL CLEAR
180 PRINT X$;" IS THE BEST BUY"
```

```
190 PRINT
200 NM=INT(100*NM+0.5)/100
210 RQ=INT(100*RQ+0.5)/100
220 PRINT X$;" UNIT","$";NM
230 PRINT Y$;" UNIT","$";RQ
240 PRINT
250 PRINT
260 PRINT "TO DO MORE, PRESS ANY KEY"
270 CALL KEY(0,Z,X)
280 IF X=0 THEN 270
290 GOTO 10
300 CALL CLEAR
310 PRINT X$;" = ";Y$
320 GOTO 190
```

# 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 CALL CLEAR
20 GOSUB 210
30 PRINT "AUTOMOBILE PAYMENT"
40 GOSUB 210
50 PRINT
60 PRINT
70 INPUT "PURCHASE PRICE    $":T
80 INPUT "DOWN PAYMENT      $":R
90 INPUT "NUMBER OF MONTHS   ":N
```

```
100 INPUT "ANNUAL % INTEREST ":I
110 I=(0.01*I)/12
120 P=(T-R)*I/(1-1/(1+I)^N)
130 PP=INT(100*P+0.5)/100
140 PRINT
150 PRINT "PAYMENT $";PP
160 PRINT
170 PRINT
180 PRINT
190 INPUT "FOR MORE, PRESS ENTER":KY$
200 GOTO 10
210 FOR L=1 TO 18
220 PRINT "+";
230 NEXT L
240 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 CALL CLEAR
20 INPUT "MANUFACTURING COST $":C
30 INPUT "PRICING MULTIPLIER ":M
40 P=INT(C*M)+0.95
50 PRINT
60 PRINT "RETAIL PRICE $";P
70 PRINT
80 PRINT
90 GOTO 20
```

### Sample Run

```
MANUFACTURING COST $ .21
PRICING MULTIPLIER  6
RETAIL PRICE $ 1.95

MANUFACTURING COST $ 10.3
PRICING MULTIPLIER  4
RETAIL PRICE $ 41.95
```

# 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 CALL CLEAR
20 PRINT "ENTER A NUMBER"
30 PRINT "TO MORE THAN TWO"
40 INPUT "DECIMAL PLACES ":N
50 PRINT
60 R=INT(100*N+0.5)/100
70 PRINT "TO THE NEAREST PENNY,"
80 PRINT "$";N;" IS $";R
90 PRINT
100 PRINT
110 GOTO 20
```

# 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 CALL CLEAR
20 INPUT "WHOLESALE COST $":W
30 INPUT "MARK-UP PERCENTAGE ":P
40 D=1-0.01*P
50 R=W/D
60 PRINT
70 PRINT "RETAIL PRICE $";R
80 PRINT
90 PRINT
100 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 pro-

gram 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 CALL CLEAR
20 INPUT "LIST PRICE $":L
30 INPUT "PERCENTAGE OFF ":P
40 CALL CLEAR
50 D=1-0.01*P
60 PRINT "TO COMPUTE WITH"
70 PRINT P;"PERCENT OFF"
80 PRINT "USE THE DECIMAL";D
90 PRINT
100 PRINT P;"% OFF $";L
110 PRINT "RESULTS IN"
120 PRINT "A COST OF $";D*L
130 PRINT
140 PRINT
150 PRINT
160 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 CALL CLEAR
20 INPUT "QUANTITY ":P
```

```
30 INPUT "TOTAL COST $":C
40 T=C/P
50 IF T<1 THEN 70
60 GOTO 90
70 T=100*T
80 GOTO 140
90 PRINT "EACH COST $";T
100 PRINT
110 PRINT
120 PRINT
130 GOTO 20
140 PRINT "EACH COST";T;"CENTS"
150 GOTO 100
```

# 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.

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

## Program Listing

```
10 CALL CLEAR
20 INPUT "HOURLY PAY RATE $":P
30 INPUT "NUMBER HOURS WORKED ":H
40 IF H>40 THEN 60
50 GOTO 80
60 OT=H-40
70 GOTO 140
80 W=H*P
90 PRINT "GROSS WAGES $";W
100 PRINT
110 PRINT
120 PRINT
130 GOTO 20
140 W=(40*P)+(OT*P*1.5)
150 GOTO 90
```

## Sample Run

```
HOURLY PAY RATE $ 5.65
NUMBER HOURS WORKED  40
GROSS WAGES $ 226

HOURLY PAY RATE $ 10
NUMBER HOURS WORKED  80
GROSS WAGES $ 1000
```

# 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 CALL CLEAR
20 INPUT "QUANTITY SOLD ":Q
30 INPUT "UNIT PRICE $":P
40 INPUT "SALESTAX % RATE ":S
50 INPUT "SHIPPING CHARGES $":H
60 S=0.01*S
70 C=Q*P
80 T=C*S
90 F=C+T+H
100 CC=INT(100*C+0.5)/100
110 TT=INT(100*T+0.5)/100
120 FF=INT(100*F+0.5)/100
130 CALL CLEAR
140 PRINT "TOTAL PRICE","$";CC
150 PRINT "SALES TAX","$";TT
160 PRINT "SHIP CHARGE","$";H
```

```
170 PRINT
180 PRINT "INVOICE TOTAL","$";FF
190 FOR Z=1 TO 10
200 PRINT
210 NEXT Z
220 INPUT "TO DO ANOTHER, PRESS ENTER";KY$
230 GOTO 10
```

### Sample Run

```
QUANTITY SOLD  123
UNIT PRICE $ 1.99
SALESTAX % RATE  6
SHIPPING CHARGES $ 0

TOTAL PRICE    $ 244.77
SALES TAX      $ 14.69
SHIP CHARGE    $ 0
INVOICE TOTAL  $ 259.46

TO DO ANOTHER, PRESS ENTER

QUANTITY SOLD  5
UNIT PRICE $ 19.99
SALESTAX % RATE  7.5
SHIPPING CHARGES $ 3.69

TOTAL PRICE    $ 99.95
SALES TAX      $ 7.5
SHIP CHARGE    $ 3.69
INVOICE TOTAL  $ 111.14

TO DO ANOTHER, PRESS ENTER
```

# 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 quanti-

ty 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 CALL CLEAR
20 INPUT "ITEM NAME:   ":N$
30 INPUT "QUANTITY:    ":Q
40 INPUT "TOTAL PAID: $":P
50 U=P/Q
60 UU=INT(100*U+0.5)/100
70 UU$=STR$(UU)
80 PRINT "UNIT PRICE  $";UU$
90 FOR Z=1 TO 10
100 PRINT
110 NEXT Z
120 INPUT "TO DO MORE, PRESS ENTER":KY$
130 GOTO 10
```

### Sample Run

```
ITEM NAME:     WIDGET
QUANTITY:      895
TOTAL PAID:  $695
UNIT PRICE    $.78

ITEM NAME:     CRANK
QUANTITY:      25
TOTAL PAID:  $3.99
UNIT PRICE    $.16
```

# 78 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 ENTER 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 CALL CLEAR
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$=""
150 GOTO 100
160 CALL SOUND(100,1000,1)
170 CALL CLEAR
180 PRINT "WHOLESALE","$";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
270 PRINT
280 PRINT
290 INPUT "TO DO MORE, PRESS ENTER":KY$
```

```
300 WT=0
310 WD=0
320 RT=0
330 RD=0
340 GOTO 10
```

### Sample Run

```
WHOLESALE       $ 69
 3 ENTRIES

RETAIL          $ 90
 18 ENTRIES

TOTAL           $ 159
 21 ENTRIES
```

# 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.

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

### Program Listing

```
10 RANDOMIZE
20 CALL CLEAR
30 GOSUB 200
40 PRINT "SUNDAY",C
50 GOSUB 200
60 PRINT "MONDAY",C
70 GOSUB 200
80 PRINT "TUESDAY",C
90 GOSUB 200
100 PRINT "WEDNESDAY",C
110 GOSUB 200
120 PRINT "THURSDAY",C
```

```
130 GOSUB 200
140 PRINT "FRIDAY",C
150 GOSUB 200
160 PRINT "SATURDAY",C
170 CALL KEY(0,Z,X)
180 IF X=0 THEN 170
190 GOTO 20
200 C=INT(10000*RND)
210 IF C<1000 THEN 200
220 RETURN
```

### Sample Run

```
SUNDAY          3557
MONDAY          5166
TUESDAY         5814
WEDNESDAY       1234
THURSDAY        2844
FRIDAY          3677
SATURDAY        8839
```

# 80 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 CALL CLEAR
20 INPUT "AD COST:   ":A
30 INPUT "CIRCULATION:   ":C
40 M=1000*(A/C)
50 PRINT "$";M;"COST PER THOUSAND"
60 FOR Q=1 TO 10
70 PRINT
80 NEXT Q
90 GOTO 20
```

## Sample Run

```
AD COST:          2000
CIRCULATION:      300000
$ 6.666666667 COST PER THOUSAND

AD COST:          500
CIRCULATION:      10000
$ 50 COST PER THOUSAND

AD COST:          10
CIRCULATION:      175000
$ .0571428571 COST PER THOUSAND
```

# 81 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 CALL CLEAR
20 INPUT "AD COST:     ":A
30 INPUT "UNITS SOLD: ":U
40 C=A/U
50 PRINT "THE AD COST $";C;"PER UNIT"
60 FOR P=1 TO 11
70 PRINT
80 NEXT P
90 GOTO 20
```

# 82 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 CALL CLEAR
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
240 PRINT
250 NEXT Z
260 A=0
270 D=0
280 N=0
290 P=0
300 Q=0
310 GOTO 20
```

## Sample Run

```
SALE AMOUNT CENTS:    33
CENTS TENDERED:       50

 0 QUARTERS
 1 DIMES
 1 NICKLES
 2 PENNIES

SALE AMOUNT CENTS:    71
CENTS TENDERED:       80

 0 QUARTERS
 0 DIMES
 1 NICKLES
 4 PENNIES
```

# Colorful Graphics

# 83 Box The Screen

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 around the graphics display-screen area on your TV monitor. Lines 30 and 50 draw the vertical sides of the box while lines 20 and 40 draw the horizontal bottom and top. Line 60 is a freeze-frame loop to hold the picture so you can see it.

## Program Listing

```
10 CALL CLEAR
20 CALL HCHAR(1,1,64,32)
30 CALL VCHAR(1,32,64,24)
40 CALL HCHAR(24,1,64,32)
50 CALL VCHAR(1,1,64,24)
60 GOTO 60
```

# 84 Moving Illusion

Beware! This constantly-moving image may drive you batty.

## Program Listing

```
10 CALL CLEAR
20 CALL CHAR(128,"FFFFFFFFFFFFFFFF")
30 CALL SCREEN(9)
```

```
40 FOR L=1 TO 24
50 PRINT TAB(L);CHR$(128)
60 NEXT L
70 GOTO 40
```

# 85 Super Moving Illusion

If you liked Tip Number 84 above, you'll love this one! Here the background color changes as well as the color of the lines. Very striking!

## Program Listing

```
10 RANDOMIZE
20 CALL CLEAR
30 CALL CHAR(128,"FFFFFFFFFFFFFFFF")
40 SC=INT(16*RND)
50 IF SC<3 THEN 40
60 CALL SCREEN(SC)
70 CL=INT(17*RND)
80 IF CL<3 THEN 70
90 CALL COLOR(13,CL,SC)
100 FOR L=1 TO 24
110 PRINT TAB(L);CHR$(128)
120 NEXT L
130 GOTO 40
```

# 86 Giant Clock Exercise

More round graphics on your rectangular picture tube.

## Program Listing

```
10 CALL CLEAR
20 FOR N=1 TO 12
30 IF N>3 THEN 60
```

```
40 CX=49
50 GOTO 70
60 CX=32
70 CN=N+45
80 IF CN<49 THEN 100
90 GOTO 110
100 CN=CN+2
110 R=12-(10*SIN(N/6*3.14))
120 C=16-(10*COS(N/6*3.14))
130 CALL HCHAR(R,C-1,CX)
140 CALL HCHAR(R,C,CN)
150 NEXT N
160 GOTO 160
```

# 87 Aztec Art

Our program reminded us of *Aztec* artwork.

## Program Listing

```
10 CALL CLEAR
20 CALL SCREEN(11)
30 FOR X=10 TO 1 STEP -1
40 FOR N=1 TO 12
50 CN=INT(126*RND)
60 IF CN<33 THEN 50
70 R=12-(X*SIN(N/6*3.14))
80 C=16-(X*COS(N/6*3.14))
90 CALL HCHAR(R,C,CN)
100 NEXT N
110 NEXT X
120 GOTO 120
```

# 88 Circling Dot

More round graphics on your rectangular picture tube.

### Program Listing

```
10 CALL CLEAR
20 CALL CHAR(128,"FFFFFFFFFFFFFFFF")
30 CALL COLOR(1,9,6)
40 FOR N=1 TO 12
50 R=12-(10*SIN(N/6*3.14))
60 C=16-(10*COS(N/6*3.14))
70 CALL HCHAR(R,C,128)
80 FOR T=1 TO 75
90 NEXT T
100 CALL CLEAR
110 NEXT N
120 GOTO 40
```

# 89 Screen Filler

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

### Program Listing

```
10 CALL CLEAR
20 CALL CHAR(128,"FFFFFFFFFFFFFFFF")
30 CALL SCREEN(10)
40 C=INT(33*RND)
50 IF C<1 THEN 40
60 R=INT(25*RND)
70 IF R<1 THEN 60
80 CALL HCHAR(R,C,128)
90 GOTO 40
```

# 90 Window Twinklers

Well, what would you call them?

## Program Listing

```
10 CALL CLEAR
20 FC=INT(17*RND)
30 IF FC<3 THEN 20
40 CALL COLOR(1,FC,5)
50 CH=INT(38*RND)
60 IF CH<33 THEN 50
70 C=INT(33*RND)
80 IF C<1 THEN 70
90 R=INT(25*RND)
100 IF R<1 THEN 90
110 CALL HCHAR(R,C,CH)
120 GOTO 20
```

# 91 Show The Colors

This demonstrator program shows each of the available screen colors slowly. As they pass in review, watchers learn what is available. For a longer time of display of each color, increase the number 500 in line 40.

## Program Listing

```
10 CALL CLEAR
20 FOR C=3 TO 16
30 CC=C
40 FOR T=1 TO 500
50 NEXT T
60 IF C>9 THEN 80
70 GOTO 90
80 CC=CC-10
90 X=32
100 IF C>9 THEN 120
110 GOTO 130
120 X=49
130 CALL SCREEN(C)
140 CALL HCHAR(1,15,X)
150 CALL HCHAR(1,16,CC+48)
160 NEXT C
170 GOTO 20
```

# 92 Blackboard

This program appears to draw a blackboard on the screen. You can write messages on it, draw football plays, create art or words etc. on this electronic chalkboard.

## Program Listing

```
10 CALL CLEAR
20 CALL CHAR(128,"FFFFFFFFFFFFFFFF")
30 CALL SCREEN(9)
40 FOR C=10 TO 22
50 FOR R=5 TO 19
60 CALL HCHAR(R,C,128)
70 NEXT R
80 NEXT C
90 GOTO 90
```

# 93 Snowfall

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

## Program Listing

```
10 CALL CLEAR
20 CALL SCREEN(5)
30 CALL COLOR(2,16,5)
40 R=INT(25*RND)
50 IF R<1 THEN 40
60 C=INT(32*RND)
70 IF C<1 THEN 60
80 CALL HCHAR(R,C,42)
90 GOTO 40
```

# 94 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 CALL CLEAR
20 CALL CHAR(128,"FFFFFFFFFFFFFFFF")
30 FOR C=2 TO 32
40 CALL HCHAR(12,C,128)
50 FOR T=1 TO 25
60 NEXT T
70 CALL CLEAR
80 NEXT C
90 FOR C=31 TO 1 STEP -1
100 CALL HCHAR(12,C,128)
110 FOR T=1 TO 25
120 NEXT T
130 CALL CLEAR
140 NEXT C
150 GOTO 30
```

# 95 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 50 to 390 accept your up, down, right, or left commands, as U, D, R, or L. No other letters will work. Line 400 draws your lines.

## Program Listing

```
10 CALL CLEAR
20 CALL CHAR(128,"FFFFFFFFFFFFFFFF")
30 R=1
40 C=1
50 CALL KEY(0,Z,X)
60 IF X=0 THEN 50
70 IF Z=85 THEN 90
80 GOTO 150
90 R=R-1
100 IF R<1 THEN 120
110 GOTO 140
120 R=1
130 GOTO 50
140 GOTO 400
150 IF Z=68 THEN 170
160 GOTO 230
170 R=R+1
180 IF R>24 THEN 200
190 GOTO 220
200 R=24
210 GOTO 50
220 GOTO 400
230 IF Z=82 THEN 250
240 GOTO 310
250 C=C+1
260 IF C>32 THEN 280
270 GOTO 300
280 C=32
290 GOTO 50
300 GOTO 400
310 IF Z=76 THEN 330
320 GOTO 390
330 C=C-1
340 IF C<1 THEN 360
350 GOTO 380
360 C=1
370 GOTO 50
380 GOTO 400
390 GOTO 50
400 CALL HCHAR(R,C,128)
410 GOTO 50
```

# 96 Centered Boxed Titles

Here's how to dress up your program titles.

After you type in and RUN the program, press any key on your keyboard to do more.

## Program Listing

```
10 CALL CLEAR
20 INPUT "WHAT IS THE TITLE?  ":T$
30 LT=LEN(T$)
40 IF LT>20 THEN 60
50 GOTO 80
60 PRINT "OOPS, TOO LONG, TRY AGAIN"
70 GOTO 20
80 LB=LT+6
90 SP=((32-LB)/2)-1
100 AS$="*"
110 CALL CLEAR
120 PRINT TAB(SP-1);" ";
130 FOR L=1 TO LB
140 PRINT AS$;
150 NEXT L
160 PRINT TAB(SP);"** ";T$;" **"
170 PRINT TAB(SP-1);" ";
180 FOR L=1 TO LB
190 PRINT AS$;
200 NEXT L
210 FOR L=1 TO 15
220 PRINT
230 NEXT L
240 CALL KEY(0,Z,X)
250 IF X=0 THEN 240
260 GOTO 10
```

# 97 Five-Item Checklist

The computer asks you to give it the names of five items. It then prints them in a column with small check-off boxes alongside.

## Program Listing

```
10 CALL CLEAR
20 CALL CHAR(128,"FF8181818181FF")
30 DIM I$(5)
40 INPUT "FIRST ITEM:  ":I$(1)
50 INPUT "SECOND ITEM: ":I$(2)
60 INPUT "THIRD ITEM:  ":I$(3)
70 INPUT "FOURTH ITEM: ":I$(4)
80 INPUT "FIFTH ITEM:  ":I$(5)
90 PRINT
100 PRINT
110 PRINT
120 PRINT
130 PRINT "CHECKLIST"
140 FOR L=1 TO 9
150 PRINT CHR$(126);
160 NEXT L
170 PRINT
180 FOR L=1 TO 5
190 PRINT CHR$(128);" ";I$(L)
200 NEXT L
210 CALL KEY(0,Z,X)
220 IF X=0 THEN 210
230 GOTO 10
```

## Sample Run

```
☐ PENCIL
☐ PAPER
☐ INK
☐ PEN
☐ NOTEBOOK
```

☐ RADIO
☐ TELEVISION
☐ NEWSPAPER
☐ MAGAZINE
☐ BOOKS

☐ DOG
☐ CAT
☐ HORSE
☐ COW
☐ SHEEP

☐ RAIN
☐ SNOW
☐ SLEET
☐ HAIL
☐ SLUSH

☐ HAT
☐ COAT
☐ GLOVES
☐ SCARF
☐ BOOTS

# 98 Flashing Graphics Cursor

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.

Use this flashy little indicator to spot whatever you like on the graphics screen. Change the location of the cursor spot by changing the two 12s in line 50.

## Program Listing

```
10 CALL CLEAR
20 CALL CHAR(128,"FFFFFFFFFFFFFFFF")
30 FOR C=3 TO 16
```

```
40 CALL COLOR(13,C,4)
50 CALL HCHAR(12,12,128)
60 NEXT C
70 GOTO 30
```

# 99 Color Bar Graph Generator

The bar graph generated by this program can have up to 20 bars. The lengths of the bars are limited to the range of zero to 23. You type in the letter X to end the input loop. The bars are numbered sequentially, from top to bottom, starting with number one.

## Program Listing

```
10 CALL CLEAR
20 DIM R$(20)
30 DIM NM$(20)
40 CALL CHAR(128,"FFFFFFFFFFFFFFFF")
50 PRINT "WHAT IS THE TITLE"
60 INPUT "OF THE CHART?  ":T$
70 N=N+1
80 PRINT
90 INPUT "BAR LENGTH (1-23)?  ":R$(N)
100 IF R$(N)="X" THEN 160
110 IF VAL(R$(N))>23 THEN 130
120 GOTO 150
130 PRINT "OOPS, TOO LONG, TRY AGAIN"
140 GOTO 90
150 GOTO 180
160 R$(N)="0"
170 GOTO 210
180 NM$(N)=STR$(N)
190 IF N+1=21 THEN 210
200 GOTO 70
210 CALL CLEAR
220 CALL SCREEN(5)
230 FOR W=1 TO 13
```

```
240 CALL COLOR(W,16,5)
250 NEXT W
260 LT=LEN(T$)
270 TP=INT((32-LT)/2)
280 PRINT TAB(TP);T$
290 PRINT
300 FOR K=1 TO N
310 PRINT NM$(K);" ";
320 NR=VAL(R$(K))
330 PRINT TAB(4);" ";
340 FOR J=1 TO NR
350 PRINT CHR$(128);
360 NEXT J
370 PRINT
380 NEXT K
390 CALL KEY(0,Z,X)
400 IF X=0 THEN 390
410 FOR P=1 TO 20
420 R$(P)=""
430 NM$(P)=""
440 N=0
450 NEXT P
460 CALL CLEAR
470 CALL SCREEN(4)
480 FOR W=1 TO 13
490 CALL COLOR(W,2,1)
500 NEXT W
510 GOTO 10
```

# 100 Draw Bar Graphs

Drawing graphs on the video screen are a popular form of communication today. This program establishes a bar graph on the computer display.

We have selected the business-like example, shown here, to demonstrate how you go about setting up a bar graph on the TV screen.

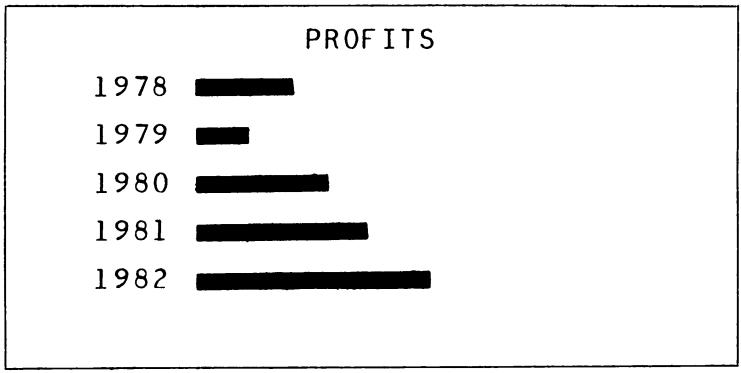
After a run, the computer awaits your press of any key to do another.

## Program Listing

```
10 CALL CLEAR
20 CALL CHAR(128,"FFFFFFFFFFFFFFFF")
30 INPUT "1978 PROFITS? ":A
40 IF A>23 THEN 30
50 INPUT "1979 PROFITS? ":B
60 IF B>23 THEN 50
70 INPUT "1980 PROFITS? ":C
80 IF C>23 THEN 70
90 INPUT "1981 PROFITS? ":D
100 IF D>23 THEN 90
110 INPUT "1982 PROFITS? ":E
120 IF E>23 THEN 110
130 CALL CLEAR
140 PRINT TAB(11);"PROFITS"
150 PRINT
160 PRINT
170 PRINT "1978 ";
180 FOR L=1 TO A
190 PRINT CHR$(128);
200 NEXT L
210 PRINT
220 PRINT
230 PRINT "1979 ";
240 FOR L=1 TO B
250 PRINT CHR$(128);
260 NEXT L
270 PRINT
280 PRINT
290 PRINT "1980 ";
300 FOR L=1 TO C
310 PRINT CHR$(128);
320 NEXT L
330 PRINT
340 PRINT
350 PRINT "1981 ";
360 FOR L=1 TO D
370 PRINT CHR$(128);
```

```
380 NEXT L
390 PRINT
400 PRINT
410 PRINT "1982 ";
420 FOR L=1 TO E
430 PRINT CHR$(128);
440 NEXT L
450 PRINT
460 CALL KEY(0,Z,X)
470 IF X=0 THEN 460
480 GOTO 10
```

### Sample Run




# 101 Backward Writer

Give the computer a message of up to 99 characters. The computer will print the message backward. Then press R to repeat the same message. Or press W to start over with a new message.

The computer is simply amazing!

### Program Listing

```
10 CALL CLEAR
20 DIM X$(100)
```

```
30 PRINT "TYPE A MESSAGE"
40 INPUT A$
50 IF LEN(A$)>100 THEN 70
60 GOTO 90
70 PRINT "OOPS, TOO LONG, TRY AGAIN"
80 GOTO 30
90 L=LEN(A$)
100 CALL CLEAR
110 FOR J=L+1 TO 1 STEP -1
120 X$(J)=SEG$(A$,J,1)
130 PRINT X$(J);
140 NEXT J
150 FOR Q=1 TO 10
160 PRINT
170 NEXT Q
180 PRINT "PRESS R TO REPEAT BACKWARD"
190 PRINT "PRESS M TO SEE ORIGINAL"
200 PRINT "PRESS W TO WRITE NEW"
210 PRINT "PRESS Q TO QUIT"
220 CALL KEY(0,Z,X)
230 IF X=0 THEN 220
240 IF Z=82 THEN 100
250 IF Z=77 THEN 300
260 IF Z=87 THEN 10
270 IF Z=81 THEN 370
280 CALL CLEAR
290 GOTO 180
300 CALL CLEAR
310 PRINT A$
320 FOR G=1 TO 10
330 PRINT
340 NEXT G
350 GOTO 180
360 GOTO 10
370 CALL CLEAR
380 PRINT "BYE BYE"
```

# Appendix A: BASIC Words

| | |
|---|---|
| ABS | absolute value |
| ASC | ASCII number of string's first character |
| ATN | trig arctangent |
| BREAK | stops program run |
| BYE | leaves BASIC |
| CALL CHAR | redefines ASCII characters |
| CALL CLEAR | erases video display |
| CALL COLOR | sets video colors |
| CALL GCHAR | finds video location contents |
| CALL HCHAR | places character on video screen |
| CALL JOYST | joystick input |
| CALL KEY | find keypress; like INKEY$ |
| CALL SCREEN | change video screen color |
| CALL SOUND | causes tones, noise |
| CALL VCHAR | places character on video screen |
| CHR$ | changes number to character |
| CLOSE | shuts a file |
| CON | same as CONTINUE |
| CONTINUE | resume run after BREAK |
| COS | trig cosine |
| DATA | stores numbers, letters in program |
| DEF | user-defined function |
| DELETE | removes program or file |
| DIM | dimensions an array |
| DISPLAY | PRINT |
| EDIT | displays line for changing |
| END | concludes program execution |
| EOF | end of file |
| EXP | exponential value $e^x$ |
| FOR | FOR/NEXT loop |
| GOSUB | move to subroutine |
| GOTO | move to line number |
| IF | IF/THEN decision maker |
| ELSE | IF/THEN/ELSE decision maker |
| INPUT | takes in info |
| INPUT# | takes in info from external file |
| INT | finds integer of number |
| LEN | changes character to number |

| | |
|---|---|
| LET | optional; assign value to variable |
| LIST | display program lines |
| LOG | natural logarithm |
| NEW | empty the program memory |
| NEXT | FOR/NEXT loop |
| NUM | same as NUMBER |
| NUMBER | automatic line numbering |
| OLD | load from mass storage device to memory |
| ON | ON/GOSUB or ON/GOTO |
| OPEN | establish a file |
| OPTION BASE | lowest array subscript |
| POS | position of a substring |
| PRINT | output to display or external device |
| RANDOMIZE | shuffle random number generator |
| READ | finds DATA lines and moves info |
| REC | record number; with PRINT |
| REM | remarks |
| RES | same as RESEQUENCE |
| RESEQUENCE | renumber program lines |
| RESTORE | resets DATA/READ |
| RETURN | go back to main program after GOSUB |
| RND | random number generator |
| RUN | start program execution |
| SAVE | copy program to external file |
| SEG$ | finds substring |
| SGN | find whether number is positive or negative |
| SIN | trig sine |
| SQR | square root |
| STEP | FOR/NEXT loop increment control |
| STOP | ends program run |
| STR$ | changes number to string |
| TAB | control PRINT or DISPLAY location |
| TAN | trig tangent |
| THEN | IF/THEN decision maker |
| TO | FOR/NEXT loop |
| TRACE | debug program lines |
| UNBREAK | remove breakpoints |
| UNTRACE | cancels TRACE |
| VAL | changes string to number |

# Appendix B: Character Sets

| Set | ASCII Code |
|---|---|
| 1 | 32-39 |
| 2 | 40-47 |
| 3 | 48-55 |
| 4 | 56-63 |
| 5 | 64-71 |
| 6 | 72-79 |
| 7 | 80-87 |
| 8 | 88-95 |
| 9 | 96-103 |
| 10 | 104-111 |
| 11 | 112-119 |
| 12 | 120-127 |
| 13 | 128-135 |
| 14 | 136-143 |
| 15 | 144-151 |
| 16 | 152-159 |

# Appendix C: Color Codes

| Number | Color |
|---|---|
| 1 | Transparent |
| 2 | Black |
| 3 | Medium green |
| 4 | Light green |
| 5 | Dark blue |
| 6 | Light blue |
| 7 | Dark red |
| 8 | Cyan |
| 9 | Medium red |
| 10 | Light red |
| 11 | Dark yellow |
| 12 | Light yellow |
| 13 | Dark green |
| 14 | Magenta |
| 15 | Gray |
| 16 | White |

# Computer books from ARCsoft Publishers

**For the TIMEX/Sinclair 1000, Sinclair ZX-81 and MicroAce:**

Practical TIMEX/Sinclair Computer Programs for Beginners
Edward Page $7.95 ISBN 0-86668-027-6

101 TIMEX 1000/Sinclair ZX-81 Programming Tips & Tricks
Edward Page $7.95 ISBN 0-86668-020-9

TIMEX/Sinclair Computer Games Programs
Edward Page $7.95 ISBN 0-86668-026-8

37 TIMEX 1000/Sinclair ZX-81 Programs for Home, School & Office
Edward Page $8.95 ISBN 0-86668-021-7

TIMEX/Sinclair Computer Program Writing Workbook
Edward Page $4.95 ISBN 0-86668-810-2

**For the Texas Instruments TI-99/4A Home Computer:**

101 Programming Tips & Tricks for the Texas Instruments TI-99/4A
Len Turner $8.95 ISBN 0-86668-025-X

36 Texas Instruments TI-99/4A Programs for Home, School & Office
Len Turner $8.95 ISBN 0-86668-024-1

Texas Instruments Computer Program Writing Workbook
Len Turner $4.95 ISBN 0-86668-812-9

**For the VIC-20 and Commodore computers:**

101 Programming Tips & Tricks for the VIC-20
Howard Adler $8.95 ISBN 0-86668-030-6

34 VIC-20 Computer Programs for Home, School & Office
Howard Adler $8.95 ISBN 0-86668-029-2

VIC-20 Computer Program Writing Workbook
Howard Adler $4.95 ISBN 0-86668-811-0

**For the ATARI 400, 600, 800, 1200XL computers:**

101 ATARI Computer Programming Tips & Tricks
Alan North $8.95 ISBN 0-86668-022-5

31 New ATARI Computer Programs for Home, School & Office
Alan North $8.95 ISBN 0-86668-018-7

ATARI Computer Program Writing Workbook
Alan North $4.95 ISBN 0-86668-814-5

**For the TRS-80 Color Computer and TDP-100 computers:**

Color Computer Graphics
Ron Clark $9.95 ISBN 0-86668-012-8

101 Color Computer Programming Tips & Tricks
Ron Clark $7.95 ISBN 0-86668-007-1

55 Color Computer Programs for Home School & Office
Ron Clark $9.95 ISBN 0-86668-005-5

55 MORE Color Computer Programs for Home, School & Office
Ron Clark $9.95 ISBN 0-86668-008-X

The Color Computer Songbook
Ron Clark $7.95 ISBN 0-86668-011-X

TRS-80 Color Computer Program Writing Workbook
Ron Clark $4.95 ISBN 0-86668-816-1

My Buttons Are Blue and Other Love Poems
Edited by Ron Clark $4.95 ISBN 0-86668-013-6

**For the APPLE and Franklin ACE computers:**
101 APPLE Computer Programming Tips & Tricks
Fred White $8.95 ISBN 0-86668-015-2

33 New APPLE Computer Programs for Home, School & Office
Fred White $8.95 ISBN 0-86668-016-0

APPLE Computer Program Writing Workbook
Fred White $4.95 ISBN 0-86668-813-7

**For the TRS-80, Sharp and Casio Pocket Computers:**
Practical PC-2/PC-1500 Pocket Computer Programs
Jim Cole $7.95 ISBN 0-86668-028-4

Pocket Computer Programming Made Easy
Jim Cole $8.95 ISBN 0-86668-009-8

99 Tips & Tricks for the New Pocket Computers
Jim Cole $7.95 ISBN 0-86668-019-5

101 Pocket Computer Programming Tips & Tricks
Jim Cole $7.95 ISBN 0-86668-004-7

Murder In The Mansion and Other Computer Adventures—2nd Edition
Jim Cole $6.95 ISBN 0-86668-501-4

50 Programs in BASIC for Home, School & Office—2nd Edition
Jim Cole $9.95 ISBN 0-86668-502-2

50 MORE Programs in BASIC for Home, School & Office
Jim Cole $9.95 ISBN 0-86668-003-9

Pocket Computer Program Writing Workbook
Jim Cole $4.95 ISBN 0-86668-817-X

35 Practical Programs for the CASIO Pocket Computer
Jim Cole $8.95 ISBN 0-86668-014-4

**For the TRS-80 Model I and Model III computers:**
TRS-80 Model I/III Computer Program Writing Workbook
Mark Lewis $4.95 ISBN 0-86668-815-3

**For the IBM Personal Computer:**
IBM Personal Computer Program Writing Workbook
George Bridges $4.95 ISBN 0-86668-818-8

**Universal for use with any computer system:**
Universal BASIC Computer Program Writing Workbook
Don Roberts $4.95 ISBN 0-86668-819-6

ISBN: International Standard Book Number


# ARCsoft Publishers
**Woodsboro, Maryland 21798**

Texas Instruments Home Computer Book II

# 101 Programming Tips & Tricks For The Texas Instruments TI-99/4A

by Len Turner

Here's a giant collection of practical, useful, efficient programming techniques and operating shortcuts, for the TI-99/4A Home Computer, TI-99/2 Home Computer, Compact 40 Computer and other Texas Instruments personal, home and business microcomputers, right out of a master programmer's notebook.

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

Each of the 101 computer software tips in this book features a complete 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 TI-99/4A Home Computer and are ready to type in and run.

Learn insider's how-to secrets for using the BASIC program language, including the exciting special words READ, DATA, RESTORE, INPUT, GOSUB, RETURN, VAL, ASC, CHR$, PRINT, DIM, FOR, NEXT, IF, THEN, CALL CLEAR, CALL COLOR, CALL CHAR, CALL KEY, CALL SOUND, LEN, SEG$, STR$, TAB and many more.

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

You will find special techniques for graphics, colorful program titles and billboards, coin toss, dice throw, secret message, sound off, mystery clues, code groups, 60 second timer, sorting, ordering, searching, averaging, passwords, sentence writing, creating tables, memory testing, exam sorting, rounding off, random numbers, shopper's friend, interest on money, car payment, wages & hours, percentage off and markup, invoicing, inventory counter, lists, and many more.

You'll learn what to do, how to do it, when to make changes and when not to, all from this info-packed TI computer programmer's handbook of tips and tricks.

ARCsoft Publishers
WOODSBORO, MARYLAND

ISBN 0-86668-025-X