

Radio Shack

One Dollar and Ninety-Five Cents

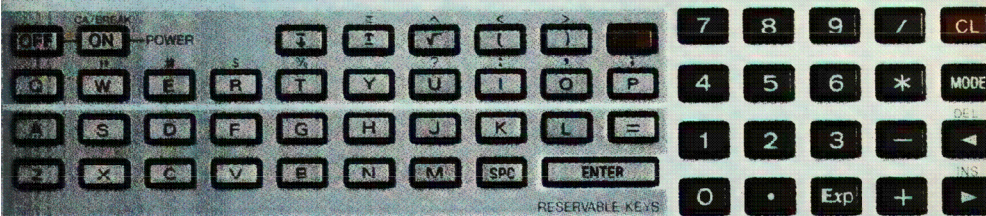
Cat. No. 62-2086

TRS-80[®]

POCKET COMPUTER PROGRAMS

5: INPUT B,C:A=√(B*B+C*C)

Radio Shack
TRS-80
POCKET
COMPUTER



By Jim Cole

50 ready-to-run programs in Pocket-BASIC, including:

BUSINESS Hourly Wages/Overtime • Pocket Datebook • Unit Cost • Commissions • Advertising Cost-per-1000 •

EDUCATION Math Drill • Metric Conversion • Averages • Temperature Conversion • Slope & Distance • Factors •

HOME Compound Interest • Checking Account Balance • Simple Interest • Mortgage Loan • Whodunit? • Craps • Old West Shootout • Klingon Killer • And many more!

TRS-80[®] POCKET COMPUTER PROGRAMS

By Jim Cole

PRINTED IN USA FOR

Radio Shack

A DIVISION OF TANDY CORPORATION
FORT WORTH, TEXAS 76102

© 1981 by ARCsoft Publishers, Woodsboro, MD 21798

Reproduction of the content of this book, in any manner,
without express permission of the publisher, is prohibited.
No liability is assumed with respect to the use of the
information herein.

ISBN 0-86668-902-8

Preface

We are convinced that the Pocket Computer is the wave of the future. Miniaturization and portability continue to be major design goals in all areas of electronics: TV, stereo, amateur radio, calculators, auto, home, etc. The same applies to computers.

It's not hard to visualize engineers marrying the Pocket Computer with handheld printer technology since small printers already exist in pocket calculators. And it's not hard to imagine memory chips which could expand the present 1.9k of memory to 16k or even more RAM, plus an elaborate, friendly operating system atop a superexpanded BASIC in ROM. Even the miniature, flat-screen LCD television display technology could be used for graphics and multi-line displays. All this in the handheld package!

The "small" desktop computers of the late 1970s and early 1980s will be the museum pieces of the late 80s and early 90s.

Jim Cole

Table of Contents

- 6 Introduction**
- 9 Business Programs**
 - 9 Profit Estimator
 - 14 Hourly Wages, With Overtime
 - 15 Pocket Datebook
 - 18 Advertising Cost-per-Thousand
 - 19 Advertising Cost-per-Unit Sold
 - 19 Unit Cost
 - 21 Mark-Up: Standard Percentage
 - 21 Mark-Up: Manufacturing to Retail
 - 23 Mark-Up: Wholesale to Retail
 - 23 Making Change
 - 25 Invoice Totaling
 - 26 Salesman's Commissions
 - 27 Rounding Cents to Nearest Penny
 - 28 Yes/No Executive Decision Maker
 - 29 Adding Numbers: One-Key Entry
- 30 Education Programs**
 - 30 Grade Scoring
 - 32 Slope and Distance
 - 34 dB
 - 35 Flash Cards Math Drill
 - 39 Smart Computer
 - 40 Frequency and Wavelength
 - 41 Metric Converter
 - 44 Factors

- 45 Ohm's Law
- 47 Reciprocals
- 47 Memory Tester
- 49 Pythagorean Theorem
- 50 Averages
- 52 Temperature Conversion (F,C,K)
- 53 Rounding Off Numbers
- 55 Fractional Portion of a Number
- 55 Which Number is Largest?

57 Programs For The Home

- 57 Deposit Doubler
- 58 Simple Interest
- 59 Compound Interest: Present Value
- 60 Compound Interest: Future Value
- 63 Timer
- 64 Checking Account Monthly Balancer
- 66 Mortgage Loan
- 67 Random Number Generators
- 71 Random Numbers: Averages
- 72 Beep Commander
- 72 Whodunit?
- 73 Klingon Killer
- 74 Parameters
- 75 Craps
- 76 Cylon Raider Alert
- 78 Old West Shootout

Introduction

There's strong demand for *useful* software for microcomputers. Games have bloomed everywhere while practical, serviceable programs for the home, school and office have been in short supply. This volume, second in our series of program books written and tested on the *TRS-80 Pocket Computer*, provides more than 50 ready-to-run work-doers and time-savers.

As noted in the first volume in this series, *Murder In The Mansion and Other Computer Adventures In Pocket-BASIC For The TRS-80*, working in the Pocket Computer's version of the BASIC language requires a sharp editing pencil. Remarks, software explanations and lengthy strings are out. Honing, fine tuning and waste trimming are in. Take *Profit Estimator*, one of our handy business-office programs in this volume. When the idea occurred to me, I wrote it out at great length on a *Pocket-BASIC Coding Form* programming worksheet. It originally required *five* worksheets! That's 150 program lines. But that was only the warmup. The real work was in pruning it down to a slim, trim 38 lines of code.

Cutting the fat out of programs for the Pocket Computer is worthwhile since I now can retain several different work programs in the computer at the same time. Using the *Def* mode, I can call on just the right program for the right job at the right time.

Even though they may be headed toward the same goal, no two programmers will write exactly the same list

of lines in BASIC. I know, as you load the programs in this book into your computer, you'll make modifications to suit your own needs or interests. The author would like to hear of improvement suggestions as well as ideas for future volumes in this series. You can write the author in care of *ARCsoft Publishers*, P.O. Box 132, Woodsboro, MD 21798.

These programs will run on *any* microcomputer set up for BASIC. If you are using a system other than the TRS-80 Pocket Computer or Sharp PC-1211, be on the lookout for statements which differ in the version of BASIC used by your machine.

For example, PRINT and PAUSE commands used here may be different on your machine. Line numbers may be required rather than labels sometimes used here. ABC multiplication may have to be changed to A*B*C format. Logical tests used here may have to be stated differently on other computers. Your machine might not accept symbols for PI and square root and SIN as we use them here. You may have to renumber program lines if your system requires it.

The BASIC set used on the Pocket Computer is very powerful. Not at all a toy, the Pocket Computer is able to squeeze very elaborate, useful programs into its 1.9k memory. Editing is easier than on most other systems. Running is a joy. Slow speed is a worthwhile payment for hand-size portability.

Many of the programs in this book could be used as portions of larger listings, as subroutines or GOTO objects. To do so, make appropriate changes to the first line (usually 10) and the last line of each program. If you wish to create a subroutine, RETURN must be the last line of the small program. If you work one of these into a larger program as a subroutine, be especially careful of your memory labels. They must agree with your selection of labels for the main program.

It's convenient for the businessman, teacher, student, housewife or other Pocket Computer user to retain a number of favorite programs in the machine at the same time. This is especially easy to use in the *Def* mode.

For example, suppose you are a school teacher with a small mail-order business on the side. You keep *Grade*

Scorer, Profit Estimator, Metric Converter and *Whodunit?* all loaded into your Pocket Computer at the same time. *Grade Scorer* is handy for summarizing student test results. *Profit Estimator* is useful when you want to know roughly how much you can expect to make from your next mail-order ad campaign. *Metric Converter* helps your students and helps you around the house. And *Whodunit?* beats jogging for lunchtime fun.

From the listings in this book, you can see that *Grade Scorer* has a "G" in its first line. *Profit Estimator* starts with a "B". *Metric Converter*, an "=" and *Whodunit?* an "M".

To use any of these in any order you want, select the *Def* mode and key in SHIFT G or SHIFT B or SHIFT = or SHIFT M. Depending upon whether or not you keyed in G, B, = or M, the program you wanted will start. The *Def* mode is especially convenient and time-saving.

Table 1 shows how we reproduce the alphanumeric and special symbols used on the Pocket Computer.

Table 1: Symbols In This Book

\$	DOLLAR SIGN	:	COLON
*	ASTERISK	;	SEMICOLON
()	PARENTHESES	,	COMMA
+	PLUS SIGN	.	PERIOD
=	EQUALS	?	QUESTION MARK
-	MINUS SIGN	/	SLASH BAR
0	ZERO	π	PI
∅	ZERO	>	GREATER THAN
o	LETTER O	<	LESS THAN
!	EXCLAMATION POINT	^	EXPONENT CARET
"	QUOTATION MARK	¥	YEN (SHIFT Y)
#	NUMBER SYMBOL	%	PERCENT SYMBOL

Profit Estimator

How much profit will I generate if I spend this amount on advertising? A question frequently confronting the business person. Whether you sell large lots at wholesale or individual items via mail order, this program gives you a fast estimate of potential profits to be realized from various levels of advertising-dollar expenditures. It allows fast comparisons when quick decisions are needed.

Manufacturing and advertising costs are included. Labor and overhead are not.

For large-lot wholesale movement of goods, it gives you the information you need to compare quantity sold vs. discounts allowed.

In advertising response, it allows comparison of ad costs vs. sales. It also calculates an approximate break-even point when you tell it that the profit you want is \$1.

To judge mail-order response, it permits comparisons of mailing costs vs. size of mailings vs. rate of returns on orders.

A useful tool for small business, whether a local furniture store or a national mail-order house.

Program Listing

```
10 "B" CLEAR
20 INPUT" W OR D OR M ?",Q$
30 INPUT"MFG COST = $";C
40 INPUT"QUANTITY MFGD = ";P
50 INPUT"LIST PRICE = $";L
60 INPUT"TOTAL SOLD = ";S
70 IF Q$="D" THEN "¥"
80 IF Q$="M" THEN "#"
85 IF S=Ø THEN 30
90 INPUT"WHLSL DISCOUNT % = ";D
100 W=LS*((100-D)/100)
110 G=LS*((100-D)/100)-CP
120 PRINT"WHLSL INVOICE = $";W
130 PRINT"WHLSL PROFIT = $";G
140 GOTO 60
150 "#" INPUT"AD COST/INSERT = $";A
160 INPUT"TOTAL INSERTIONS = ";I
170 IF S THEN "π"
180 INPUT"PROFIT WANTED = $";F
190 IF F=Ø THEN 30
200 B=(F+AI+PC)/L:B=INT B+1
210 PRINT"SELL ";B;" PROFIT $";F
220 GOTO 180
230 "π" N=SL:N=INT N
240 E=SL-PC-AI:E=INT E
250 PRINT"GROSS ORDERS = $";N
260 PRINT"DRCT ML PROFIT = $";E
270 GOTO 60
280 "¥" INPUT"FLYERS MAILED = ";K
290 INPUT"FLYER PRINT COST = $";R
300 INPUT"POSTAGE COST = $";M
310 J=INT (100*S/K)
320 T=LS-(R+M+PC):T=INT T
330 U=LS
340 PRINT" D-MAIL RETURN % = ";J
350 PRINT" D-MAIL GROSS = $";U
360 PRINT" D-MAIL PROFIT = $";T
370 GOTO 60
```

Sample Run ADVERTISING ORDER RESPONSE

DEF MODE

SHIFT B
W OR D OR M?
M ENTER

MFG COST = \$
.89 ENTER

QUANTITY MFGD =
1000 ENTER

LIST PRICE = \$
5.95 ENTER

TOTAL SOLD =
300 ENTER

AD COST/INSERT = \$
330 ENTER

TOTAL INSERTIONS =
3 ENTER

GROSS ORDERS = \$1785
DRCT ML PROFIT = \$-95

TOTAL SOLD =
500 ENTER

AD COST/INSERT = \$
ENTER

TOTAL INSERTIONS =
ENTER

GROSS ORDERS = \$2975
DRCT ML PROFIT = \$1095

TOTAL SOLD =
Ø ENTER

AD COST/INSERT = \$
330 ENTER

TOTAL INSERTIONS =
3 ENTER

PROFIT WANTED = \$
1 ENTER

SELL 317. PROFIT \$1

PROFIT WANTED = \$
1000 ENTER

SELL 485. PROFIT \$1000

Sample Run WHOLESALE PROFIT ESTIMATOR

DEF MODE

SHIFT B
W OR D OR M?
W ENTER

MFG COST = \$
.89 ENTER

QUANTITY MFGD =
1000 ENTER

LIST PRICE = \$
5.95 ENTER

TOTAL SOLD =
300 ENTER

WHL SL DISCOUNT % =
40 ENTER

WHL SL INVOICE = \$1071
WHL SL PROFIT = \$181

TOTAL SOLD =
500 ENTER

WHLSSL DISCOUNT % =
50 ENTER

WHLSSL INVOICE = \$1487.50
WLSL PROFIT = \$597.50

TOTAL SOLD *
Ø ENTER

MFG COST = \$

Sample Run DIRECT MAIL RESPONSE

DEF MODE

SHIFT B
W OR D OR M?
D ENTER

MFG COST = \$
.89

QUANTITY MFGD =
1000 ENTER

LIST PRICE = \$
5.95 ENTER

TOTAL SOLD =
50 ENTER

FLYERS MAILED =
200 ENTER

FLYER PRINT COST = \$
15 ENTER

POSTAGE COST = \$
.15*200 ENTER

D-MAIL RETURN % =25
D-MAIL GROSS = \$297.50
D-MAIL PROFIT = \$-638

ENTER
TOTAL SOLD =
500 ENTER

FLYERS MAILED =
5000 ENTER

FLYER PRINT COST = \$
75 ENTER

POSTAGE COST = \$
.15*5000 ENTER

D-MAIL RETURN % = 10
D-MAIL GROSS = \$2975
D-MAIL PROFIT = \$1260

Hourly Wages, With Overtime

Payroll bookkeeping is quick and simple with this hourly wage finder. The program knows that overtime starts after 40 hours and calculates accordingly.

To modify this program, you'll want to know that P is the employee's hourly wage rate; X is overtime pay at time-and-a-half; W is total hours worked; R is regular non-overtime hours worked; Q is overtime hours worked (in other words, $Q=W-40$); and C is the gross amount of the paycheck before tax and other deductions.

Program Listing

```
10 "H" CLEAR
20 BEEP 1:INPUT"HOURLY PAY = $";P
30 BEEP 1:INPUT"HOURS WORKED = ";W
40 IF W>40 LET Q=W-40:R=40:GOTO 60
50 R=W:Q=Ø
```



```

60 X=1.5P:C=PR+XQ
70 BEEP 2:PRINT"GROSS PAY = $";C
80 GOTO 10

```

Sample Run

```

DEF MODE
      SHIFT H
BEEP   HOURLY PAY = $
      3.35 ENTER
BEEP   HOURS WORKED =
      53 ENTER
BEEPBEEP GROSS PAY = $199.325

```

Pocket Datebook

Imagine your Pocket Computer is a 20-page notebook. Each page has space for seven letters or numbers. It makes a convenient file for data you will need to recall or calendar for dates you need to remember: birthdays, anniversaries, meetings.

The program includes "error trapping" which means it won't accept a command to read from a page number larger than 20. Nor will it write, in its present configuration, beyond page 20.

The program searches for an empty page to write on. If none is available, it displays a table of contents and asks which page you would like to replace with new data. Remember to abbreviate your data down to combinations of seven letters and numbers.

Program Listing

```

10 "D"
20 PAUSE" DO YOU WANT TO..."
30 PAUSE" WRITE IN..."
40 PAUSE" OR READ"
50 PAUSE" THE POCKET DATEBOOK ?"
60 BEEP 1
   :INPUT"WRITE OR READ (W OR R)?",US

```

```

70 IF U$="W" THEN "W"
80 IF U$="R" THEN "R"
90 GOTO 60
100 "W" V=Ø
110 V=V+1
120 IF V>20 THEN 150
130 IF A$(V) PAUSE" SEARCHING FOR EMPTY PAGE"
    :GOTO 110
140 GOTO 180
150 PAUSE"ALL 20 PAGES FULL":GOSUB 220
160 PAUSE"WHICH TO REPLACE ?"
170 BEEP 1:INPUT"WHICH PAGE (1 TO 20) ?",V
175 IF V>20 THEN 170
180 BEEP 1:INPUT"WRITE NEW INFO",A$(V)
190 BEEP 1:INPUT"WANT TO WRITE MORE ?",X$
200 IF X$="YES" THEN "W"
210 GOTO 60
220 PAUSE"TABLE OF CONTENTS"
230 FOR V=1 TO 20
240 Z$=" "
250 PAUSE V;Z$;A$(V)
260 NEXT V
270 RETURN
300 "R"
310 PAUSE"TABLE OF CONTENTS"
320 FOR V=1 TO 20
330 IF A$(V) THEN 350
340 GOTO 370
350 Z$=" ":PAUSE V;Z$;A$(V)
360 NEXT V
370 PAUSE"WHICH PAGE TO SEE ?"
380 BEEP 1:INPUT"PAGE (1 TO 20) = ";W
390 IF W>20 THEN 380
400 PRINT A$(W)
410 BEEP 1
    :INPUT"WANT TO READ MORE ?",X$
420 IF X$="YES" THEN 380
430 GOTO 60

```

Sample Run

```

DEF MODE
      SHIFT D.

```

DO YOU WANT TO...
WRITE IN...
OR READ
THE POCKET DATEBOOK ?
BEEP WRITE OR READ (W OR R)?
W ENTER
BEEP WRITE NEW INFO
BJ10/21 ENTER
BEEP WANT TO WRITE MORE?
YES ENTER
BEEP SEARCH FOR EMPTY PAGE
WRITE NEW INFO
ES 3/21 ENTER
BEEP WANT TO WRITE MORE?
NO ENTER
BEEP WRITE OR READ (W OR R)?
R ENTER
TABLE OF CONTENTS
1. BJ10/21
2. ES 3/21
WHICH PAGE TO SEE?
BEEP PAGE(1 TO 20)=
2 ENTER
ES 3/21
ENTER
WANT TO READ MORE?
NO ENTER

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 "C" CLEAR
20 INPUT"AD COST = $";A
30 INPUT"CIRCULATION = ";C
40 M=1000*(A/C)
50 PRINT" $";M;" COST/THOUSAND"
60 GOTO 10
```

Sample Run

```
DEF MODE
      SHIFT C

      AD COST = $
      330 ENTER
      CIRCULATION =

      160000 ENTER
      $2.0625 COST/THOUSAND
      ENTER
      AD COST = $
      133 ENTER
      CIRCULATION =
      51000 ENTER
      $2.607843137 COST/THOUSAND
```

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 "A" CLEAR
20 INPUT"AD COST = $";A
30 INPUT "UNITS SOLD = ";U
40 C=A/U
50 PRINT"AD COST $";C;" PER UNIT"
60 GOTO 10
```

Sample Run

```
DEF MODE
      SHIFT A

      AD COST = $
      330 ENTER
      UNITS SOLD =
      77 ENTER
      AD COST $4.28 PER UNIT
```

Unit Cost

You know you've just printed 200 newsletters. Your costs came to \$11.50. But how much did it cost you *per each item* among the 200? The answer from this program: 5.75¢.

The first program here compares the quantity of items you manufacture with the total cost of manufacturing those items and gives you the *unit cost* per item, in dollars.

Program Listing

```
10 "K" CLEAR
20 INPUT"QUANTITY MFGD = ";M
30 INPUT"TOTAL MFG COST = $";C
40 T=C/M
50 PRINT"EACH COST $";T
60 GOTO 10
```

The second program is a modification of the first, permitting display of low-unit-cost items in *cents* rather than in dollars. The example is in printing where unit costs often are in cents.

Program Listing

```
10 "K" CLEAR
20 INPUT"QUANTITY PRINTED = ";P
30 INPUT"TOTAL PRINT COST = $";C
40 T=C/P
50 IF T<1 LET T=100T:GOTO 80
60 PRINT"EACH COST $";T
70 GOTO 10
80 PRINT USING"###.##";"EACH COST "
   ;T;" CENTS"
90 GOTO 10
```

Sample Run

```
DEF MODE
      SHIFT K

      QUANTITY PRINTED =
      500 ENTER
      TOTAL PRINT COST = $
      35 ENTER
      EACH COST 7.00 CENTS
      ENTER
      QUANTITY PRINTED =
      200 ENTER
      TOTAL PRINT COST = $
      11.50 ENTER
      EACH COST 5.75 CENTS
```

Mark-Up: Standard Percentage

In pricing goods for retail sale in your store, you like to take a standard 40 percent mark-up, rounded up to the nearest 95¢. This program does that work for you.

Program Listing

```
10 "J" CLEAR
20 BEEP 1:INPUT"WHOLESALE PRICE = $";W
30 BEEP 1:INPUT"MARK-UP % = ";M
40 P=INT (W/(1-.01M))+.95
50 BEEP 2:PRINT"RETAIL PRICE = $";P
60 GOTO 10
```

Sample Run

```
DEF MODE          SHIFT J

BEEP              WHOLESALE PRICE = $
                  3.67 ENTER

BEEP              MARK-UP % =
                  40 ENTER

BEEPBEEP         RETAIL PRICE = $6.95
```

Mark-Up: Manufacturing to Retail

Your manufacturing firm, selling products directly to the public, calculates retail price by using a standard multiplier times manufacturing cost. You use such a standard because exact manufacturing/production costs are too complex to calculate exactly for every product.

Suppose your firm uses 10x as the multiplier. That is, it multiplies manufacturing cost by ten to set retail price.

If one product has a manufacturing cost of 86¢, the formula would indicate a retail price of \$8.60.

Many firms round the intermediate price (\$8.60 in this example) up to the next 95¢. You decide to do just that. Thus, the item which cost 86¢ to manufacture will carry a retail price of \$8.95.

This program, which multiplies and rounds up to 95¢, is equally useful in calculating wholesale or retail prices.

Program Listing

```
10 "H" CLEAR
20 BEEP 1:INPUT"MFG COST = $";C
30 BEEP 1:INPUT"PRICING MULTIPLIER = ";M
40 P=INT(CM)+.95
50 BEEP 2:PRINT"RETAIL PRICE = $";P
60 GOTO 10
```

Sample Run

```
DEF MODE
      SHIFT H
BEEP      MFG COST = $
          1.12 ENTER
BEEP      PRICING MULTIPLIER =
          10 ENTER
BEEPBEEP  RETAIL PRICE =$11.95
          ENTER
BEEP      MFG COST = $
          .79 ENTER
BEEP      PRICING MULTIPLIER =
          8 ENTER
BEEPBEEP  RETAIL PRICE =$6.95
```


Mark-Up: Wholesale to Retail

You know the discount your supplier gives you. You have his invoice showing the wholesale price you paid him. This program combines the wholesale discount you received with the wholesale price you paid to recreate the retail price, or "list" price.

Program Listing

```
10 "G" CLEAR
20 BEEP 1:INPUT"WHOLESALE DISCOUNT % = ";D
30 BEEP 1:INPUT"WHOLESALE PRICE = $";W
40 D=.01D:P=W/(1-D)
50 BEEP 2:PRINT"RETAIL PRICE = $";P
60 GOTO 10
```

Sample Run

```
DEF MODE          SHIFT G

BEEP              WHOLESALE DISCOUNT %=
                  40 ENTER

BEEP              WHOLESALE PRICE = $
                  3.57 ENTER

BEEPBEEP         RETAIL PRICE = $5.95
```

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 "C" CLEAR
20 BEEP 1:INPUT"SALE AMOUNT CENTS = ";X
30 BEEP 1:INPUT"CENTS TENDERED = ";T
40 A=T-X
50 IF A<25 THEN 80
60 Q=Q+1:A=A-25
70 GOTO 50
80 IF A<10 THEN 120
90 D=D+1:A=A-10
100 GOTO 80
120 IF A<5 THEN 150
130 N=N+1:A=A-5
140 GOTO 120
150 P=A
200 BEEP 2
    :PRINT USING "###";Q;" Q. ";D;" D. "
    ;N;" N. ";P;" P."
210 GOTO 10
```

Sample Run

```
DEF MODE          SHIFT C

BEEP              SALE AMOUNT CENTS =

                  51 ENTER

BEEP              CENTS TENDERED =

                  75 ENTER

BEEPBEEP         ØQ. 2D. ØN. 4P.

                  ENTER
```

BEEP SALE AMOUNT CENTS =
 6 ENTER

BEEP CENTS TENDERED =
 100 ENTER

BEEPBEEP 3Q. 1D. 1N. 4P.

Invoice Totaling

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 spits out the various totals you need to plug into your invoice. It will give you total retail price for all goods sold on the invoice, total sales tax, total shipping charges and, finally, the total amount due you from your customer.

Program line 40 directs the computer to ask you for the sales tax percentage rate in your state. Key in that sales tax rate if you wish to collect the tax with this invoice. If you don't need to collect sales tax, key in a zero when the computer asks for the percentage.

Line 50 asks for shipping charges. When running the program, these can be keyed in as a total amount, or as a multiplication of units-times-cost. For instance, you might key in your answer as $.21 * 20$ or you might key it in simply as 4.2.

Program Listing

```
10 "M" CLEAR
20 BEEP 1:INPUT"QUANTITY SOLD = ";Q
30 BEEP 1:INPUT"UNIT PRICE = $";P
40 BEEP 1:INPUT"SALES TAX % =";S
50 BEEP 1:INPUT"SHIPPING = $";H
60 S=.01S:C=QP:T=CS:F=C+T+H
70 BEEP 2:PRINT"TOTAL PRICE = $";C
```

```

80 BEEP 2:PRINT"SALES TAX = $";T
90 BEEP 2:PRINT"SHIPPING = $";H
100 BEEP 2:PRINT"INVOICE TOTAL = $";F
110 GOTO 10

```

Sample Run

```

DEF MODE
      SHIFT M

BEEP      QUANTITY SOLD =
          20
          ENTER
BEEP      UNIT PRICE $=
          5.95
          ENTER
BEEP      SALES TAX % =
          5
          ENTER
BEEP      SHIPPING = $
          .21*20
          ENTER
BEEPBEEP  TOTAL PRICE = $119

BEEPBEEP  SALES TAX = $5.95
BEEPBEEP  SHIPPING = $4.20
BEEPBEEP  INVOICE TOTAL = $129.15

```

Salesman's Commissions

Here's the no-sweat way to compute *commissions* to be paid to your salesmen or reps, all of whom are at the same percentage rate.

If you want to change the percentage in mid stream, as you are running this program, key in SHIFT S rather than ENTER after your last calculation.

Program Listing

```

10 "S" CLEAR
20 BEEP 1
   :INPUT"COMMISSION RATE % = ";R

```

```

30 R=.01R
40 BEEP 1
   ?INPUT"COMMISSION SALES = $";S
50 C=RS
60 BEEP 2:PRINT"COMMISSION = $";C
70 GOTO 40

```

Sample Run

```

DEF MODE
      SHIFT S

BEEP      COMMISSION RATE % =
          8   ENTER

BEEP      COMMISSION SALES = $
          2751.85   ENTER

BEEP      COMMISSION = $220.148
          ENTER

BEEP      COMMISSION SALES = $
          946.30   ENTER

BEEPBEEP  COMMISSION = $75.704
          SHIFT S

BEEP      COMMISSION RATE % =
          15 ENTER

BEEP      COMMISSION SALES = $
          1126.50 ENTER

BEEPBEEP  COMMISSION = $168.975
          ENTER

BEEP      COMMISSION SALES = $
          3108   ENTER

BEEPBEEP  COMMISSION = $466.20

```

Rounding Cents to Nearest Penny

This program is especially useful when used in conjunction with others which generate dollars-and-cents answers to more than two decimal places. Such as \$151.6972.

It would make a good subroutine in a larger program. To use it as a subroutine, insert a GOSUB statement at the appropriate place in your set of instructions. Modify the line numbers so the subroutine will be located in an unused position in your listing. Change the last line of the subroutine (GOTO 10) to a RETURN statement. Delete the first line ("M" CLEAR).

Program Listing

```
10 "M" CLEAR
20 BEEP 1:INPUT"ORIG. AMOUNT = $";N
30 R=INT(100N+.5)/100
40 BEEP 2:PRINT "$";R
50 GOTO 10
```

Sample Run

```
DEF MODE
SHIFT M
BEEP      ORIG. AMOUNT = $
          151.6972  ENTER
BEEPBEEP $151.70
```

YES/NO Executive Decision Maker

Stumped by a toughie? Got one too hot to handle alone? Need help with major decisions? When there is no other way to decide, punch up this executive decision maker and get the answer: YES or NO. In 200 sample runs we produced 107 YES and 93 NO answers.

Program Listing

```
10 "B" N=N+1:A=NN:Y=(997*X+A)/199
   :X=Y-INT Y:X=X*E2
20 IF X>50 PRINT" YES":GOTO"B"
30 IF X<51 PRINT" NO":GOTO"B"
```

Adding A Column Of Numbers Using One-Key Entry

The object is to minimize the number of keystrokes. Key in as many numbers as you like, in any order. When you have completed your list of numbers, type in a zero. The computer will add all of the numbers and display the total.

Program Listing

```
700 "J" CLEAR
710 INPUT A
720 B=B+A
730 IF A<>0 THEN 710
740 PRINT B
750 GOTO 700
```

Sample Run

```
26
49
18
07
34
55
0
—
189
```

Grade Scoring

Ideal for a teacher with lots of tests to evaluate, this program permits keying in up to 999 different scores in any order.

To conclude entering a list of scores, the letter "X" is entered after the last score. The program then will display the total of all scores entered; range of scores; quantity of scores in each grade range; mean (average) score; and median score.

Test scores must be in the range of zero to 100 for use by the program as it is written. The range could be modified.

The computer assigns the letter grade of A for scores ranging from 90-100; B from 80-89; C from 70-79; D from 60-69; and F to scores below 60.

A great time saver!

Program Listing

```
10 "G" CLEAR
20 PAUSE"ENTER SCORES"
30 X=999
40 FOR E=1 TO 999
50 INPUT"SCORE = ";G
60 IF G=X THEN 180
70 N=N+1
80 IF G<60 LET F=F+1:GOTO 130
90 IF G<70 LET D=D+1:GOTO 130
```



```

100 IF G<80 LET C=C+1:GOTO 130
110 IF G<90 LET B=B+1:GOTO 130
120 A=A+1
130 IF N=1 LET L=G:H=G
140 IF G<L LET L=G
150 IF G>H LET H=G
160 S=S+G
170 NEXT E
180 P=S/N:M=L+((H-L)/2)
190 PRINT"SCORES RANGE ";L;" TO ";H
200 PRINT"MEDIAN SCORE =" ;M
210 PRINT"AVERAGE (MEAN) =" ;P
220 PAUSE"TOTALS OF EACH GRADE"
230 PRINT A;" A"
240 PRINT B;" B"
250 PRINT C;" C"
260 PRINT D;" D"
270 PRINT P;" P"
280 PRINT"TOTAL ";N;" SCORES"
290 GOTO 10

```

Sample Run

```

DEF MODE
      SHIFT G

      ENTER SCORES
      SCORE =

      87 ENTER

      SCORE =

      55 ENTER

      SCORE =

      91 ENTER

      SCORE =

      82 ENTER

```

SCORE=

79 ENTER

SCORE =

75 ENTER

SCORE =

63 ENTER

SCORE =

X ENTER

SCORES RANGE 55 TO 91

MEDIAN SCORE =73

AVERAGE (MEAN) =76

TOTALS OF EACH GRADE

1 A

2 B

2 C

1 D

1 F

TOTAL 7 SCORES

Slope and Distance

This program computes the slope of a line between two points on a graph and the distance between those two points. The letter X designates the horizontal position on the graph and Y the vertical position.

Program Listing

```
10 "S" CLEAR
20 BEEP 1:INPUT"X1= ";A
30 BEEP 1:INPUT"Y1= ";B
40 BEEP 1:INPUT"X2= ";C
```

```

50 BEEP 1:INPUT"Y2= ";D
60 IF A=C THEN 80
70 S=(D-B)/(C-A)
80 E=(ABS (C-A)2 + ABS (D-B)2).5
90 BEEP 2:PRINT"SLOPE = ";S
100 BEEP 2:PRINT"DISTANCE = ";E
110 GOTO 10

```

Sample Run

```

DEF MODE
BEEP          SHIFT S
              X1=
              5 ENTER
BEEP          Y1=
              8 ENTER
BEEP          X2=
              11 ENTER
BEEP          Y2=
              14 ENTER
BEEPBEEP     SLOPE=1
              ENTER
BEEPBEEP     DISTANCE = 8.485281374
              ENTER
BEEP          X1=
              14 ENTER
BEEP          Y1=
              26 ENTER
BEEP          X2=

```

```

5 ENTER
BEEP Y2=
40 ENTER
BEEPBEOP SLOPE = -1.555555556
ENTER
BEEPBEOP DISTANCE = 16.64331698

```

dB

Here's a handy quickie for engineers, radio technicians, electronics buffs, hams, shortwave listeners and others who know how many bels there are in a decibel.

Program Listing

```

10 "DB" CLEAR
20 BEEP 1:INPUT"INPUT POWER = ";I
30 BEEP 1:INPUT"OUTPUT POWER = ";O
40 D=10*LOG(O+I)
50 BEEP 2:PRINT D;" DB"
60 GOTO 10

```

Sample Run

```

RUN MODE RUN"DB "
ENTER ENTER
BEEP INPUT POWER =
100
ENTER
BEEP OUTPUT POWER=
200
ENTER
BEEPBEOP 3.010299957 DB

```

Flash Cards Math Drill

Here's a fun way to practice your math, or drill your kids on adding, subtracting, multiplying and dividing. The computer asks word questions which you answer YES or NO *yes* or *no*. Then it presents appropriate math problems for you to demonstrate your skills.

Note: this is a long program to key in, leaving only 81 steps/10 memories when fully loaded. However, it is a pleasure to operate with its human-like conversation.

Program Listing

```
10 "A" CLEAR
20 FOR X=1 TO 3:PAUSE" * FLASH CARDS *"
  :NEXT X
30 A$="ADD":B$="SUBTRACT"
  :C$="MULTIPLY":D$="DIVIDE"
40 PAUSE"":PAUSE" SHALL WE...
50 PAUSE" ADD":PAUSE" SUBTRACT"
  :PAUSE" MULTIPLY":PAUSE" DIVIDE"
60 BEEP 1:INPUT" WHICH ?",E$
70 IF E$=A$ THEN "ADD"
80 IF E$=B$ THEN "SUBTRAC"
90 IF E$=C$ THEN "MULTIPL"
100 IF E$=D$ THEN "DIVIDE"
110 GOTO 50
200 "ADD" PAUSE" OKAY"
210 FOR X=6 TO 7
220 S=((PI+S)A5)-INT((PI+S)A5)
  :S=10*S:R=INT S
222 IF R>10 THEN 220
224 IF R<1 THEN 220
230 A(X)=R
240 NEXT X
250 PAUSE USING "###";F;" +";G
260 BEEP 1:INPUT H
270 IF H=F+G PAUSE" CORRECT":J=J+1
  :GOTO 285
280 PAUSE" WRONG":PAUSE" TRY AGAIN"
  :K=K+1:GOTO 250
```

```

285 BEEP 1:INPUT" MORE ?",L$
290 IF L$="YES" THEN "ADD"
295 GOTO 600
300 "SUBTRAC" PAUSE" OKAY"
310 FOR X=6 TO 7
320 S=((π+S)A5)-INT((π+S)A5):S=10*S
   :R=INTS
322 IF R>10 THEN 320
324 IF R<1 THEN 320
330 A(X)=R
340 NEXT X
350 PAUSE USING "###";F;" -";G
360 BEEP 1:INPUT H
370 IF H=F-G PAUSE" CORRECT":J=J+1
   :GOTO 385
380 PAUSE" WRONG":PAUSE" TRY AGAIN"
   :K=K+1:GOTO 350
385 BEEP 1:INPUT" MORE ?",L$
390 IF L$="YES" THEN "SUBTRAC"
395 GOTO 600
400 "MULTIPL" PAUSE" OKAY"
410 FOR X=6 TO 7
420 S=((π+S)A5)-INT((π+S)A5):S=10*S
   :R=INT S
422 IF R>10 THEN 420
424 IF R<1 THEN 420
430 A(X)=R
440 NEXT X
450 PAUSE USING "###";F;" X";G
460 BEEP 1:INPUT H
470 IF H=F*G PAUSE" CORRECT":J=J+1
   :GOTO 485
480 PAUSE" WRONG":PAUSE" TRY AGAIN"
   :K=K+1:GOTO 450
485 BEEP 1:INPUT" MORE ?",L$
490 IF L$="YES" THEN "MULTIPL"
495 GOTO 600
500 "DIVIDE" PAUSE" OKAY"
510 FOR X= 6TO 7
520 S=((π+S)A5)-INT((π+S)A5):S=10*S
   :R=INT S

```

```

522 IF R>10 THEN 520
524 IF R<1 THEN 520
530 A(X)=R
540 NEXT X
545 Q=FG
550 PAUSE USING "###";Q;" /";G
560 BEEP 1:INPUT H
570 IF H=Q/G PAUSE" CORRECT":J=J+1
      :GOTO 585
580 PAUSE" WRONG":PAUSE" TRY AGAIN"
      :K=K+1:GOTO 550
585 BEEP 1:INPUT" MORE ?",L$
590 IF L$="YES" THEN "DIVIDE"
600 PAUSE"":BEEP 1
      :INPUT" TRY SOMETHING ELSE ?",I$
610 IF I$="YES" THEN 50
620 PAUSE" OKAY":PAUSE" YOUR SCORE..."
630 PRINT J;" CORRECT ";K;" WRONG"
640 GOTO 10

```

Sample Run

```

DEF MODE
      SHIFT A
          * FLASH CARDS *
          * FLASH CARDS *
          * FLASH CARDS *
          SHALL WE...
          ADD
          SUBTRACT
          MULTIPLY
          DIVIDE
BEEP   WHICH?
          DIVIDE ENTER
          OKAY
          12/2
BEEP   ?
          6 ENTER

```

BEEP CORRECT
 MORE?

 YES ENTER

 OKAY
 9/9
BEEP ?

 1 ENTER

BEEP CORRECT
 MORE?

 NO ENTER

BEEP TRY SOMETHING ELSE?

 YES ENTER

 ADD
 SUBTRACT
 MULTIPLY
 DIVIDE
BEEP WHICH?

 MULTIPLY ENTER

 OKAY
 3 X 7
BEEP ?

 14 ENTER

 WRONG
 TRY AGAIN
 3 X 7
BEEP ?

 21 ENTER
BEEP CORRECT
 MORE?

NO ENTER

BEEP TRY SOMETHING ELSE?
NO ENTER

OKAY
YOUR SCORE...
3 CORRECT 1 WRONG

MEMORY REMAINING: 81 STEPS 10 MEMORIES

Smart Computer

This game is so quick you can work it in while your secretary is away sharpening her pencil. It's more or less the reverse of the old favorite High-Low Number bit. In this rendition, you come up with the secret three-digit number and the computer guesses it!

In old-fashioned High-Low, the computer could keep you guessing for hours. Here, the computer asks one little old question and...*bingo!* It has the correct answer in an instant.

By the way, this program can make you look like the company genius at the next office party. Ask your friends to pick a secret number and ask them the one question the computer asks you. You'll get the right answer from your Pocket Computer in a flash.

The mystery number has to be a three-digit number with all digits the same.

Program Listing

```
10 "X" CLEAR
20 PAUSE"PICK A 3-DIGIT NUMBER"
30 PAUSE"WITH ALL DIGITS SAME"
40 PAUSE"ADD THE 3 DIGITS"
50 BEEP 1
   :INPUT"SUM OF THE 3 DIGITS ?",N
60 Q=37N
70 PRINT"YOUR NUMBER IS ";Q
80 GOTO 10
```

Sample Run

```
DEF MODE
      SHIFT X

      PICK A 3-DIGIT NUMBER
      WITH ALL DIGITS SAME
      ADD THE 3 DIGITS
BEEP  SUM OF THE 3 DIGITS ?

      24 ENTER

      YOUR NUMBER IS 888
```

Frequency and Wavelength

Engineering student. Radio technician. Ham operator, CBer, or shortwave listener. You hate to dig out a pencil and paper just to convert frequency to wavelength. Or wavelength to frequency. This program does it in an instant.

Follow it up with our *Metric Converter* program and you'll even know how many "inches" of wavelength in your favorite frequency.

Program Listing

```
10 "F" CLEAR
20 PAUSE"FREQ OR WAVELENGTH"
30 INPUT"WANT F OR W ?",A$
40 IF A$="F" THEN 90
50 INPUT"FREQUENCY (MHZ) = ";F
60 W=300/F
70 PRINT F;" MHZ = ";W;" METERS"
80 GOTO 10
90 INPUT"WAVELENGTH (METERS) = ";W
100 F=300/W
110 PRINT W;" METERS = ";F;" MHZ"
120 GOTO 10
```

Sample Run

DEF MODE

SHIFT F

FREQ OR WAVELENGTH

WANT F OR W?

W ENTER

FREQUENCY(MHZ) =

7.5 ENTER

7.5MHZ = 40 METERS

FREQ OR WAVELENGTH

WANT F OR W?

F ENTER

WAVELENGTH (METERS) =

160 ENTER

160 METERS = 1.875 MHZ

Metric Converter

Keep this math conversion program loaded in your Pocket Computer and you'll always have a handy way to quickly change feet, inches or yards into millimeters, centimeters or meters. Or vice versa.

Program Listing

```
10 "=" CLEAR
20 PAUSE"METRIC CONVERSION"
30 INPUT"TO METRIC (Y OR N) ?",A$
40 IF A$="N" THEN "METERS"
50 INPUT"IN, FT OR YDS ?",B$
60 IF B$="IN" THEN "INCHES"
70 IF B$="FT" THEN "FEET"
80 IF B$="YDS" THEN "YARDS"
90 GOTO 30
100 "INCHES" INPUT"INCHES = ";I
```

```

110 M=I/39.37:C=100M:N=1000M
120 PRINT I;"IN = ";N;"MM"
130 PRINT I;"IN = ";C;"CM"
140 PRINT I;"IN = ";M;"MTRS"
150 GOTO 30
200 "FEET" INPUT"FEET = ";F
210 I=12F:M=I/39.37:C=100M:N=1000M
220 PRINT F;"FT = ";N;"MM"
230 PRINT F;"FT = ";C;"CM"
240 PRINT F;"FT = ";M;"MTRS"
250 GOTO 30
300 "YARDS" INPUT"YARDS = ";Y
310 I=36Y:M=I/39.37:C=100M:N=1000M
320 PRINT Y;"YDS = ";N;"MM"
330 PRINT Y;"YDS = ";C;"CM"
340 PRINT Y;"YDS = ";M;"MTRS"
350 GOTO 30
400 "METERS"
410 INPUT"CM, MM OR MTRS ?",D$
420 IF D$="CM" THEN "CM"
430 IF D$="MM" THEN "MM"
440 IF D$="MTRS" THEN "MTRS"
450 GOTO 30
500 "CM" INPUT"CENTIMETERS = ";C
510 I=.3937C:F=I/12:Y=I/36
520 PRINT C;"CM = ";I;"IN"
530 PRINT C;"CM = ";F;"FT"
540 PRINT C;"CM = ";Y;"YDS"
550 GOTO 30
600 "MM" INPUT"MILLIMETERS = ";N
610 I=.03937N:F=I/12:Y=I/36
620 PRINT N;"MM = ";I;"IN"
630 PRINT N;"MM = ";F;"FT"
640 PRINT N;"MM = ";Y;"YDS"
650 GOTO 30
700 "MTRS" INPUT"METERS = ";M
710 I=39.37M:F=I/12:Y=I/36
720 PRINT M;"MTRS = ";I;"IN"
730 PRINT M;"MTRS = ";F;"FT"
740 PRINT M;"MTRS = ";Y;"YDS"
750 GOTO 30

```

Sample Run

DEF MODE

SHIFT =
METRIC CONVERSION
TO METRIC (Y OR N)?
N ENTER

CM, MM OR MTRS?
CM ENTER

CENTIMETERS =
225 ENTER

225 CM=88.5825 IN

ENTER

225CM = 7.381875 FT

ENTER

225 CM = 2.460625 YDS
ENTER

TO METRIC (Y OR N)?
Y ENTER

IN, FT OR YDS?
FT ENTER

FEET =
49 ENTER

49 FT = 14935.22987 MM
ENTER

49 FT = 1493.522987 CM

ENTER

49 FT = 14.93522987 MTRS

Factors

This program lists the factors of any number you specify up to 129. It could be used as a subroutine in a larger program.

Program Listing

```
10 "M" CLEAR
20 BEEP 1:INPUT" NUMBER = ";A(130)
30 IF A(130)<3 GOTO 10
40 FOR A(131)=2 TO (A(130)/2)
50 PAUSE" CHECKING ";A(131)
   :A(132)=A(130)/A(131)
60 IF A(132)=INT A(132) LET A(133)=A(133)+1
   :A(A(133))=A(132)
   :PAUSE A(131);" IS A FACTOR"
70 NEXT A(131)
80 BEEP 2:PAUSE" RECAPING..."
90 FOR A(134)=1 TO (A(130)/2)
100 IF A(A(134))>1 BEEP 1
    :PAUSE A(A(134));" IS A FACTOR"
110 NEXT A(134)
120 GOTO 10
```

Sample Run

```
DEF MODE
      SHIFT M
BEEP  NUMBER =
      18 ENTER
      CHECKING 2
      2 IS A FACTOR
      CHECKING 3
      3 IS A FACTOR
      CHECKING 4
      CHECKING 5
      CHECKING 6
      6 IS A FACTOR
```

CHECKING 7
CHECKING 8
CHECKING 9
9 IS A FACTOR

BEEP BEEP RECAPPING...

BEEP 9 IS A FACTOR

BEEP 6 IS A FACTOR

BEEP 3 IS A FACTOR

BEEP 2 IS A FACTOR

Ohm's Law

The basis of all electronics and electricity. E is volts. I is amps. R is ohms. If you know any two of the values, the computer will give you the third.

Program Listing

```
10 "L" CLEAR
20 PAUSE"ENTER KNOWN DATA"
30 BEEP 1:INPUT"CURRENT (AMPS) = ";I
40 BEEP 1:INPUT"RESISTANCE(OHMS)=";R
50 BEEP 1:INPUT"VOLTAGE = ";E
200 IF I THEN 300
210 I=E/R
220 BEEP 2:PRINT"CURRENT = ";I;" AMPS"
230 GOTO 10
300 IF E THEN 400
310 E=IR
320 BEEP 2:PRINT"VOLTS = ";E
330 GOTO 10
400 IF R THEN 10
410 R=E/I
420 BEEP 2:PRINT"RESISTANCE = ";R;" OHMS"
430 GOTO 10
```

Sample Run

DEF MODE

SHIFT L

BEEP

ENTER KNOWN DATA
CURRENT (AMPS)=

.500
ENTER

BEEP

RESISTANCE (OHMS)=

ENTER

BEEP

VOLTAGE =

12
ENTER

BEEPBEEP

RESISTANCE = 24 OHMS

ENTER

BEEP

ENTER KNOWN DATA
CURRENT (AMPS)=

ENTER

BEEP

RESISTANCE (OHMS)=

1000
ENTER

BEEP

VOLTAGE =

24.6
ENTER

BEEPBEEP

CURRENT =0.0246 AMPS

Reciprocals

Key in a number and the computer displays its reciprocal.

Program Listing

```
10 "L" CLEAR
20 BEEP 1:INPUT" NUMBER TO CONVERT",N
30 R=1/N
40 BEEP 2:PRINT"RECIP = ";R
50 GOTO 20
```

Sample Run

```
DEF MODE
                                SHIFT L

BEEP                            NUMBER TO CONVERT
                                5 ENTER

BEEPBEEP                       RECIP = 0.2
                                ENTER

BEEP                            NUMBER TO CONVERT
                                32 ENTER

BEEPBEEP                       RECIP = 0.03125
```

Memory Tester

Suppose I pick a number out of the air. Say, 5. You can remember that, right? Or 73. Or 841. But, just how big a number can you see briefly and remember? This program throws ever-increasingly-larger random numbers at you and asks you to remember them. And it keeps score. What's your memory's upper limit?

Program Listing

```
10 "N" CLEAR
20 Z=1
30 BEEP 1:INPUT"ANY NUMBER = ";S
40 A=SS:Y=(997*X+A)/199
   :X=Y-INT Y:X=10X:S=INT X
50 IF S=Ø THEN 40
60 N=INT (SZ)
70 PAUSE"REMEMBER ";N
80 FOR L=1 TO 25:NEXT L
90 IF W=3 PAUSE" FORGET IT !"
   :PAUSE" START OVER":GOTO 10
100 BEEP 1:INPUT"WHAT WAS THE NUMBER ?",S
110 IF S<>N PAUSE" WRONG !"
   :W=W+1:GOTO 90
120 PAUSE" RIGHT !":R=R+1:W=Ø:Z=11Z
130 PAUSE USING"###";R;" RIGHT SO FAR"
   :USING:GOTO 40
```

Sample Run

```
DEF MODE
      SHIFT N
BEEP  ANY NUMBER =
      44 ENTER

      REMEMBER 7289
      (20 SECOND PAUSE)
      WHAT WAS THE NUMBER ?

      7280 ENTER

      WRONG !
BEEP  WHAT WAS THE NUMBER ?

      7289 ENTER

      RIGHT !
      1 RIGHT SO FAR
```

Pythagorean Theorem

The theorem lets you find the length of any side of a right triangle if you know the length of the other two sides. The two shorter sides of the triangle are labeled A and B. The longer side, the hypotenuse, is C. The length of C is always longer than A or B.

Program Listing

```
10 "H" CLEAR
20 BEEP 1:INPUT"SIDE A LENGTH = ";A
30 BEEP 1:INPUT"SIDE B LENGTH = ";B
40 IF A THEN 60
50 GOTO 70
60 IF B THEN 150
70 BEEP 1:INPUT"HYPOTENUSE (C) = ";C
90 IF A THEN 120
100  $A=\sqrt{CA^2-BA^2}$ 
110 GOTO 170
120 IF B THEN 150
130  $B=\sqrt{CA^2-AA^2}$ 
140 GOTO 190
150  $C=\sqrt{AA^2+BA^2}$ 
160 GOTO 210
170 BEEP 2:PRINT"SIDE A = ";A
180 GOTO 10
190 BEEP 2:PRINT"SIDE B = ";B
200 GOTO 10
210 BEEP 2:PRINT"HYPOTENUSE = ";C
220 GOTO 10
```

Sample Run

```
DEF MODE
SHIFT H
BEEP SIDE A LENGTH =
16 ENTER
BEEP SIDE B LENGTH =
12 ENTER
```

```

BEEP BEEP      HYPOTENUSE = 20
                ENTER
BEEP           SIDE A LENGTH =
                ENTER
BEEP           SIDE B LENGTH =
                25.3 ENTER
BEEP           HYPOTENUSE (C) =
                36.9 ENTER
BEEP BEEP      SIDE A = 26.86112432

```

Averages

Now you can average a series of up to 150 numbers. Just key them in, in any order. After the last number in the series is keyed in, press ENTER or \emptyset ENTER to complete the computation. Obviously, zero can't be one of the numbers in the series you are averaging.

Program Listing

```

10 "X" CLEAR
20 FOR A=2 TO 151
30 BEEP 1:INPUT"NUMBER = ";A(A)
40 IF A(A)= $\emptyset$  THEN 70
50 A(153)=A(153)+A(A)
60 NEXT A
70 A=A-2
80 A(152)=A(153)/A
90 BEEP 2:PRINT"AVERAGE = ";A(152)
100 GOTO 10

```

Sample Run

```

DEF MODE      SHIFT X

BEEP         NUMBER =

              11
              ENTER

```

BEEP NUMBER =
 22
 ENTER

BEEP NUMBER =
 33
 ENTER

BEEP NUMBER =
 44
 ENTER

BEEP NUMBER =
 55
 ENTER

BEEP NUMBER =
 ENTER

BEEPBEEP AVERAGE = 33
 ENTER

BEEP NUMBER =
 567
 ENTER

BEEP NUMBER =
 234
 ENTER

BEEP NUMBER =
 ENTER

BEEPBEEP AVERAGE = 400.5

Temperature Conversion (F, C, K)

F is Fahrenheit. C is Celsius. K is Kelvin. This program transforms temperature readings in one system to their equivalent values in the other two.

For an easier-to-read, less precise temperature readout, make these changes to program lines 140, 240 and 340:

140 BEEP 2: PRINT INT F: " DEGREES F"

240 BEEP 2: PRINT INT C: " DEGREES C"

340 BEEP 2: PRINT INT K: " DEGREES K"

Then the sample would be the same as above except the numbers of degrees will be displayed without decimals.

Program Listing

```
10 "X" CLEAR
20 BEEP 1: INPUT "PRESENT FAHRENHEIT = "; F
30 IF F THEN 200
40 BEEP 1: INPUT "PRESENT CELSIUS = "; C
50 IF C THEN 100
60 BEEP 1: INPUT "PRESENT KELVIN = "; K
100 IF F THEN 200
110 IF K THEN 130
120 F=1.8C+32:GOTO 140
130 F=1.8K-459.67
140 BEEP 2: PRINT F; " DEGREES F"
200 IF C THEN 300
210 IF K THEN 230
220 C=(F-32)/1.8:GOTO 240
230 C=K-273.15
240 BEEP 2: PRINT C; " DEGREES C"
300 IF K THEN 10
310 IF F THEN 330
320 K=C+273.15:GOTO 340
330 K=(F+459.67)/1.8
340 BEEP 2: PRINT K; " DEGREES K"
350 GOTO 10
```

Sample Run

```
DEF MODE                SHIFT X

BEEP                    PRESENT FAHRENHEIT =
                        72
                        ENTER

BEEPBEEP               22.22222222 DEGREES C
BEEPBEEP               295.3722222 DEGREES K
```

Rounding Off Numbers

You often don't need a number answer from the computer with a fractional part displayed as decimal. Rounding off to the nearest whole number is fine. 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 .5 rounds down. Another view is that "any number less than five rounds down," in which case exactly .5 rounds up!

The first set of line listings below is for the fellow with the "more than five rounds up" philosophy.

Program Listing

```
10 "A" CLEAR
20 BEEP 1
   :INPUT" NUMBER TO BE ROUNDED",N
30 IF N>INT N THEN 50
40 R=N:GOTO 100
50 D=N-INT N
60 IF D>.5 THEN 90
70 R =INT N
80 GOTO 100
90 R=INT N +1
100 BEEP 1:PAUSE N;" ROUNDS TO..."
```

```
110 PRINT R
120 GOTO 10
```

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

Program Listing

```
10 "B" CLEAR
20 BEEP 1
   :INPUT" NUMBER TO BE ROUNDED",N
30 IF N>INT N THEN 50
40 R=N:GOTO 100
50 D=N-INT N
60 IF D<.5 THEN 90
70 R=INT N + 1
80 GOTO 100
90 R=INT N
100 BEEP 1:PAUSE N;" ROUNDS TO..."
110 PRINT R
120 GOTO 10
```

Sample Run

```
DEF MODE
      SHIFT A

BEEP   NUMBER TO BE ROUNDED

      123.5678
      ENTER

BEEP   123.5678 ROUNDS TO...
      124

      ENTER

BEEP   NUMBER TO BE ROUNDED

      765.4321
      ENTER

      765.4321 ROUNDS TO...
      765
```


Fractional Portion of a Number

Most computers make it easy to obtain the integer value of a number. It's a bit harder to get rid of the integer and keep only the decimal value, to the right of the decimal point in the original number. This program gets rid of the integer portion of the number and displays the decimal part. It also would make a good subroutine in a larger program.

Program Listing

```
10 "D" CLEAR
20 BEEP 1:INPUT"ORIGINAL NUMBER =",N
30 F=N - INT N
40 BEEP 2:PRINT F
50 GOTO 20
```

Sample Run

```
DEF MODE
                                SHIFT D

BEEP                            ORIGINAL NUMBER =
                                12.34567  ENTER

BEEPBEEP                        .34567
                                ENTER

BEEP                            ORIGINAL NUMBER =
                                567.890123 ENTER

BEEPBEEP                        .890123
```

Which Number Is Largest?

This program will sort through 10 numbers and tell you which number is the largest. The numbers can be keyed into the computer in any order. If you want to handle more than 10 numbers, change the loop in line 20 by increasing the last number in that line.

This program also would be a good subroutine if needed in a larger program. To convert it from a stand-alone program to a subroutine, delete line 10 and change line 130 to a RETURN statement. The line numbers for the entire subroutine will have to be changed to fit available space in your master program. And variables' names may have to be changed.

Program Listing

```
10 "A" CLEAR
20 FOR Z=1 TO 10
30 BEEP 1:INPUT"NUMBER = ";A(Z)
40 IF A(Z)=ØTHEN 70
50 N=N+1
60 NEXT Z
70 Y=A
80 FOR X=2 TO N
90 IF A(X) <= A THEN 110
100 Y=A(X)
110 NEXT X
120 BEEP 1:PRINT" LARGEST # = ";Y
130 GOTO 10
```

Sample Run

```
DEF MODE
                                SHIFT A

BEEP                            NUMBER =
                                11 ENTER

BEEP                            NUMBER =
                                22 ENTER
BEEP                            NUMBER =
                                33 ENTER
BEEP                            NUMBER =
                                44 ENTER
BEEP                            NUMBER =
                                ENTER

BEEPBEEP                        LARGEST #=44
```

Deposit Doubler

There's an old rule of thumb used by some bankers that holds that money left on deposit will *double* in value as interest compounds. If you don't take this *Rule of 72* too seriously, it's a handy way to gauge roughly the number of years you need to leave your money in a bank account (or other interest-bearing system) to see it double in value.

The rule holds that the number of years to double is roughly equal to 72 divided by the annual interest rate.

Program Listing

```
10 "D" CLEAR
20 BEEP 1
   :INPUT"ANNUAL INTEREST % = ";I
30 Y=72/I
40 IF Y>INT Y THEN 60
50 GOTO 110
60 D=Y-INT Y
70 IF D>.5 THEN 90
80 Y=INT Y
90 GOTO 110
100 Y=INT Y+1
110 BEEP 2
   :PAUSE"MONEY DOUBLES IN..."
120 PRINT"ABOUT ";Y;" YEARS"
130 GOTO 10
```

Sample Run

DEF MODE

SHIFT D

BEEP ANNUAL INTEREST % =
11 ENTER

BEEPBEEP MONEY DOUBLES IN...
ABOUT 7 YEARS

ENTER

BEEP ANNUAL INTEREST % =
21 ENTER

BEEPBEEP MONEY DOUBLES IN...
ABOUT 3 YEARS

Simple Interest

My savings account has \$400 in it. The bank pays me 7.5 percent interest on the account. If I leave the money in that account for 12 years, how much interest will I have earned after 12 years? What will the new amount total in the account?

Program Listing

```
10 "F" CLEAR
20 BEEP 1:INPUT"PRESENT VALUE = $";P
30 BEEP 1:INPUT"ANNUAL INTEREST % = ";I
40 BEEP 1:INPUT"NUMBER OF YEARS = ";Y
50 I=.01I:X=PIY
60 PAUSE Y;" YEARS INTEREST..."
70 PAUSE "TOTALS $";X
80 A=P+X
90 PRINT"PRINCIPAL = $";A
100 GOTO 10
```

Sample Run

DEF MODE

SHIFT F

```

BEEP    PRESENT VALUE = $
        400 ENTER

BEEP    ANNUAL INTEREST % =
        7.5 ENTER

BEEP    NUMBER OF YEARS
        12 ENTER

        12 YEARS INTEREST...
        TOTALS $360
        PRINCIPAL = $760

```

Compound Interest: Present Value

Using this program, the computer can tell you how much money to deposit in your savings account today, at what interest rate, and how long to leave it in that account to get a specific amount you require in that account at a point in the future.

Program Listing

```

10 "B" CLEAR
20 BEEP 1:INPUT"FUTURE VALUE = $";F
30 BEEP 1:INPUT"ANNUAL INTEREST % = ";I
40 BEEP 1:INPUT"NUMBER YEARS = ";Y
50 I=.01I:P=F/(1+I)^Y
60 PRINT"PRESENT VALUE = $";P
70 GOTO 10

```

Sample Run

```

DEF MODE
        SHIFT B

BEEP    FUTURE VALUE = $

```

```

2000 ENTER

BEEP   ANNUAL INTEREST % =

10 ENTER

BEEP   NUMBER YEARS =

8 ENTER

PRESENT VALUE = $933.01

```

Compound Interest: Future Value (Annual Compounding)

Here the computer uses the present value of your savings account, the annual interest rate and the number of years you plan to leave the money in the account to compute the future value of the account with annual compounding.

Program Listing

```

10 "C" CLEAR
20 BEEP 1:INPUT"PRESENT VALUE = $";P
30 BEEP 1:INPUT"ANNUAL INTEREST % =";I
40 BEEP 1:INPUT"NUMBER OF YEARS = ";Y
50 I=.01I:F=P*(1+I)^Y
60 PRINT"FUTURE VALUE = $";F
70 GOTO 10

```

Sample Run

```

DEF MODE
      SHIFT C

BEEP   PRESENT VALUE = $

400 ENTER

BEEP   ANNUAL INTEREST % =

```

```

7.5 ENTER

BEEP    NUMBER OF YEARS =

12 ENTER

FUTURE VALUE = $952.71

```

Compound Interest: Future Value (Quarterly Compounding)

If you know how much you have in a savings account, how much interest you are earning, and how many years you plan to leave the money in the account, the computer will tell you how many quarters are in those years and compute the future total value of the account.

Program Listing

```

10 "D" CLEAR
20 BEEP 1:INPUT"PRESENT VALUE = $";P
30 BEEP 1:INPUT"ANNUAL INTEREST % = ";I
40 BEEP 1:INPUT"NUMBER OF YEARS = ";Y
50 Q=4Y:I=.01I/4:F=P*(1+I)^Q
60 PAUSE"NUMBER OF QUARTERS = ";Q
70 PRINT"FUTURE VALUE = $";F
80 GOTO 10

```

Sample Run

```

DEF MODE
      SHIFT D

BEEP    PRESENT VALUE = $

400 ENTER

BEEP    ANNUAL INTEREST % =

7.5 ENTER

```

```
BEEP      NUMBER OF YEARS =  
          12 ENTER  
  
          NUMBER OF QUARTERS=48  
          FUTURE VALUE = $975.67
```

Compound Interest: Future Value (Showing Intermediate Years)

Here the computer uses the present value of your savings account, the interest rate, and the number of years you plan to leave the money in the account. It then lists the total amount in the account after each year through the period you plan to maintain the account.

Program Listing

```
10 "A" CLEAR  
20 BEEP 1:INPUT"PRESENT VALUE = $";P  
30 BEEP 1:INPUT"INTEREST RATE %.= ";I  
40 L=I/100+1  
50 BEEP 1:INPUT"NUMBER OF YEARS = ";Y  
60 PAUSE"STARTING WITH $";P;"..."  
70 FOR K = 1 TO Y  
80 P=PL  
90 PAUSE"AFTER ";K;" YEARS..."  
100 PAUSE "PRINCIPAL = $";P  
110 NEXT K  
120 GOTO 20
```

Sample Run

```
DEF MODE  
      SHIFT A  
  
BEEP      PRESENT VALUE = $  
  
          525 ENTER  
  
BEEP      INTEREST RATE %=
```



```

7.5 ENTER
BEEP    NUMBER OF YEARS =
        12 ENTER
        FUTURE VALUE = $952.71

```

Compound Interest: Future Value (Quarterly Compounding)

If you know how much you have in a savings account, how much interest you are earning, and how many years you plan to leave the money in the account, the computer will tell you how many quarters are in those years and compute the future total value of the account.

Program Listing

```

10 "D" CLEAR
20 BEEP 1:INPUT"PRESENT VALUE = $";P
30 BEEP 1:INPUT"ANNUAL INTEREST % = ";I
40 BEEP 1:INPUT"NUMBER OF YEARS = ";Y
50 Q=4Y:I=.01I/4:F=P*(1+I)^Q
60 PAUSE"NUMBER OF QUARTERS = ";Q
70 PRINT"FUTURE VALUE = $";F
80 GOTO 10

```

Sample Run

```

DEF MODE
      SHIFT D

BEEP    PRESENT VALUE = $
        400 ENTER

BEEP    ANNUAL INTEREST % =
        7.5 ENTER

```

```
BEEP      NUMBER OF YEARS =  
          12 ENTER  
  
          NUMBER OF QUARTERS=48  
          FUTURE VALUE = $975.67
```

Compound Interest: Future Value (Showing Intermediate Years)

Here the computer uses the present value of your savings account, the interest rate, and the number of years you plan to leave the money in the account. It then lists the total amount in the account after each year through the period you plan to maintain the account.

Program Listing

```
10 "A" CLEAR  
20 BEEP 1:INPUT"PRESENT VALUE = $";P  
30 BEEP 1:INPUT"INTEREST RATE % = ";I  
40 L=I/100+1  
50 BEEP 1:INPUT"NUMBER OF YEARS = ";Y  
60 PAUSE"STARTING WITH $";P;"..."  
70 FOR K = 1 TO Y  
80 P=PL  
90 PAUSE"AFTER ";K;" YEARS..."  
100 PAUSE "PRINCIPAL = $";P  
110 NEXT K  
120 GOTO 20
```

Sample Run

```
DEF MODE  
      SHIFT A  
  
BEEP      PRESENT VALUE = $  
  
          525 ENTER  
  
BEEP      INTEREST RATE %=
```

```

7.5 ENTER

BEEP      NUMBER OF YEARS

3 ENTER

STARTING WITH $525
AFTER 1 YEAR...
PRINCIPAL = $564.375

AFTER 2 YEARS...
PRINCIPAL = $606.703125

AFTER 3 YEARS...
PRINCIPAL = $652.2058594

```

Timer

The program has the computer ask you how many minutes you want on the clock. Keying in the number starts the clock ticking. At the end of the period, the clock will tell you "time is up."

To speed up or slow down the clock, change the .85 in line 30. A larger number will speed it up; a smaller number will slow it down. For example, .90 is faster while .20 is slower.

Program Listing

```

10 "S" CLEAR
20 BEEP 1:INPUT " MINUTES = ";A
30 C=60A/.85
40 FOR B=1 TO C
50 PAUSE"TICK"
60 NEXT B
70 BEEP 2:PRINT" TIME IS UP"
80 GOTO 20

```

Checking Account Monthly Balancer

On the first day Man created the computer. On the second day, he created the High-Low Numbers game. On the third day, he created check-balancing software.

For the pocket computer, you need a fast-loading easy-to-use monthly balancer which won't bust your memory budget. This handy program does the whole bit, including warnings of impending overdrafts and service-charge deductions. Note: the service charges, overdraft penalties, and other rules of your bank may differ from mine. See your bank's monthly statement for specifics.

The brief sample run included here doesn't begin to cover all the surprises built into this software. Load it and give 'er a try. You'll love it!

Program Listing

```
410 "B" CLEAR
420 FOR X=1 TO 3
      :PAUSE" * CHECKBOOK BALANCER *"
      :NEXT X
430 IF B THEN 460
440 BEEP 1:INPUT"OPENING BALANCE = $";B
445 IF B THEN 460
450 BEEP 3
      :PAUSE"ACCOUNT BALANCE = ZERO"
455 PRINT"ACCOUNT CANCELLED":GOTO440
460 BEEP 1:PAUSE"CHECK OR DEPOSIT ?"
470 INPUT" C OR D ?",A$
480 PAUSE" THANK YOU"
490 IF A$="C" THEN 600
500 BEEP1:INPUT"DEPOSIT AMOUNT = $";D
510 B=B+D
520 GOTO 900
600 BEEP 1:INPUT"AMOUNT OF CHECK = $";C
610 B=B-C
620 GOTO 800
700 IF B>=300 PAUSE"NO SERVICE CHARGE"
```

```

710 IF B<300 LET B=B-1
      :PAUSE"BALANCE UNDER $300"
      :PAUSE"$1 SERVICE CHARGE"
720 IF B<200 LET B=B-1
      :PAUSE"BALANCE UNDER $200"
      :PAUSE"$1 MORE SERVICE CHARGE"
730 IF B<100 LET B=B-1
      :PAUSE"BALANCE UNDER $100"
      :PAUSE"$1 MORE SERVICE CHARGE"
800 IF B<Ø THEN 820
810 GOTO 870
820 BEEP 3:FOR Z=1 TO 3:BEEP 1
      :PAUSE"OVERDRAFT":NEXT Z
830 PAUSE"$5 OVERDRAFT FEE PAID"
      :W=W+1
840 B=B-5:E=INT((ABS B)+1)
850 IF W>2 PAUSE"EXCESSIVE OVERDRAFTS"
      :GOTO455
860 PAUSE"DEPOSIT $";E;" IMMEDIATELY"
      :GOTO 500
870 IF B =Ø BEEP 3:PAUSE"BALANCE = ZERO"
      :PAUSE"DEPOSIT $1 IMMEDIATELY"
      :GOTO 500
900 BEEP 2:PRINT"YOUR NEW BALANCE = $";B
910 BEEP 1:PAUSE"WANT TO DEDUCT..."
920 INPUT"SERVICE CHARGES ? ",F$
930 IF F$="YES" THEN 700
940 GOTO 445

```

Sample Run

```

DEF MODE
      SHIFT B
      * CHECKBOOK BALANCER *
      * CHECKBOOK BALANCER *
      * CHECKBOOK BALANCER *
BEEP  OPENING BALANCE = $
      300 ENTER
BEEP  CHECK OR DEPOSIT ?
      C OR D ?
      C ENTER
BEEP  AMOUNT OF CHECK = $
      225 ENTER
BEEPBEEP  YOUR NEW BALANCE=$75

```

Mortgage Loan: Payment, Present Value, Number of Payments

Planning on buying a new home? Thinking of refinancing the old mortgage? This program calculates either mortgage payment amount, number of payments, or amount borrowed.

Program Listing

```
10 "M" CLEAR
20 BEEP 1
   :INPUT"WANT PRESENT VALUE ?",A$
30 IF A$="YES" THEN 200
40 BEEP 1
   :INPUT"WANT PAYMENT AMOUNT ?",B$
50 IF B$="YES" THEN 300
60 BEEP 1
   :INPUT"WANT NUMBER PAYMENTS ?",C$
70 IF C$="YES" THEN 400
80 IF C$="NO" THEN 10
90 PAUSE"ANSWER YES OR NO"
100 GOTO 10
200 BEEP 1:INPUT"PAYMENT AMOUNT = $";P
210 BEEP 1:INPUT"INTEREST RATE % = ";I
220 BEEP 1:INPUT"NUMBER OF MONTHS = ";N
230 I=.01I/12
240 V=P*((1-(1+I)^-N)/I)
250 BEEP 2:PRINT"PRESENT VALUE = $";V
260 GOTO 10
300 BEEP 1:INPUT"PRESENT VALUE = $";V
310 BEEP 1:INPUT"INTEREST RATE % = ";I
320 BEEP 1:INPUT"NUMBER OF MONTHS = ";N
330 I=.01I/12
340 P=V*(I/(1-(1+I)^-N))
350 BEEP 2:PRINT"PAYMENT = $";P
360 GOTO 10
400 BEEP 1:INPUT"PAYMENT AMOUNT = $";P
410 BEEP 1:INPUT"PRESENT VALUE = $";V
```

```

420 BEEP 1:INPUT"INTEREST RATE % = ";I
430 I=.01I/12
440 N=-((LOG(1-I*V/P))/(LOG(1+I)))
450 BEEP 2
      :PRINT"NUMBER OF PAYMENTS = ";N
460 GOTO 10

```

Sample Run

```

DEF MODE
BEEP          SHIFT M
              WANT PRESENT VALUE?

              YES
              ENTER

BEEP          PAYMENT AMOUNT = $

              50
              ENTER

BEEP          INTEREST RATE % =

              11
              ENTER

BEEP          NUMBER OF MONTHS =

              36
              ENTER

BEEPBEEP     PRESENT VALUE = $1527.24

```

Random Number Generator I

This program accepts a seed number in the range of 1 to 100 and generates three pseudorandom numbers in the range of 1 to 10.

All of these random number generators could be used as subroutines or GOTO objects in larger programs.

Program Listing

```
10 "H" CLEAR
20 BEEP 1:INPUT " HOW OLD ARE YOU ?",U
30 R=.01U
40 FOR W = 1 TO 3
50 R=((R+R)A5)-INT((R+R)A5)
   :T=10*R:X=INT T
60 IF X>10 THEN 50
70 IF X<1 THEN 50
80 A(W)=X
90 NEXT W
100 PAUSE"FIRST RANDOM NUMBER = ";A
110 PAUSE"SECOND RANDOM NUMBER = ";B
120 PAUSE"THIRD RANDOM NUMBER = ";C
130 END
```

Sample Run

AGE	A	B	C
23	6	5	3
37	9	7	9
41	1	6	6
56	9	5	3
62	1	8	9
78	4	5	7
84	6	2	2

Random Number Generator II

This pseudorandom number generator accepts seed numbers in the range of 1 to 100 and generates numbers in the range of 1 to 9.

Program Listing

```
10 "G" CLEAR
20 BEEP 1:INPUT" HOW OLD ARE YOU ?",S
30 A=SS:Y=(997*X+A)/199:X=Y-INT Y
   :X=X*11:S=INT X
40 IF S=0 THEN 30
```



```

50 PRINT" RANDOM NUMBER = ";S
60 GOTO 30

```

Sample Run

SEED	33	18	27	41	59	66	72
	4	6	6	4	4	8	5
	7	6	4	4	7	8	9
	6	7	7	4	9	5	5
	2	3	9	4	1	1	6
	9	9	4	9	4	3	8
	4	2	9	1	8	5	6
	5	1	2	2	5	9	6
	1	3	7	4	1	1	4
	7	1	1	9	2	1	9
	6	8	3	5	3	5	2

To limit the output range to 1 to 6, add line 45:

```

45 IF S>6 THEN 30

```

Here is a sample run with output limited to the 1-6 range:

Sample Run

SEED	33
	4
	6
	2
	4
	5
	1
	6
	1

Random Number Generator III

This pseudorandom number generator computes 26 random numbers in the range of 1 to 100 and deposits those numbers in memory locations A through Z.

Program Listing

```
10 "V" CLEAR
30 FOR A(27)=1 TO 26
40 A(30)=(A(27)*A(27))
   :A(28)=(997*A(29)+A(30))/199
   :A(29)=A(28)-INT(A28)
50 A(29)=A(29)*2:A(29)=INT A(29)
60 A(A(27))=A(29)
70 NEXT A(27)
80 END
```

Sample Run

A = 0	H = 0	O = 19	U = 58
B = 2	I = 40	P = 47	V = 1
C = 6	J = 90	Q = 92	W = 66
D = 14	K = 51	R = 55	X = 55
E = 26	L = 23	S = 36	Y = 69
F = 44	M = 8	T = 37	Z = 9
G = 68	N = 6		

Random Number Generator IV

This program generates a series of 25 pseudorandom numbers in the range of 1 to 100. The series will always be the same as long as 25 is the last number in line 50. You can increase the number of numbers generated by changing the 25 in line 50 to a larger number. That will change the string of numbers by adding to it.

Program Listing

```
10 "Z" CLEAR
20 REM THIS PROGRAM GENERATES A SERIES OF
30 REM RANDOM NUMBERS IN THE RANGE
40 REM OF 1 TO 100
50 FOR N= 1 TO 25
60 A=N*N
70 Y=(997*X+A)/199
80 X=Y-INT Y
```

```

90 X=X*E2
100 X=INT X
110 IF X=0 THEN 130
120 BEEP 1:PAUSE X
130 NEXT N
140 BEEP 2:PRINT"END OF RUN"
150 END

```

Sample Run

2	21	22	14	23
56	25	54	25	11
57	37	95	78	24
37	57	21	19	35
25	85	22	63	END OF RUN

Random Numbers: Averages

It is interesting to know what the average number is of a series of randomly-generated numbers. This software generates pseudorandom numbers in the range of 1 to 100 in a series. The program totals the numbers in the series after each generation and displays the average of all numbers so far in the series.

Program Listing

```

10 "C" CLEAR
20 R=π+995R:R=R-INT R:R=99R+1:W=INT R
30 PAUSE" RANDOM NUMBER = ";W
40 X=X+R:Y=Y+1:Z=X/Y
50 PAUSE "AVERAGE = ";Z
60 GOTO 20

```

Sample Run

RANDOM NUMBER = 15	RANDOM NUMBER = 80
AVERAGE = 15	AVERAGE = 55.67
RANDOM NUMBER = 72	RANDOM NUMBER = 91
AVERAGE = 43.5	AVERAGE = 64.5

RANDOM NUMBER = 65
AVERAGE = 64.6

RANDOM NUMBER = 41
AVERAGE = 62.14

RANDOM NUMBER = 71
AVERAGE = 65.67

RANDOM NUMBER = 26
AVERAGE = 57.63

Beep Commander

This re-beeper talks as many times as you tell it to. As long as you key in the value of A in line 410 each time, you are the commander. If you substitute a random number generator (found elsewhere in this book) to obtain the value for A, you lose control. This would make a good subroutine in a larger program.

Program Listing

```
400 "C" CLEAR
410 INPUT "NUMBER BEEPS = ";A
420 BEEP A
430 GOTO 410
```

Whodunit?

The lord of the manor has been found murdered. The butler, the gardener, the nanny all seem guilty. And a burglar was on the scene. Whodunit?

This is the first in a series of fast-loading games which can be keyed quickly into your Pocket Computer. Each of these games will provide tons of fun in those spare moments at your work or school desk.

Program Listing

```
10 "M" CLEAR
20 BEEP 1:INPUT"MYSTERY NUMBER = ";S
30 A=SS:Y=(997*X+A)/199:X=Y-INT Y
   :X=10X:S=INT X
40 IF S=0 THEN 30
50 IF S>4 THEN 30
```

```

60 A$="BUTLER":B$="NANNY":C$="GARDNER"
   :D$="BURGLAR"
70 PAUSE"WHO KILLED THE DUKE ?"
   :PAUSE"WAS IT...":FOR J=1 TO 4
   :PAUSE A$(J):NEXT J
80 BEEP 1:INPUT"WHODUNIT ?",P$
90 IF A$(S)=P$ LET R=R+1:BEEP 2
   :PAUSE" RIGHT !":GOTO 90
100 PAUSE"NOT THE ";P$:W=W+1:GOTO 70
110 PAUSE"YOUR SCORE IS..."
   :PRINT USING"###";" ";R;" RIGHT "
   ;W;" WRONG"
120 USING:GOTO 20

```

Klingon Killer

BLEEP. BLEEP. BLEEP. The warning alarm is screaming. A Klingon raider has been spotted. It is attacking. You swing your laser gun and fire. Will you kill him? Damage him? Miss him? If you miss, will he get you?

This is another in the series of quickie games, just the right length for keying into your Pocket Computer over lunch at your desk at work or in school. Beware: You're liable to become so excited you'll miss the end of your lunch hour!

Program Listing

```

10 "K" CLEAR
20 FOR L=1 TO 3:BEEP 1
   : PAUSE" RAIDERS ATTACKING":NEXT L
30 PAUSE"STANDBY TO FIRE LASERS"
40 Q=0:BEEP 1
   :INPUT"NUMBER SHOTS FIRED = ";S
50 B=S
60 IF Q=B THEN 120
70 A=SS:Y=(997*X+A)/199
   :X=Y-INT Y:X=10X:S=INT X
80 IF S<4 PAUSE" YOU DAMAGED HIM"
   :Q=Q+1:GOTO 60

```

```

90 IF S=4 BEEP 2
   :PAUSE" YOU KILLED HIM":W=W+1
   :GOTO 130
100 IF S=5 PAUSE" HE GOT YOU"
   :V=V+1:GOTO 130
110 IF S>5 PAUSE" YOU MISSED"
   :Q=Q+1:GOTO 60
120 PAUSE" HE GOT YOU"
   :V=V+1:GOTO 130
130 PAUSE" SCORE..."
   :PRINT USING"###";" KLINGONS"
   ;V;" YOU";W
140 USING:GOTO 40

```

Parameters

What! Another high-low numbers game? Yep. It's the all-time most favorite computer game. The difference here is we've turned it into one of our three-minute fast-loaders so you won't be bored after school. Or in the back seat of the limo while your driver pedals you to Old Work Twenty.

Any rules? Sure. The computer picks a secret number. You try to guess it. The computer gives you too-high or too-low clues and keeps score. *Same old game.* Yep. But perfect during a boring lecture.

Program Listing

```

10 "B" CLEAR
20 R=INT(99*(( $(\pi+R)^5$ )-INT( $(\pi+R)^5$ )))
30 BEEP 1:INPUT"GUESS THE NUMBER",B
40 T=T+1
50 IF B>R BEEP1
   :INPUT"TOO HIGH. GUESS AGAIN",B
   :GOTO 40
60 IF B<R BEEP1
   :INPUT"TOO LOW. GUESS AGAIN",B
   :GOTO 40
70 PAUSE USING "###";"YES!";R
   ;" IS THE NUMBER"

```

```

80 PRINT"IT TOOK YOU ";T;" TRIES"
90 T=0:GOTO 20

```

Craps

The world's oldest game transformed into a futuristic setting: computer dice. The computer rolls the dice, notes your point, cheers your wins and commiserates after your losses.

Snake eyes. Lucky seven. The roll. The point. Just be sure not to crap out!

Program Listing

```

10 "C" CLEAR
20 FOR G=1 TO 3:PAUSE"          ** CRAPS **"
  :NEXT G
30 PRINT" READY TO ROLL DICE"
40 C=C+1
50 GOSUB 600
60 X=W:PAUSE USING"##";"          >;X;" <"
70 GOSUB 600
80 Y=W"PAUSE"          >;Y;" <"
90 Z=X+Y
100 IF C=1 LET B=Z:GOTO 120
110 IF Z=B THEN 400
120 IF Z=7 THEN 200
130 IF Z=2 THEN 300
140 BEEP 1
  :PRINT USING"###";X;Y;"  YOUR POINT=";B
150 GOTO 40
200 IF C=1 PAUSE X;Y;" HOORAY!"
  :PAUSE " LUCKY SEVEN!":BEEP 3
  :PRINTX;Y;" YOU WIN!":D=D+1:GOTO 500
210 PAUSE X;Y;" TOUGH LUCK!"
  :PAUSE" CRAPPED OUT":BEEP2
  :PRINT X;Y;" YOU LOSE!":E=E+1
  :GOTO 500
300 PAUSE X;Y;" SORRY!":PAUSE" SNAKE EYES"
  :BEEP 2:PRINT X;Y;" YOU LOSE!"
  :E=E+1:GOTO 500

```

```

400 PAUSE" POINT!":PAUSE" YOU GOT ";B
    :BEEP 3:PRINT X;" +";Y;"=";B;" YOU WIN!"
    :D=D+1:GOTO 500
500 PAUSE"":PAUSE" YOU ROLLED ";C;" TIMES"
    :C=0:PAUSE""
510 BEEP 1:PAUSE" ROLL AGAIN OR QUIT ?"
    :INPUT" R OR Q ?",F$
520 IF F$="R" PAUSE" OKAY"
    :PAUSE" HERE WE GO...":GOTO 40
530 IF F$="Q" PAUSE" OKAY"
    :PRINT" YOU WON ";D;" AND LOST ";E
540 PAUSE" PLEASE ANSWER R OR Q":GOTO 510
550 END
600 R=((PI+R)^5)-INT((PI+R)^5):R=10*R
    :W=INT R
610 IF W>6 THEN 600
620 IF W<1 THEN 600
630 RETURN
640 END

```

Cylon Raider Alert

Danger! Three Cylon raiders have entered your sector of the galaxy. Alarms sound. You check your laser torpedoes, find 40 left. Now...find the Cylons and blast them.

Note: planets in your sector don't have names. Rather, they are numbered 1 to 99. You guess the Cylons are hiding behind any two planet numbers *including anywhere between* those two planets. In other words, you are selecting a range of numbers when you key in two planet numbers.

There are only three Cylons. The game starts over after you find all three.

Program Listing

```

10 "C" CLEAR
20 P=40
30 GOSUB 360
40 X=Z

```



```

50 GOSUB 360
60 Y=Z
70 GOSUB 360
80 N=Ø
90 BEEP 3:PAUSE" THREE CYLON RAIDERS"
100 PAUSE " IN YOUR SECTOR"
110 PAUSE""
120 PAUSE" ARM LASER TORPEDOES"
130 PAUSE USING"###";" YOU HAVE ";P;" LEFT"
140 PAUSE""
150 PAUSE" YOU MUST FIND..."
160 PAUSE" AND BLAST THE CYLONS"
165 PAUSE""
170 INPUT" WHERE IS ONE HIDING ?",A
180 IF J=1 INPUT" OR A SECOND LOCATION ?",B
:GOTO 200
190 INPUT" AND ANOTHER CYLON ?",B
200 C=Ø
210 N=N+2:P=P-2:GOSUB 570
220 FOR I=24 TO 26
230 J=AA(I)+BA(I)-AB-A(I)A(I)
240 IF J=Ø BEEP 2
:PRINT USING "###";"DIRECT HIT! AT";A(I)
:A(I)=Ø:Q=Q+1
250 C=C+(J>Ø)
260 NEXT I
270 J=SGN X + SGN Y + SGN Z
280 IF J=Ø PAUSE" YOU GOT THEM ALL !"
:PRINT USING"###";" ";N;" TORPEDOES
LAUNCHED":GOTO 400
300 PAUSE USING"###";" YOU CORNERED "
;C;" OF ";J
310 PAUSE USING;" BETWEEN STAR ";A;".."
320 PAUSE" AND STAR ";B
330 PAUSE USING"###";" YOU FIRED ";N
;" TORPEDOES"
340 PAUSE " YOU HAVE ";P;" TORPEDOES"
350 GOTO 170
360 R=R+983R
370 R=R-INT R
380 Z=INT 99R+1
390 RETURN

```

```

400 K=N/3
410 PAUSE USING"###.#"; " THATS "
    ;K;" PER CYLON"
420 IF K<2 PAUSE" A PERFECT SCORE !"
    :GOTO 470
430 IF K<3 PAUSE" VERY NICE !":GOTO 470
440 IF K<4 PAUSE" NOT TOO BAD":GOTO 470
450 PAUSE" THATS TOO MANY..."
460 PAUSE" YOU WASTED TORPEDOES"
470 PAUSE""
480 BEEP 1:INPUT" WANNA TRY AGAIN ?",L$
490 IF L$="NO"GOTO 520
500 IF L$="YES" PAUSE" OKAY. STANDBY..."
    :GOTO 30
510 PAUSE " JUST SAY YES OR NO"
    :GOTO 480
520 PAUSE" OKAY":GOTO 630
530 PAUSE " IF MORE CYLONS SHOW UP"
540 PAUSE " I WILL BEEP"
550 BEEP 3
560 PRINT" TURN ME OFF"
570 IF P=Ø THEN 590
580 RETURN
590 BEEP 2
600 PAUSE" NO MORE TORPEDOES"
610 PAUSE" RETURN TO BASE..."
620 PAUSE" FOR RESUPPLY."
630 PAUSE""
640 PAUSE" YOU BLASTED A TOTAL..."
650 PAUSE USING"###"; " OF ";Q;" CYLONS"
660 PAUSE""
670 GOTO 530

```

Old West Shootout

You are Marshall Matt Dillon. Billy the Kid is in town. You can't avoid your duty: the kid must be arrested. It's high noon!

You must plug the gunfighter before he guns you down. But where is he hiding? He could be down in the

corral or up on the hotel roof. He might have slipped into the stable or down behind the bar in the saloon. He could be inside the house or outside in the wagon. He might be behind the railroad station or in the doctor's office. And, worst of all, he may have brought a friend!

Follow the clues from your handy computer, fastest figurer in the West. Just don't stop any bullets.

Program Listing

```

10 "X"CLEAR
20 FOR A=1 TO 3
   :PAUSE" ** OLD WEST SHOOTOUT **"
   :NEXT A
30 A$="CORRAL":B$="HOTEL":C$="STABLE"
   :D$="SALOON":E$="HOUSE":F$="WAGON"
35 G$="STATION":H$="STORE":I$="OFFICE"
40 PAUSE"":PAUSE"YOU MUST SHOOT"
   :PAUSE"THE GUNFIGHTER"
   :PAUSE"BEFORE HE SHOOTS YOU"
50 PAUSE"":PAUSE"HE COULD BE..."
60 FOR J=1 TO 9:PAUSE"IN THE ";A$(J)
   :NEXT J
70 BEEP 1:INPUT"WHERE IS HE ?",K$
80 BEEP 1:INPUT"WHERE ARE YOU ?",L$
90 PAUSE"YOU ARE SHOOTING"
   :PAUSE"FROM THE ";L$
   :PAUSE"INTO THE ";K$
95 IF Z THEN 110
100 GOSUB "X"
110 PAUSE"          > BANG <".
   :PAUSE"          > BANG <":T=T+1
115 IF K$=A$(S)THEN 200
120 IF L$=A$(S)THEN "SAME"
125 PAUSE"SORRY !"
   :PAUSE"YOU MISSED HIM":P=P+1
   :W=W+1:Z=Z+1
130 IF W>2 THEN "HE WINS"
135 IF W=1 THEN "NOTCLU"
140 V=S:GOSUB "CLUE"
150 PAUSE"CLUE: ";Q$:PAUSE"TRY AGAIN"
   :GOTO 70

```

```

200 PAUSE"YOU GOT HIM !"
    :PAUSE"HE WAS IN THE ";A$(S):M=M+1
210 PAUSE"":PAUSE"THERES ANOTHER BAD GUY"
    :W=Ø:INPUT"WANNA GUNFIGHT AGAIN?",N$(
220 IF N$="YES" THEN 50
230 IF N$="NO" THEN 250
240 PAUSE"ANSWER YES OR NO":GOTO 210
250 PAUSE"":PAUSE"OKAY. YOU MISSED..."
260 PAUSE USING"###";P;" OUT OF ";T;" SHOTS"
270 U=INT (100*((T-P)/T))
    :PAUSE USING"####";"YOU ARE SHOOTING "
    ;U;"%"
280 PRINT USING"##";"YOU WON ";M;" LOST ";O
300 "¥" R=((W+R)∧5)-INT((W+R)∧5)
    :R=10*R:S=INT R
310 IF S>9 THEN 300
320 IF S<1 THEN 300
330 RETURN
400 "CLUE" IF V=1 LET Q$="HORSES"
410 IF V=2 LET Q$="ROOMS"
420 IF V=3 LET Q$="HAY"
430 IF V=4 LET Q$="BAR"
440 IF V=5 LET Q$="HOME"
450 IF V=6 LET Q$="WHEEL"
460 IF V=7 LET Q$="RAIL"
470 IF V=8 LET Q$="GROCER"
480 IF V=9 LET Q$="DOC"
490 RETURN
500 "NOTCLU" V=S+1
510 IF V>9 LET V=1
520 GOSUB "CLUE"
530 PAUSE"CLUE: NOT ";Q$
    :PAUSE" TRY AGAIN":GOTO 70
600 "SAME" PAUSE"":PAUSE" OOPS !"
610 PAUSE"HE IS IN THE ";L$(
620 PAUSE"HE JUST SHOT YOU !":O=O+1
630 GOTO 210
700 "HE WINS" PAUSE" <- BANG ->"
710 PAUSE"HE GOT YOU"
    :PAUSE"YOU ARE DEAD !":O=O+1
720 PAUSE"HE WAS IN THE ";A$(S)
730 GOTO 210

```


OTHER EXCITING BOOKS FROM THE SHACK® LIBRARY FOR THE TRS-80 MODEL I/III



TRS-80 Assembly Language Programming. Covers microcomputers, hardware, and software. 304 pages. **62-2006** **3.95**

BASIC Computer Programming. Introduction to computers and writing programs. 156 pages. **62-2015** **2.95**

BASIC Computer Language. Self-paced programmed instruction for Level I BASIC quickly gets you into programming. Sample programs. 232 pages. **62-2016** **5.95**

LEVEL II BASIC. Self-study Level II BASIC for the TRS-80 Model I and III. 351 pages. **62-2061** **9.95**

Programming Techniques for Level II. An excellent supplement to Level II BASIC (above). 142 pages. **62-2062** **4.95**

TRS-80 Graphics. Shows how to create computer graphic displays—from line drawings to geometrics to simple moving-figure animation. 142 pages. **62-2063** **5.95**

TRS-80 Programs. 32 fully-documented, ready-to-run programs. 267 pages. **62-2054** **5.95**

Program In Style. Packed with applications for the TRS-80 Level II, plus ideas for creating your own programs. 343 pages. **62-2067** **10.95**

Computer Games. Exciting and challenging games for the Level II TRS-80. 152 pages. **62-2058** **7.95**

Home Computer Programs. Includes finance, automobile-related, kitchen helpmates, tutorial, recreation, more. 330 pages. **62-2069** **7.95**

Introduction to TRS-80 Graphics. Discusses line drawings, special graphics characters, pixel graphics, simple animation. 134 pages. **62-2073** **7.95**

Advanced Level II BASIC. Helps you fully utilize the capabilities of the TRS-80 Model I and III. Sample programs and chapter summaries. 250 pages. **62-2072** **9.95**