PROBLEM-SOLVING ON THE TRS-80™ POCKET COMPUTER

A SELF-TEACHING GUIDE

DON INMAN
JIM CONLAN

PROBLEM-SOLVING ON THE TRS-80 POCKET COMPUTER®

DON INMAN

JIM CONLAN

Dymax Corporation
Menlo Park, California

1807 175 YEARS OF PUBLISHING 1982

John Wiley & Sons, Inc.
New York • Chichester • Brisbane • Toronto • Singapore

Library of Congress Number 81-10358

ISBN: 0 471 09270-3
ISBN: 0 471 86808-6 RS

Printed in the United States of America

82 83 10 9 8 7 6 5 4 3 2 1

TO THE READER

Microcomputers continue to appear in more and more places. As they become smaller, they become more portable and, hence, useful in more places. The Radio Shack TRS-80 Pocket Computer is the ultimate in portability. It can be tucked away in your pocket and used wherever you go. Due to its portability, new uses will be continually discovered for this versatile tool.

This book shows a wide variety of problems that are easily solved with the Pocket Computer. Although some problems may seem to be presented with tongue in cheek, don't let the light-hearted approach fool you. All demonstration programs are practical in some way and, hopefully, will coax you to further applications. The problems cover a wide range of difficulty, but hints are provided that may be either used or ignored as your needs demand.

We assume that you have some familiarity with computing and with BASIC language as used on other computers. If you have never encountered BASIC language, we suggest *TRS-80 BASIC;* Albrecht, Inman, and Zamora; John Wiley & Sons, Inc., 1980. (It is also sold at Radio Shack stores under the title *TRS-80 Level II BASIC.*)

Our book has been organized to meet two main objectives:

1. Teaching the Tool – The TRS-80 Pocket Computer BASIC language and the use of the Pocket Computer's keyboard dominate the early chapters in the book. Problems are introduced as demonstrations to facilitate this objective.
2. Teaching Problem Solving Techniques – The techniques of problem solving dominate the discussions in the later chapters. Here the computer is regarded as a precision tool that is used to solve problems.

After covering the first five chapters, you will be able to make full use of the Pocket Computer. You will then be ready to solve the numerous problems that follow. Many different types of problems demonstrating a wide variety of applications are given throughout the book. Hints that are provided for the solutions to the problems are often quite extensive. You may choose not to use them, but they are there if you need them.

Often, more than one solution is given for a problem. This is true to life. There are many ways to solve most problems, and we do not all solve our problems in the same way.

The TRS-80 Pocket Computer is the tool used in this book. It is a small, but sophisticated, tool. You will find that the more that you use it, the more useful it is to you. This book is designed to help you explore the Pocket Computer in depth so that you can use it successfully in solving problems that you encounter outside of this book.

Contents

CHAPTER ONE

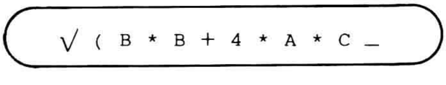
Pocket View of the Pocket Computer

The TRS-80 Pocket Computer is a unique mixture of a programmable calculator and a general-purpose computer. Although this book's main purpose is to illustrate procedures and techniques for solving problems with the Pocket Computer, some space must be allotted to a description of this unique problem-solving tool. Like any artisan, the problem solver is dependent on the tools that are available, and the ability to make use of fine tools is dependent on one's understanding of them.

This chapter is, therefore, devoted to the introduction of the capabilities of the TRS-80 Pocket Computer. In this chapter you will:

- become acquainted with the general characteristics of the Pocket Computer;
- learn how characters appear on the display;
- discover what kind of variables can be used in a program;
- learn a few BASIC commands and statements that are unique to the Pocket Computer or are used in unique ways;
- discover four different operating modes;
- learn how to use some keys in two different ways;
- learn something about error codes and how to clear them;
- discover PAUSE, a new form of the PRINT statement;
- learn all about the NEW statement;
- learn how to DEBUG programs;
- learn how to check for the amount of unused memory; and
- learn how to continue an interrupted program.

The TRS-80 Pocket Computer can be used as a calculator, as described in the *TRS-80 Pocket Computer Software Manual* (Catalog Number 26–3501) that comes with the computer. There are many wondrous features to be exploited in the calculator mode — you can even find the solution to an algebraic equation. For example, if you have previously entered a value for the variables A, B, and C, the Pocket Computer is capable of solving the square root of the following expression.



```
√ ( B * B + 4 * A * C _
```

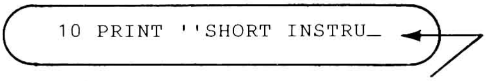
Each of the symbols shown in the example is entered from the keyboard of the Pocket Computer. When you press the ENTER key, the result is displayed.

Description of the Pocket Computer

Imagine carrying around the power of a BASIC-speaking computer in your pocket. No longer are you tied to a computer room or office desk — or even an electric company. Just tuck the computer into your shirt pocket and take it with you wherever a problem exists.

> Note: We didn't say tuck the computer into your *back* pants pocket. The liquid crystal display (LCD) is made of glass. If you sit down with it in your back pocket, you may crunch the LCD.

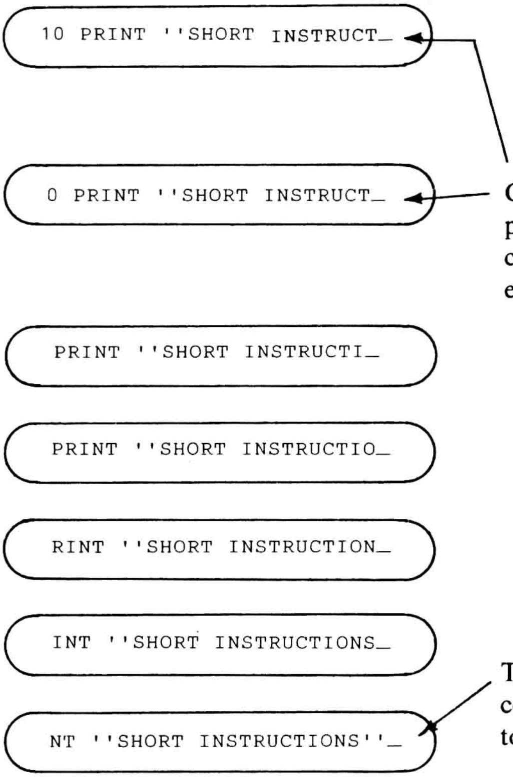
As long as the Liquid Crystal Display has been mentioned, we might as well talk about it first. If you're used to a 64-character, 16-line display of the TRS-80 Model I or Model III (or something similar), the Pocket Computer's 24-character, 1-line display is going to take a little adjusting to. However, within a few minutes, you'll find yourself writing shorter instructions. You may even find out that your programs become easier to read and understand because of the limit of the display size.



```
10 PRINT ''SHORT INSTRU_
```

The cursor is shown at the 24th position. What happens when the next character is typed?

Although only 24 characters are displayed, a BASIC line may contain a maximum of 80 characters. Computer buffs would say, "The size of the input buffer is 80 characters." Let's not get so technical. What they mean is that you can only input 80 characters on one program line. The computer will hold all of them even though you can't see them all at the same time on the display. If you continue to type the previous line, you will see the letters move to the left on the display and off the other end.



```
10 PRINT ''SHORT INSTRUCT_
0 PRINT ''SHORT INSTRUCT_
PRINT ''SHORT INSTRUCTI_
PRINT ''SHORT INSTRUCTIO_
RINT ''SHORT INSTRUCTION_
INT ''SHORT INSTRUCTIONS_
NT ''SHORT INSTRUCTIONS''_
```

Cursor indicates position of next character to be entered.

There! The instruction's complete, ready for you to press ENTER.

Yes, the TRS-80 Pocket Computer has an ENTER key that works just like the ENTER key on other TRS-80 computers (or like the RETURN key on some other computers). The Pocket Computer ENTER key looks like this:

ENTER

The keys of the Pocket Computer are small and closely spaced, but they have a good feel and make clean contact. Numeric and arithmetic operation keys are in a separate area on the right side of the keyboard. Your entries may not go as fast as if you were touch-typing, but you'll soon get up to a respectable speed.

The computer can hold a maximum of 1,424 program steps. There are 26 fixed memories (used for variables) and 178 flexible memories (shared between program steps and variables). You'll learn all about the use of memory in Chapter 2. Calculations are carried out to 10-digit accuracy.

The computer also has editing functions, which include shifting the cursor from left or right, inserting characters, deleting characters, and shifting up or down one line at a time. We'll discuss these functions in Chapter 3.

Calculating capabilities include four arithmetic operations, power calculations, trigonometric and inverse trigonometric functions, logarithms and exponential functions, angular conversions, square roots, sign functions, absolute values, and integer and logic functions.

Power is supplied by mercury batteries. Memory content is *not* lost when the computer is turned off. Memory is protected (or saved) by the batteries when the rest of the computer is turned off.

As interface is available for a cassette recorder to save programs and data on tape and to retrieve them at a later time. There is also a 16-column printer that Radio Shack sells, which includes an interface for both a recorder and the printer.

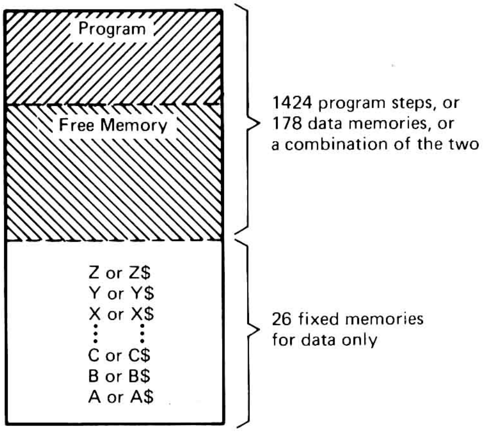
The computer itself is in a small package, 6⅞ inches long, 2¾ inches wide, and 19/32 inches thick. Its weight is just over one-third of a pound. A carrying case, two keyboard templates (more about them later), and a *User's Manual* are all included with the computer.

TRS-80 Pocket Computer BASIC

The BASIC language used by the Pocket Computer can recognize both numeric and string variables. A data memory is called a numeric variable when it stores a numeric value. Numeric variables use labels, or names, such as A, B, C, A(1), A(28), etc. A data memory is called a string variable when it contains a string of characters (limited to seven letters, numbers, or special symbols). String variables are distinguished by the $ symbol and are labeled as A$, B$, C$, A$(1), etc.

> CAUTION: There is a limit of 26 fixed memories used to hold variables. A (a numeric variable) and A$ (a string variable) both assign data to the same fixed memory. This must not be done at the same time. Only one value may be held in a given memory at a given time.

The Pocket Computer's memory is used in much the same way as that of a programmable calculator. Here is how the fixed and flexible memories may be assigned if no arrays are used. The use of memory for arrays is discussed in Chapter 2.



Any memory not used for program steps can be used for data memory, in addition to the 26 fixed memories. This may be done when using arrays. The number of data memories beyond the 26 fixed memories therefore depends upon how many steps are used in the program. The free memory in the previous diagram is used for additional program steps or data memories. In general, you may find out how much memory is not being used by typing in the MEM command in any of the Pocket Computer's operating modes. The display will then show how much data memory and how many program steps are unused. This is more thoroughly discussed in Chapter 2.

Basic Functions, Statements, Commands, and Tape Control Statements

Now let's take a look at the functions, statements, and commands available in Pocket Computer BASIC.

BASIC Functions

ABS	Absolute value
ACS	Arc cos
ASN	Arc sin
ATN	Arc tan
COS	Cosine
DEG	Degree/minute/second to decimal
DMS	Decimal to degree/minute/second
EXP	Exponential function
INT	Integer
LN	Natural logarithm
LOG	Common logarithm
SGN	Sign (positive or negative)
SIN	Sine
TAN	Tangent
√	Square root
π	Pi

BASIC Statements

AREAD	Contents displayed at start of definable program are read into the specified variable
BEEP	Beep sound is generated as many times as specified
CLEAR	Data memory clear
DEGREE	Degree is designated as angle measure
END	Indicates the end of a program
FOR	Beginning of FOR-NEXT loop
GOSUB	Execution is shifted to specified line or label where a subroutine begins
GOTO	Specified line or label is executed
GRAD	Grad is designated as angle measure
IF	Based on the specified condition, a branch is either taken or not taken
INPUT	Allows data to be input
LET	Assigns a value to a variable
NEXT	Ends a FOR-NEXT loop. Increments the step
PAUSE	Displays an output for approximately .85 seconds before going on
PRINT	Displays the specified contents and waits for ENTER key to be pressed
RADIAN	Radian is designated as angle measure
REM	Designates a non-executable statement
RETURN	Returns from subroutine
STEP	Increments a FOR-NEXT loop
STOP	Stops the execution of a program
THEN	Used only with IF statement as a jump instruction
USING	Format designation for display

BASIC Commands

CONT	Restart an interrupted program
DEBUG	Used to debug a program line by line
LIST	For listing a program in memory
MEM	Display amount of unused memory
NEW	Clear memory for a new program
RUN	Execute a program

Tape Control Statements

CHAIN	Program recorded on tape is read and executed
CLOAD	Transfer a program from tape to computer
CLOAD?	Check contents of program with those placed on tape
CSAVE	Record a program on tape
INPUT #	Transfer data from tape to memory of computer
PRINT #	Record data memory contents on tape

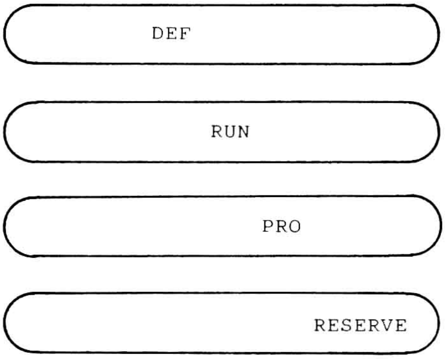
In writing this book, we assume that you have some knowledge of BASIC language. If you do not, we suggest you read *TRS-80 BASIC*; Albrecht, Inman, and Zamora; Wiley & Sons, Inc., 1980. As you look through the Pocket Computer's BASIC commands and statements, you will recognize many of them. We will not dwell on those statements and commands that are common to other computers. However, the Pocket Computer has some new ones, as well as some familiar ones that are used in different ways, which we *will* explain.

Modes of Operation

Before you are introduced to any more BASIC commands or statements, you should take a look at the four available operating modes of the Pocket Computer.

1) Program mode (PRO)
2) Run mode (RUN)
3) Definable mode (DEF)
4) Reserve program mode (RESERVE)

The mode currently being used is displayed at the top of the display.



A mode key, MODE, changes the operating mode from left to right. The mode moves clockwise, as indicated in the following diagram, each time the MODE key is pressed.

→ DEF → RUN → PRO → RESERVE

The operations carried on in each mode are briefly summarized as follows:

1) PRO – The program mode is used to write, enter, and edit programs.
2) RUN – The run mode is used to execute programs or for direct calculations.
3) DEF – The definable mode is used to execute programs that have been defined by a label.
4) RESERVE – The reserve program mode is used to write, enter, and edit programs or functions for reserved keys that will be frequently used.

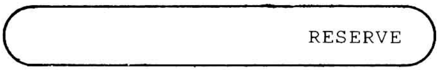
The use of BASIC commands and statements depends upon the operating mode selected. In this chapter, you'll find out how to use the PRO and RUN modes. In actual usage, the operating modes interact with each other and should not be considered isolated from each other. Let's investigate the PRO mode first.

The Program Mode

Access the program mode by pressing the MODE key a number of times (depending on the mode that you are presently in) until you see the PRO prompt on the top of the display. When you turn the Pocket Computer on, it is always in the mode in which it was last used.

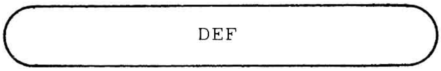
Example:

Suppose you last used the RESERVE mode, and you want the PRO mode now.



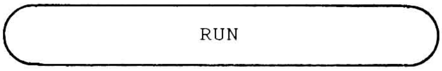
```
                    RESERVE
```

Press the MODE key once, and you see:



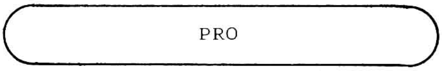
```
DEF
```

Press the MODE key again, and you see:



```
RUN
```

Press the MODE key once more, and you see:

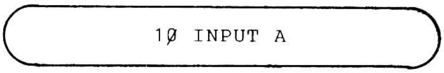


```
PRO
```

You are now in the mode normally used to enter programs into the computer's memory. Lines of a program are numbered using integers ranging from 1 through 999. When a line is complete, it is entered into memory by pressing the ENTER key. One line may contain one or more statements. If more than one statement is used within a line, the statements must be separated by a colon.

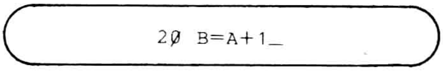
Example:

One statement per line:



```
1Ø INPUT A
```

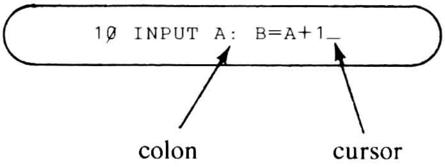
First line



```
2Ø B=A+1_
```

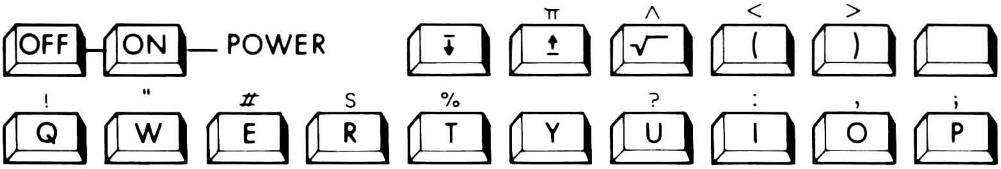
Second line

Or two statements per line:

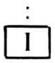


```
1Ø INPUT A: B=A+1_
```

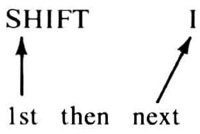
You may have trouble discovering how to display the colon. Notice that some of the top two rows of keys have symbols above them.

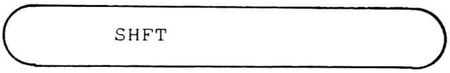


The colon is the symbol above the I key.



To enter the colon, you must first press the SHIFT key and then the I key. This selects the colon for entry. A shift designator appears on the display when the shift key is depressed: It tells the user that the next key stroke will be shifted.





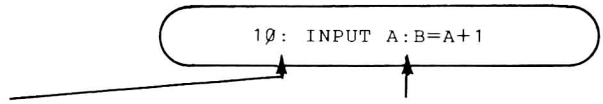
```
SHFT
```

When the shift key is pressed, this appears on the display.

> If you should ever press the SHIFT key by mistake, press it a second time. This will cancel the affect of the first shift.

When the ENTER key has been pressed following the completion of a program line, the line is entered into memory, and the line is displayed once again with a colon following the line number.

> We will not always remind you to press the ENTER key. Just remember that it *must* be pressed at the end of each program line.

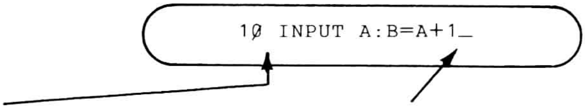


```
1Ø: INPUT A:B=A+1
```

colon follows
line number

colon separating two
statements on one line

If you should ever forget whether you have pressed the ENTER key or not, *look* at the display. If the cursor shows at the end of the line and a colon *does not* show after the line number,



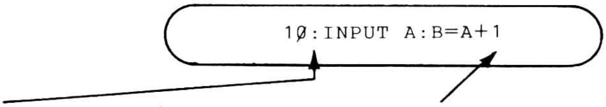
```
1Ø INPUT A:B=A+1_
```

no colon

cursor

the ENTER key *has not been pressed.*

If the cursor does not show and there is a colon following the line number,



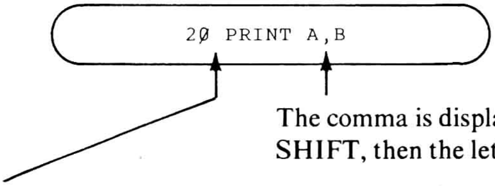
```
1Ø:INPUT A:B=A+1
```

colon

no cursor

the ENTER key *has been pressed.*

Suppose that you have pressed the ENTER key after the first statement at line 10. The computer waits for you to type in the next line of the program. It doesn't give you any visual indication, but it is waiting for you to go on. Type in the next line. For example, type

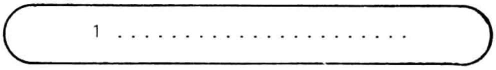


```
2Ø PRINT A,B
```

The comma is displayed by pressing SHIFT, then the letter O.

Do not put a colon here, the computer will do it after you press the ENTER key.

If you put a colon of your own after the line number, the computer will add another. You'll then get an error message when you run the program.

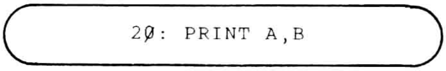


```
1 . . . . . . . . . . . . . . . . . . . . . .
```

To enter line 20, the sequence of keys pressed should be:

2 0 SPC P R I N T SPC A SHIFT O B ENTER

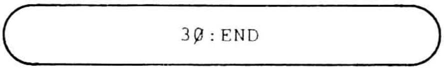
The display would then show



```
2Ø: PRINT A,B
```

Now finish off the program by adding an END statement. Type 30 END, and (of course) press ENTER.

The display:

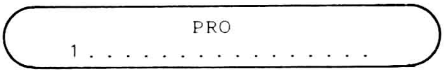


```
3Ø:END
```

The END statement is optional If it is not there, the computer will stop after the last executable line. However, sometimes the END statement must be used to separate the end of the program from data or other information that might follow.
We know that you are just itching to run the program that you just entered, so . . .

The RUN Mode

Suppose that you tried to run the program while you were in the PRO (Program) mode. If you did, this is what you probably saw:

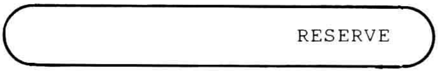


```
PRO
1 . . . . . . . . . . . . . . . .
```

As you know, this is an error code.

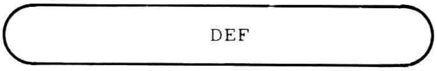
To clear an error code: press the red CL key.

You can't RUN a program while you are in the PRO mode. Press the MODE key three times.



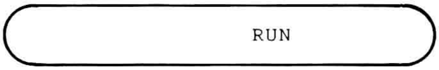
```
RESERVE
```

First time



```
DEF
```

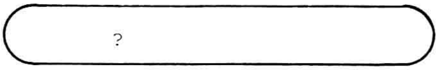
Second time



```
RUN
```

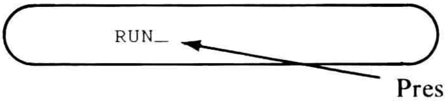
Third time

Now, you can type RUN.
If you remembered to press the ENTER key after typing RUN, you saw this:



```
?
```

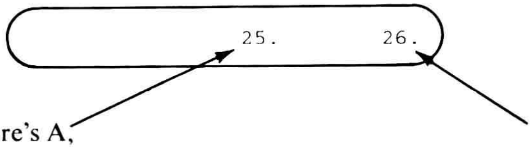
The question mark means ready for an INPUT. However, if you forgot to press the ENTER key after typing the word RUN, the computer is still waiting at this point.



```
RUN_
```

Press the enter key to see the question mark prompt.

Try 25 for an input. Don't forget to press the ENTER key following the input. This is what you will see:



```
25.          26.
```

There's A, your input

There's B, (B = A + 1)

> When the Pocket Computer encounters a PRINT statement, it PRINTS the value specified and waits and waits and waits. It will not do anything else until you press the ENTER key.

When you press the ENTER key, the next statement:

```
3Ø:END
```

is executed, and the ready prompt shows at the beginning of an otherwise empty line.