


# Book Review - Problem Solving on the TRS-80 Pocket Computer

## A Self-Teaching Guide

© 2019 Valentín Albillo

### 1. Introduction

This is my review of the book *“Problem Solving on the TRS-80 Pocket Computer – A Self-Teaching Guide”* by Don Inman and Jim Conlan, nominally intended for the title pocket computer but equally valid for the SHARP models PC-1210, PC-1211 and PC-1212. First of all, the essential data for the book:

Element	Value
Title	PROBLEM SOLVING ON THE TRS-80 POCKET COMPUTER A SELF-TEACHING GUIDE
Author	Inman, Don & Conlan, Jim
Publisher	JOHN WILEY & SONS, Inc.
Copyright	Copyright © 1982 by JOHN WILEY & SONS, Inc.
Subjects	TRS-80 (Computer)—Programming. Computer programs.
Applies to	TRS-80 Pocket Computer, SHARP PC-1210, PC-1211, PC-1212
Pages	255
ISBN	0-471-09270-3
Covers	

This book belongs to the *“Self-Teaching Guide”* series and its main goal isn’t to give readers a library of ready-to-use programs on a number of disciplines but rather to encourage them to create their own programs. To achieve that goal the book essentially provides a number of lessons and exercises distributed in 12 *Chapters*, all of them listed and reviewed in 2. *Contents* below. Each chapter consists of the following sections:

- First, several sample problems of various difficulties are described and solved step by step.
- Then, a number of exercises are presented for the reader to solve, with relevant hints provided.
- Finally, one or more solutions are given for each exercise, as well as a *Summary* for the chapter.

Besides the *Chapters*, there’s also five short *Appendices* which discuss several aspects of the particular version of BASIC implemented in this pocket computer as well as some concepts relative to the optional printer.

## 2. Contents

The book includes the following 12 *Chapters* plus 5 *Appendices* and an *Index*, namely:

- *Chapter 1*      Pocket View of the Pocket Computer      (24 pages)
- *Chapter 2*      Applications, Memory Use, and Definable Mode      (27 pages)
- *Chapter 3*      Error Codes, Editing, and Cassette Use      (23 pages)
- *Chapter 4*      Data Files      (21 pages)
- *Chapter 5*      Trigonometric Functions      (32 pages)
- *Chapter 6*      Operation Time, Logic Functions, and Binary Bins      (17 pages)
- *Chapter 7*      Feedback and Systems      (16 pages)
- *Chapter 8*      Random Walk      (22 pages)
- *Chapter 9*      Computing Interest      (16 pages)
- *Chapter 10*      Storing, Sorting and Searching (Or How to Make Sure  
You Know Where It's At)      (17 pages)
- *Chapter 11*      Chaining Programs from Cassette      ( 8 pages)
- *Chapter 12*      The TRS-80 Pocket Computer Printer      (19 pages)
  
- *Appendix A*      BASIC Statements and Commands      (3 pages)
- *Appendix B*      Special BASIC Functions      (1 page)
- *Appendix C*      Acceptable Abbreviations for BASIC Statements,  
Commands, and Special Functions      (2 pages)
- *Appendix D*      Error Codes      (2 pages)
- *Appendix E*      Printer Terms      (2 pages)
  
- *Index*

The first five *Chapters* deal with the *first* main goal, which is to teach inexperienced readers how to use the “*tool*”, i.e.: the *TRS-80 Pocket Computer*'s dialect of BASIC, and to that effect simple problems are presented to demonstrate the various commands and how to solve the problems using them. A certain familiarity with BASIC and computing in general is assumed, but another particular book is recommended to be read first if that's not the case. The style is light-hearted throughout without sacrificing technical correctness.

Once the reader has acquired a good understanding of the tool and feels comfortable with the command set, the remaining *Chapters* are devoted to the *second* main goal: to teach and discuss diverse problem-solving techniques, using the tool to implement them. After the techniques have been introduced, a considerable number of practical exercises, of varying difficulty and covering a wide range of applications, are presented, which the readers must solve anew with the tool by writing their very own programs for it, with the optional help of useful and at times quite extensive hints on how to accomplish the task. Finally, one or more solutions to the exercises are given for the readers to compare their results against the correct ones.

It is expected that after the readers successfully complete the exercises they'll have acquired a proficiency level which will allow them to solve real-life problems using the knowledge and experience acquired here.

Now let's review each chapter in turn:

### ***Chapter 1 - Pocket View of the Pocket Computer***

This 24-page chapter introduces some of the principal and most basic capabilities of the *TRS-80 Pocket Computer*, such as its physical characteristics (display, keyboard) and programming capabilities (including unique features of its BASIC language dialect, types of variables, the four modes of operation, error codes, debugging), also introducing most basic commands such as **NEW**, **PRINT**, **PAUSE**, **DEBUG**, **MEM**, etc.

Each feature is described in “textbook, teacher-like” style and then the reader is asked to complete some basic operations, step by step, and the first simple programs are introduced. The final sections are “*Summing Up Chapter One*” and “*Chapter One Self-Test*”, 3 exercises for the reader to solve, with solutions given at the end.

## Chapter 2 - Applications, Memory Use, and Definable Mode

This 27-page chapter hints at possible ways to use the Pocket Computer, actually listing 25 applications, from *Up-to-the-second stock market analysis* to *Arithmetic practice*. The computer memory is described in the same interactive manner as in the previous chapter and a simple program is dissected into the steps it occupies in memory. The rest of the chapter explains *DEFInable Mode* and using arrays, all of it very interactively. The chapter ends with the usual *Summary* and the *Self-Test* (3 exercises), with answers for both the exercises and the problems featured mid-chapter (which the reader is asked to provide solutions on the go).

## Chapter 3 - Error Codes, Editing, and Cassette Use

This 23-page chapter explains the Error Codes, how to edit program lines, how to save, load and chain programs from tape, and a discussion on generating and using *pseudo-random* numbers. A problem is immediately introduced for the reader to presently provide code for in an empty text box mid-page, and an erroneous solution is given which the “teacher” then proceeds to debug, error by error, taking more than 7 pages to correct the erroneous very small (6-statement) program.

The next section is *Using the Cassette Recorder*, which is described, and the knowledge is applied to store a random-number generating subroutine used in two mid-chapter exercises for the reader to try and solve. As usual, the chapter ends with the *Summary* and the *Self-Test* (3 exercises), with all answers duly provided.

## Chapter 4 - Data Files

This 21-page chapter explains the concept of data files, how to store them to and retrieve them from tape, and elaborates in a previous toy *Inventory* program, upgrading it to enter data from the keyboard, edit the data and save it to tape. Explanations are given interactively in both *Immediate Mode* and *Program Mode*, then the internal way data files are recorded is described and some timings are given, useful to optimize the approach.

The chapter includes four mid-way exercises for the reader to complete, then ends with the *Summary* and the *Self-Test* (5 exercises), with answers provided for every exercise in the chapter.

## Chapter 5 - Trigonometric Functions

This 32-page chapter deals with the six built-in trigonometric functions and angle measurement systems, discussing at length eight problems ranging all the way from a boat crossing a stream to Fourier series (square waves), giving several hints and providing solutions to all of them, even more than one occasionally. The chapter ends with the usual *Summary* and four self-test exercises, plus their corresponding answers.

## Chapter 6 - Operation Time, Logic Functions, and Binary Bins

This 17-page chapter shows how to time various BASIC statements, how to use that information to shorten solution times by choosing the most efficient methods and variables, and how to use the diverse logic statements and binary operators available. The timing consists of manually timing some 16 assorted operations (executed 500 times each in a loop) to find the operation’s speed in ms, which are shown in a table, ranging from 62 ms for the simplest assignment to 622 ms for a trigonometric function. Finally, 6 questions are asked, with answers.

Next, logical comparisons are introduced, with 14 questions asked (and answered) and 2 problems for the reader to solve. *Binary bins* are then discussed and the reader must solve a *binary counter* problem, with 2 very extensive hints offered and as many different solutions given. The reader must then solve yet another binary-related problem (1 hint & 1 solution given), and the chapter ends with the usual *Summary* and a useful table featuring how to implement 11 frequent logical conditions using the logical operators available, like this:

Condition	Use	Result	
...	...	...	
Not A	(A=0)	1 if (A=0) is false	0 if (A=0) is true
OR	(A>B)<>B	1 if A or B (or both) is true	0 if A and B are both false
...	...	...	

### ***Chapter 7 - Feedback and Systems***

This 16-page chapter deals with growth and feedback, from computing the growth of money in the bank or weight changes from energy expended and calories consumed, to feedback in closed systems. Shortly the first problem for the reader to solve is introduced, the so-called *Multiplier Effect Problem*, complete with a hint and a solution, followed by the *English System Problem* and so on and so forth for a grand total of seven mostly interesting problems in all. A very brief *Summary* ends the chapter but no *Self-Test* exercises from now on.

### ***Chapter 8 - Random Walk***

This 22-page chapter is all concerned with random numbers of various types, showing how to generate them, discussing the different kinds of randomness, sorting data to create histograms and simulating real events. The first problem posed to the reader is to generate a certain kind of random numbers, followed by another problem about random tests, both with hints and solutions as usual.

Another 11 additional problems are introduced for the reader to solve, providing some helpful hints, with final solutions given. Some of the programs do involve histogram creation and simulation of real events (*Open House, Fox and Rabbits*), including an interesting problem about randomly generating short stories but regrettably no sample story is actually produced. The chapter ends again with a very brief *Summary*.

### ***Chapter 9 - Computing Interest***

This 16-page chapter deals with matters having to do with borrowing and lending money, explaining how to compute compound interest, the value of an annuity, time payment size, repayment time, continuous interest, time payment schedules and the present value of a cash flow. The reader is asked to solve about a dozen related exercises, with the usual hints and solutions being provided, as well as the relevant financial formulas needed to implement the solutions and detailed explanations to help focus the implementation.

The chapter ends with the by now usual brief *Summary*.

### ***Chapter 10 - Storing, Sorting and Searching (Or How to Make Sure You Know Where It's At)***

This 17-page chapter introduces the concept of lists and shows how to build them, add elements, delete elements, move elements around, sort the elements in numeric order and search for elements within the list. Seven problems are given for the reader to solve, with very extensive hints which instruct the reader about various types of sorting algorithms, including *Insertion Sort, Exchange Sort, Selection Sort* and *Enumeration Sort*, as well as how to merge two lists into a single one.

Once more, the chapter ends with a very brief *Summary*.

### ***Chapter 11 - Chaining Programs from Cassette***

This relatively short 8-page chapter is all about how to proceed when a program is so large and/or deals with so much data that it won't fit all at once into available memory. The **CHAIN** command is introduced to solve the situation and the reader learns how to use **CHAIN** to load and automatically run under program control additional sections of the program stored in tape, replacing the one in main memory, as well as how to pass variables from one section to the next.

One three-pronged mid-chapter exercise is given for the reader to solve, followed by an interactive exploration of passing values between sections, and the chapter ends with the usual short *Summary*.

### ***Chapter 12 - The TRS-80 Pocket Computer Printer***

This 19-page chapter teaches the readers everything they need to know about the optional printer, such as how to set up the printer, how to print the program listing, how to format numbers and text on the page and even how to (crudely) graph functions. The different behavior of the **PRINT** command depending on whether a printer is attached or not is explained, and the reader is directed to execute some simple printing actions interactively, with a few questions asked all along, as well as given some tips on how to perform such things as left or right alignment of printed values, or how to store long lines of text in memory to be listed later in the printer.

A section on how to graph functions follows next, crudely plotting sine and log curves and then going on to create and print random walks and histograms. Some brief instructions on how to print data to the cassette tape are given, referring the reader to chapters 3 and 4 for the details. The chapter then ends with the mandatory *Summary* and last of all, a **Conclusion** finishes the main part of the book.

The final section of the book includes five *Appendices* and an *Index*, namely:

**Appendix A – BASIC Statements and Commands**

This *Appendix* includes a table with all 34 BASIC statements and commands available, giving the syntax for their general form and describing the function performed, in this format:

<i>Statement</i>	<i>General Form</i>	<i>Function Performed</i>
...	...	...
<i>AREAD</i>	<i>AREAD</i> [ <i>variable</i> ]	Assigns a value to a <i>variable</i> when used in the DEF mode. The value must be displayed before the execution of the defined program.
<i>BEEP</i>	<i>BEEP</i> ([ <i>expression</i> ])	Generates as many beeps as specified by the <i>expression</i> in parentheses.
...	...	...

**Appendix B – Special BASIC Functions**

This *Appendix* includes a table with the 15 BASIC mathematical functions available, giving their common mathematical designation, in this format:

<i>BASIC designation</i>	<i>Common Mathematical Designation or Meaning</i>
...	...
<i>DMS</i>	decimal conversion to degree/minute/second
<i>EXP</i>	$e^x$ or exponential function or antilogarithm for LN
...	...

**Appendix C – Acceptable Abbreviations for BASIC Statements, Commands, and Special Functions**

This *Appendix* includes a table with all the acceptable abbreviations for the 48 reserved BASIC words available, in this format:

<i>Statement, Command or Function</i>	<i>Acceptable Abbreviations</i>			
...	...	...	...	...
<i>GOSUB</i>			GOS.	GOSU.
<i>GOTO</i>	G.	GO.	GOT.	
...	...	...	...	...

## Appendix D – Error Codes

This *Appendix* lists the six *Error Codes* and describes their possible causes, in this format:

Error Code Number	Remarks
...	...
2	Caused by: Line error – Check lines and labels specified by GOTO, GOSUB, RUN, DEBUG or LIST statements. (The lines may not exist.)
...	...

## Appendix E – Printer Terms

This *Appendix* deals with all matters having to do with the printer, such as power, how to recharge the battery, the connector jacks, controls and indicators, each of which is explained in detail, as well as the BASIC commands used with the printer. Two useful diagrams are included.

## Index

As usual, the *Index* is ordered alphabetically by subject and lists the page numbers of the various pages where each subject is dealt with.

## 3. Conclusion

As far as *Self-Teaching Guides* go, this book is a very fine example of the genre. Everything is dealt with in a reassuring, interactive manner, explaining every feature of the Pocket Computer in a simple, direct way but including enough detail to be able to apply them to the problem-solving that follows and beyond. Many interesting problems are introduced for the reader to solve applying what’s just been learned, and very good hints are provided to ease the burden of writing the necessary code. Finally, all solutions are given, often more than one to encourage the reader to think what’s the best way to accomplish the task.

Each chapter is introduced by first enumerating the topics which will be discussed, so that the reader will know what to expect, and then a number of questions and mid-way exercises are featured, ever more interesting as the knowledge accumulates. A useful final *Summary* refreshes the initial enumeration of topics for the chapter, and the first five chapters include additional *Self-Test* exercises at the end for the reader to solve. Solutions and answers are always given for every exercise and every question, which is a very good point for the book as I really hate those books which posit some interesting exercises but then fail to give the corresponding solutions.

Another very good point for the book is the sheer amount of exercises, most of which are quite interesting and even inspiring but nevertheless solvable in a few lines of code. They’ll make readers think about some other related but more complex applications now achievable on their own, which is indeed one of the main goals of this book. The friendly, well-humored style also helps to make the learning process that much more enjoyable.

On the negative side, the book’s very nature of being intended for teaching means that the exercises are necessarily simple to code. You won’t find here any long- or even medium-sized programs or full applications. There are no programs to, say, solve systems of equations, find roots of polynomials, compute amortization tables, fit data, etc. That’s not the goal of this book. Also, although the book is 255 pages long, many pages do include blank portions for the reader to write down the code to solve some exercises, as well as “display captures” of everything, so in the end it feels that much shorter. Finally, while this book will be excellent for people inexperienced with this Pocket Computer, BASIC or programming in general, it’ll be far too simple for others.

## Copyrights

Copyright for this paper and its contents is retained by the author. Permission to use it for non-profit purposes is granted as long as the contents aren’t modified in any way and the copyright is acknowledged.