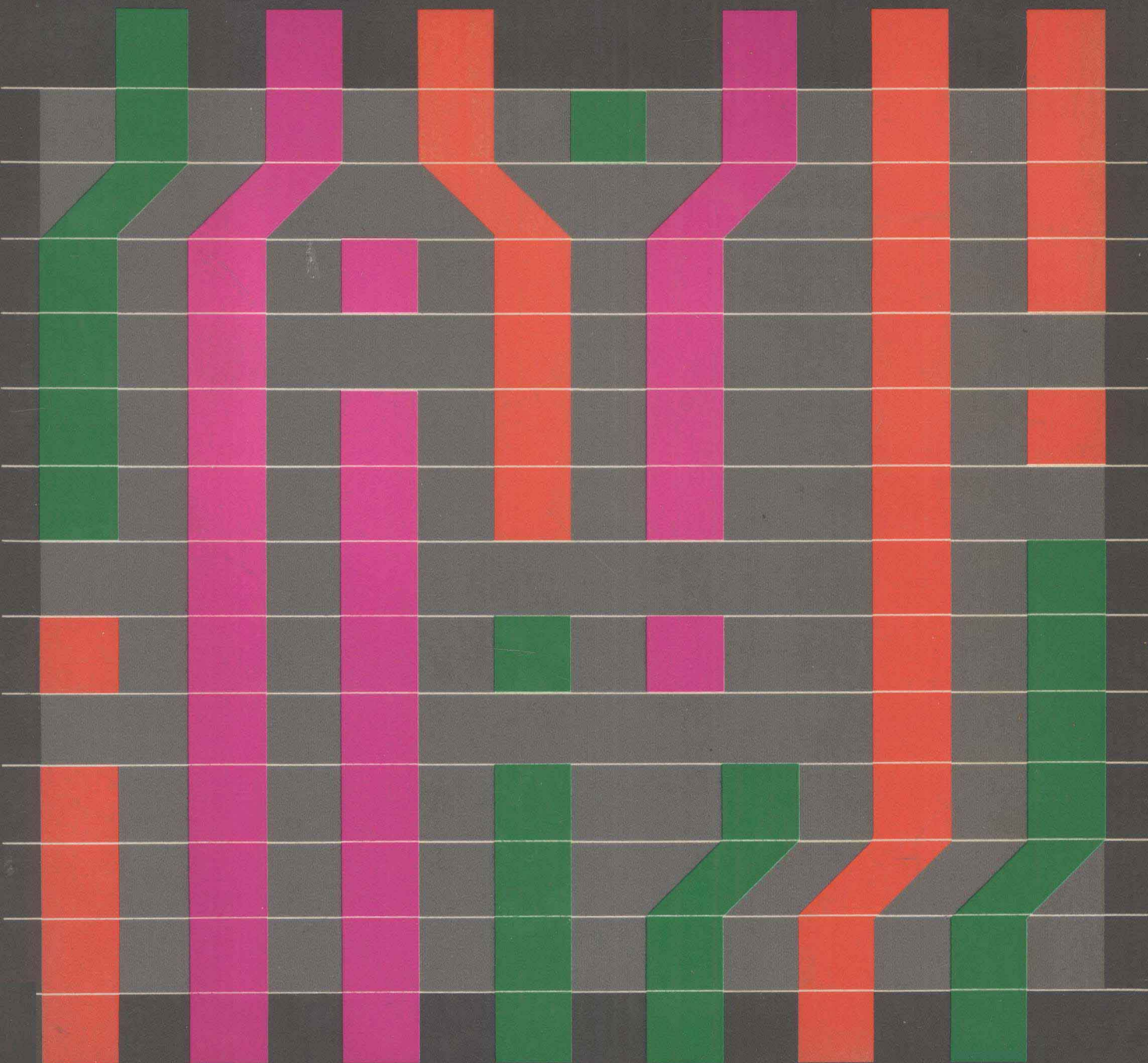


Programming in **BASIC-PLUS**

Jasper J. Sawatzky
Shu-Jen Chen



Programming in BASIC-PLUS

Jasper J. Sawatzky

*Professor, Business Information Systems
Humboldt State University*

Shu-Jen Chen

*Associate Professor, Management Science
California State University, Fullerton*

JOHN WILEY & SONS, New York • Chichester • Brisbane • Toronto • Singapore

Copyright © 1981, by John Wiley & Sons, Inc.

All rights reserved. Published simultaneously in Canada.

Reproduction or translation of any part of this work beyond that permitted by Sections 107 and 108 of the 1976 United States Copyright Act without the permission of the copyright owner is unlawful. Requests for permission or further information should be addressed to the Permissions Department, John Wiley & Sons.

Library of Congress Cataloging in Publication Data:

Sawatzky, Jasper J

Programming in BASIC-PLUS.

Includes index.

1. BASIC-PLUS (Computer program language) I. Chen, Shu-Jen, II. Title.

QA76.73.B3S3 001.64'24 80-27869

ISBN 0-471-07729-1

Printed in the United States of America

10 9 8 7 6 5 4 3

Programming in BASIC-PLUS

Preface

This text was written to enable students to experience the excitement of using a computer. More specifically it is designed to introduce students to computers by learning the BASIC language; it is also designed to provide the essentials for students interested in developing sound programming skills. The text is arranged so that it can be used for a general introductory course or for a more technical applications-oriented program.

Learning the BASIC language is important to the understanding of the logic of computers—a necessity in this computer age. BASIC is used in many practical applications, particularly in small organizations. BASIC has become the most widely used teaching language because it is easy to learn; yet, it is a powerful language. Most of today's low-cost microcomputers use the BASIC language.

This text reflects an effort to provide a serious treatment of a popular version of BASIC—BASIC-PLUS. Sound structured programming principles are stressed so that students planning to continue their study of languages and computer systems will have the background to develop well-organized software.

We have used Digital Equipment's version of BASIC-PLUS and BASIC-PLUS-2 exclusively. BASIC-PLUS is one of the most widely used versions of BASIC in the educational environment. BASIC-PLUS is similar to other versions of BASIC; thus, this text can be used with systems supporting different versions of the language. We have deliberately avoided the approach commonly used in texts which attempt to describe several different versions of BASIC when programs are explained. We have found that students become confused when studying materials that attempt to explain more than one version of a computer language.

The text includes more content than can be covered in a normal term. Thus it provides flexibility to the instructor in selecting topics appropriate for a specific course. Some instructors will find that the first 10 chapters will provide adequate treatment for a course. For courses of a technical nature, Section III, File Editing and Processing, provides content specifically designed to cover these topics in the PDP-11 environment. For the general education or introductory course, Chapter 9, Functions and Subroutines, may be omitted and Chapter 10, Character Strings, can be included since the content of these chapters is independent.

The pedagogical objective of this text is to develop students' programming proficiency in a progressive manner. A minimal mathematical background is assumed. We believe the student should interact with the computer with the first assignment.

Each chapter begins with summary statements of the content to be covered. The format of each statement introduced is listed, followed by a series of examples that illustrate how the statement can be used. A summary program incorporating the features of the statements presented is listed and explained. Terms introduced in the chapter are defined at the end of each chapter. A large selection of problems of varying difficulty are included.

The text is organized into three sections:

Section I provides coverage of a minimum set of BASIC-PLUS instructions. Chapter 1 includes a short introduction to the language and the PRINT, END, and REMARK statements. A special assignment at the end of this chapter provides the necessary practice in logging on and off the system, coding simple PRINT statements, listing and running a program, and saving programs on disk. Students begin the course using the computer terminal immediately.

Chapter 2 explains variables and constants, the assignment statement, and introduces students to loops and counters. Our experience in teaching BASIC is that including simple conditional statements early in the course, as we do in this chapter, is highly motivating to students. Problem assignments can be more realistic when simple conditional statements are included.

Chapter 3 covers the READ, DATA, INPUT, and RESTORE statements. Also included are additional concepts related to conditional statements. Chapter 4 introduces flowchart sym-

bols and flowcharting as a logical tool in problem solving. Conditional statements are explored in greater depth. The FOR. . .NEXT loop structure is introduced.

Section II, Structured Programming and Applications in BASIC-PLUS, covers a broad range of topics but focuses on the importance of structured program design. Chapter 5 provides logical problem-solving approaches to debugging programs so that students understand methods of locating errors, particularly logic errors.

Formatting output approaches in BASIC-PLUS are explained in Chapter 6. Some differences between BASIC-PLUS and other language versions appear in this chapter. Single and double subscripts are explained through numerous examples in Chapter 7. The use of the EXTEND mode for creating meaningful variable names is introduced.

The essentials of structured programming with emphasis on program documentation are thoroughly explained in Chapter 8. Chapter 9 covers both the standard built-in BASIC functions, explains user-defined functions, and develops approaches for using subroutines. Character strings and various built-in functions for manipulating strings are explained in Chapter 10.

Chapters 11 and 12 are technical chapters that presume additional mathematical background and cover topics on matrix manipulation, multiple line functions, subprograms, and error-handling routines.

Section III covers file editing and processing in the BASIC-PLUS environment. This is a particularly valuable section in the text since students are given clear examples on using the Text Editor, file processing, and batch processing.

We thank our teaching colleagues who have provided ideas for improving the content of the manuscript and who have tested some of the teaching approaches in their own classes. Particular appreciation goes to our spouses, Rose and Kang-Shen, for their encouragement and sympathy during the development of this text.

Jasper J. Sawatzky
Shu-Jen Chen

Contents

SECTION I INTRODUCTION TO BASIC-PLUS

Chapter 1 Introduction to Programming	Page
The Meaning of Programming	2
The Concept of the BASIC Language	2
The PRINT and END Statements	4
Using a Computer Terminal	5
The REMARK Statement	6
 Chapter 2 Your First Program in BASIC	
Methods of Correcting Keying Errors	10
Types of Variables and Constants	10
The Assignment Statement—LET	12
The Output Statement—PRINT	15
The Purpose of Loops and Counters	18
Simple Conditional Statements	20
 Chapter 3 Data Entry	
The READ and DATA Statements	24
More About Looping and Conditional Statements	27
The RESTORE Statement	28
The INPUT Statement	28
 Chapter 4 Multiple Decisions and Looping	
Flowchart Symbols and Flowcharting Procedures	34
More About Conditional Statements	36
Logical Operators and Compound Conditions	39
Multiple-Level Decisions	41
Applications of Conditional Statements	42
The FOR . . .NEXT Loop	44
Nested Loops	46

SECTION II STRUCTURED PROGRAMMING AND APPLICATIONS IN BASIC-PLUS

Chapter 5 Debugging Programs	
Syntax and Runtime Errors	56
Use of Simulation Techniques to Find Errors	58
The Value of Inserted PRINT Statements	59
Use of Immediate Mode	61
The STOP Statement and CONTINUE Command	62
 Chapter 6 Formatting Output	
Output Rules for PRINT Statements	66
The PRINT TAB Function	67
The PRINT USING Statement	69
Using Separate Format Statements	72

Chapter 7 Arrays and Tables

Subscripted Variables	78
Searching Arrays	80
The EXTEND Mode	81
Methods of Sorting Arrays	81
Table Organization	84
Storing Data in Tables	85
Retrieving Data from Tables	86

Chapter 8 Structured Programming

The Essentials of Structured Programming	92
Fundamental Program Structures	94
Avoiding GO TO Statements in Sequence Structure	94
USE of IFTHENELSE Statements	95
Use of ONGOTO Statements	97
Building Loop Termination Conditions	100
Use of REM Statements and Other Structured Features	105
The Concept of Top-Down Design	106

Chapter 9 Functions and Subroutines

Use of Built-in Functions	116
Computer Simulation	125
Defining Your Own Function	126
Use of Subroutines	131
More About Top-Down Design	136

Chapter 10 Character Strings

String Arrays	147
The CHANGE Statement and ASCII Function	148
Functions for Extracting Substrings	151
Linking Strings	152
Searching Strings	153
Converting Numeric Strings to Numbers	154
Other String Functions	156

Chapter 11 Matrix Manipulation

The Relationship Between Tables and Matrices	160
Approaches to Loading Values in a Matrix	161
Displaying the Contents of a Matrix	166
Arithmetic Operations of Matrices	167
Functions of a Matrix	174
Solving Simultaneous Equations With Matrices	174

Chapter 12 Advanced BASIC-PLUS Features

Defining Multiple Line Functions	184
Using Subprograms for BASIC-PLUS-2	187
Error Handling Routines	192
Writing an Arithmetic Time-Test Program	193

SECTION III FILE EDITING AND PROCESSING IN BASIC-PLUS

Chapter 13 Using a Text Editor to Prepare Files

Reasons for Using a Text Editor	202
How to Create a File by Using a Text Editor	206
How to Modify a File	207
More Efficient Methods for Correcting Errors	209

Chapter 14 File Processing

Reasons for Using a File	214
Sequential Access Files and Random Access Files	215
Display Files and Binary Files	215
Other Types of Files in BASIC-PLUS	216
Creating, Using, and Updating a formatted ASCII File	218
Creating, Using, and Updating a Virtual Array	225
Handling Large Matrices	230

Chapter 15 Batch Processing Mode

Batch Processing in a Time-Sharing System	236
Setting Up a Job Control File	237
Requesting a Job Run	238
Examining the Status of a Job	239
Obtaining Output from Batch Processing	239

Appendixes

Appendix A BASIC-PLUS Error Messages	247
Appendix B System Commands for the RSTS System	250
Appendix C Text Editor—TECO	252
Appendix D Record I/O	257
Appendix E File Handling with BASIC-PLUS-2	262
Appendix F Reserved Words in BASIC-PLUS	264
Appendix G Selected User Recoverable Error Messages	265
Appendix H How to Use the RSTS System	267

Index	271
-------	-----



Introduction to BASIC-PLUS

Chapter 1

Introduction to Programming

Objectives You will learn:

- The meaning of programming.
- About the BASIC-PLUS language.
- To read a BASIC program using PRINT and END statements.
- To use a computer terminal.
- To use the REMARK statement.

Computers are information processing machines. Computers can be programmed to do mathematical computations, control a flight to the moon, compute your income tax, and perform a wide variety of other tasks. Since the computer is a machine, it must be given detailed instructions in order to perform an information processing task.

1.1 PROGRAMMING

Programming is the process of preparing instructions that can be “understood” by a computer. In other words, a BASIC program is a series of instructions written by humans that can be translated into machine language. These instructions to the computer must be placed in a logical order, since the computer is designed to follow instructions step by step. Instructions to the computer in a program are usually given in the form of statements. These statements must conform to the rules of the computer language that is being used.

1.2 THE BASIC-PLUS LANGUAGE

There are many computer languages that have been developed since computers were invented. These languages have evolved over the years because computer technology has advanced so rapidly that more sophisticated instructions are possible.

BASIC is an acronym for “Beginners All-purpose Symbolic Instruction Code.” BASIC was developed to make computer programming easier to learn. It is considered to be a high-level language, since it does not require technical knowledge about computers. The BASIC-PLUS version evolved from standard Dartmouth BASIC. This book uses a version of BASIC-PLUS introduced in 1971 specifically for the PDP-11 series of computers manufactured by Digital Equipment Corporation. BASIC-PLUS is a powerful version of the BASIC language; yet, even though there are differences in the many forms of BASIC, learning BASIC-PLUS is valuable for using other computers including microcomputers, such as the TRS-80, PET, and APPLE.

You can enter BASIC statements on a terminal connected to a computer where your instructions will be interpreted, and your program will produce the answers you need. The results of your BASIC instructions, called output, will appear on the same terminal on which you entered your program. Two common types of terminals, called I/O devices, are a video-type terminal, the CRT—cathode ray tube, and a typewriter-type device which has paper inserted in it. A common typewriter-type terminal is the DECwriter shown in Figure 1-1. The CRT is more commonly used because it has fewer moving parts and a picture screen where all input and output appears (See Figure 1-2).



Figure 1-1 A Printing Terminal

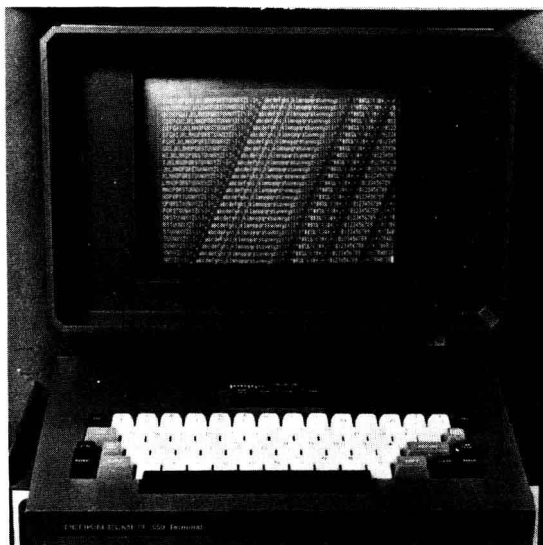


Figure 1-2 A CRT Terminal

As you key in your instructions to the computer on the terminal, your instructions will appear on the screen or paper precisely as you entered them. This enables you, the programmer, to edit and make corrections in the event they are needed. Interaction using a terminal connected to a computer also allows special computer program instructions to signal you when you key in statements that are not “understood” by the computer. Used in this way, BASIC-PLUS is operating in a timesharing mode—that is, you are using the computer to solve problems together with other users. There may be 40 or more other users also using the same computer to solve problems for them at the same time that you are entering instructions on your terminal. The term, timesharing, is used to describe multiple use of a computer designed for this purpose. Special operating and hardware systems handle many users so that one has the illusion that they are the only person using the computer. A timesharing computer can perform operations so rapidly that the user may not experience noticeable pauses or delays in obtaining responses to instructions. When the computer system is being used by large numbers of programmers, however, there may be delays and pauses due to the load placed on the system. Figure 1-3 shows a sketch of an interactive computer system.

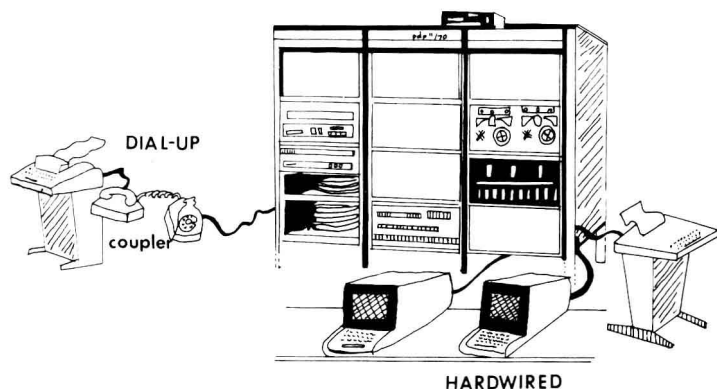


Figure 1-3 An Interactive Timesharing Computer System

4 Introduction to Programming

1.3 A BASIC PROGRAM USING PRINT AND END STATEMENTS

Let's take a look at a short BASIC program and see what it does.

Example 1.1

```
10      PRINT 'A PROGRAM TO COMPUTE 3 VALUES'
20      PRINT 10 + 20, 8 * 9, 100/25
30      PRINT 'PROGRAM BY JOHN K. HENRY'
40      END

A PROGRAM TO COMPUTE 3 VALUES
30          72          4
PROGRAM BY JOHN K. HENRY
```

The numbers for each of the BASIC instructions in Example 1.1 are used to ensure that a logical sequence is followed by the computer when this program is run. Note that there is room for additional statements to be inserted between each statement since statement numbers (commonly called line numbers) are assigned in increments of ten. Line numbers can be assigned any integer between 1 and 32767. The word PRINT is a special word that must be used whenever we wish to have words, numbers, or other symbols printed or appear on the terminal. The use of quotation marks, either single (') or double (""), signals the computer that we want the symbols appearing between the quotes to appear in output exactly as we have arranged that line. Statement 10

```
10      PRINT 'A PROGRAM TO COMPUTE 3 VALUES'
```

produces the output line

```
A PROGRAM TO COMPUTE 3 VALUES
```

Note that the quotes do not appear in output, but they are needed as part of statement 10 in order to have the line printed as it was keyed.

Statement 20 instructs the computer to print the results of the arithmetic operations specified after the word, PRINT,

```
20      PRINT 10 + 20, 8 * 9, 100/25
```

The second line of output in example 1.1 is printed as

```
30          72          4
```

The computer performs the arithmetic operation in statement 20 until it detects a comma; then it combines 10 and 20 for the sum of 30. The second operation results in the multiplication of 8 times 9 (the asterisk is the signal to multiply). The third operation results in 100 being divided by 25 with only the quotient printed. Note that in statement 20 no quotes were placed around the numbers and symbols following PRINT. Had quotes been placed before and after these symbols, they would have appeared exactly as keyed, and no computation would have occurred. Statement 30 functions in the same way as statement 10. Statement 40 is an END statement that terminates further execution of the program; thus, it should be the last statement in the program.

1.4 USING A COMPUTER TERMINAL

Procedures for using computer terminals vary depending on the types of hardware and supporting programs used. The following example illustrates a log-in procedure, a short program, and log-out procedures. The system used is Digital Equipment Corporation's PDP 11/70 (RSTS).¹ After switching on the CRT power switch, press the return key (CR) and type HELLO. Now let's examine the following printout of the steps involved in using a computer terminal and running a program:

Example 1.2

HELLO

RSTS V06C-03 CSU Fullerton Job 11 KB54 11-Jul-80 09:12 AM

#90,199

Password: _____

Summer weekend hours for the Computer Center
are from 10:00am to 6:00pm.

Ready

NEW

New file name--PROB2

Ready

10 REM SAMPLE PROBLEM 2

20 PRINT 'THE VALUE OF 16 CUBED IS'; 16**3

30 END

RUN

PROB2 09:13 AM 11-Jul-80

THE VALUE OF 16 CUBED IS 4096

Ready

BYE

Confirm? F

The underlined words and numbers are keyed in by the user. Note that a # sign appears, and the user is required to supply an account number that has been assigned to him by the instructor. The word PASSWORD appears on the screen, and the user usually is required to furnish a six-character password that has also been assigned to him. Note that when the password is keyed in by the user that it does not appear. This is for the user's security so that other individuals cannot use the password and thereby gain access to another user's programs. The next series of statements shown on the CRT (if any appear) merely are informative and will vary with each installation. When the word READY appears, the user who is creating a new program usually wishes to identify it in some way so the word NEW is keyed in by the user. The user then keys in a file name of up to six characters in length that describes the program in some meaningful way. In this example, the file name is PROB2. After the word READY appears, the user can begin keying in statements.

¹ Log-in procedures vary with installation. Some installations have dial-up terminals which require dialing a number, placing a coupler on the modem, and responding to an inquiry with an account number and a password.

6 Introduction to Programming

1.5 THE REMARK STATEMENT

Statement 10 is a REMARK statement that can be used by the programmer to make explanatory statements in the program that will not appear in the printed output. REMARK is abbreviated by REM. In statement 20, a PRINT statement, a descriptive statement enclosed by quotes is inserted before an arithmetic operation is performed. Once the END statement is keyed in, the program is ready to RUN. The word RUN is keyed in by the user. The computer prints out a header line, indicating the name of the program, the time it was run, and the current date followed by the output generated by statement 20. If the user does not wish to continue use of the terminal, he may log-off by keying in BYE. The word CONFIRM will appear on the screen, and the user can respond by keying in an "F" (Note that there are other responses to CONFIRM that also can be used, such as N. Logoff is terminated, but the user is not logged off, and the system replies READY).

1.6 HOW TO SAVE YOUR PROGRAM

In order to retain your program for future use or revisions, it is necessary to have identified a program with a specific name no longer than six characters. Once you have keyed in a program on a terminal, it is necessary to type the word, SAVE, following the END statement of the program.² In the event you make changes in your program at a later time, be sure to save the revised program statements by typing the word, REPLACE; otherwise, the changes you have made will not be stored on the disk file. Once you have saved a program, it can be retrieved later by typing OLD following log-in procedures.

Word List

BASIC. A programming language that allows the user to give instructions in the form of statements which are executed by the computer.

CRT. An I/O device using a typewriter form of keyboard with a video screen to display data that is keyed into the computer.

I/O. A device that handles both input and output of data.

Interactive system. A computer system connected to some I/O device that provides immediate response when interrogated by a user.

Operating system. Computer programs that control computer hardware and take care of the many details required to run and store programs.

Programming. The process of preparing a logical sequence of instructions that can be followed by a computer.

Statements. Numbered instructions that follow specific requirements of a computer language.

Timesharing. A computer system that handles multiple users on an interactive basis.

INTRODUCTORY ASSIGNMENT

Objectives To become familiar with a CRT Terminal and RSTS/E system commands—NEW, OLD, LIST, RUN, SAVE, CAT, REPLACE, UNSAVE.

² Procedures for saving programs will vary with different systems.

Instructions Following are instructions that will provide you with practice in using some system commands on the RSTS/E system. Print out those statements that have instructions for producing a hard copy.

CR stands for
carriage return

1. LOG ON to the system — Use Local Procedures
2. Type NEW XXX CR (XXX are your initials)
3. Type the following BASIC program:
10 PRINT "THIS IS MY FIRST BASIC PROGRAM"
100 END
4. Type LIST CR (what is the purpose of LIST?)
5. Type RUN CR (what is the purpose of RUN?)
6. Type RUNNH CR (what is different about this command?)
7. Type SAVE CR (note that this saves your program for later use)
8. Type CAT CR (what is the purpose of CAT?)
9. Type NEW TEST CR (TEST is the name of a program)
10. Type LIST CR (Explain why the response to LIST is different from #4 above)
11. Type LISTNH CR (what is the function of NH?)
12. Type OLD XXX CR (remember XXX are your initials used in #2 above)
13. Type LIST CR (explain why the response is different than #10 above)
14. Type 20 PRINT "ZZZZZZ" CR (where the Z's stand for your name)
15. Type LIST CR (has the program changed?)
16. Type 20 PRINT "LLLLL" CR (where the L's stand for only your last name)
17. Type LIST CR (explain the change in your program)
18. Type SAVE CR (do you get an error message? Why?)
19. Type REPLACE CR (this command stores and saves statement 20)
20. Type CAT CR (notice what appears)
21. Print your revised program XXX on the printer by switching on the print key on your terminal and the power on switch on the slave printer.
Type LIST CR
RUN CR
Now switch off the terminal print key by depressing it; remove your output from the printer.
22. Type UNSAVE XXX CR (this erases your program from storage)

8 Introduction to Programming

23. Type CAT CR (explain what happens)
24. Log off system using BYE Y option. Notice how little CPU time you have used as opposed to how much elapsed time (wall clock time) you have used.

Note: See Appendix H for more details on using the RSTS SYSTEM.

PROBLEMS

1. Explain the purpose of statements in the BASIC language.
2. What is the purpose of statement numbers? How should statement numbers be organized?
3. Why is BASIC used as a computer language?
4. What is the key difference between a CRT and a printing terminal?
5. What is the purpose of having a timesharing system?
6. Determine specific procedures you will need to follow to use the computer terminals available to you.
7. Write (code) a program in BASIC that will compute and print the sum of the following values:

$$10 + 15 + 20$$

8. Write additional statements for the program in problem 7 above that will produce output as follows:

PROGRAM BY (your name)

TOTAL IS 45

9. If a terminal is available, try keying in and running the program you coded in problem 8.
10. Write a BASIC program to print a short autobiography; limit the length to no more than 20 statements.