basic BASIC

An Introduction to Programming

Donald M. Monro

```
1Ø FOR I=1 TO 1Ø
2Ø PRINT TAB (3Ø);
3Ø FOR J=1 TO I
4Ø PRINT "*";
5Ø NEXT J
6Ø PRINT
7Ø NEXT I
8Ø END
```

basic

BASIC

An Introduction to Programming

DONALD M. MONRO
Imperial College of Science and Technology
London, England

Edward Arnold

© Donald M. Monro 1978

First published 1978
by Edward Arnold (Publishers) Ltd.
41 Bedford Square, London WC1B 3DP

British Library Cataloguing in Publication Data

Monro, Donald Martin
Basic BASIC,
1. Basic (Computer program language)
I. Title
001.6'424 QA76.73.B3

ISBN 0-7131-2732-5

All Rights Reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of Edward Arnold (Publishers) Ltd.

For Douglas

Reproduced and Printed by photolithography and bound in Great Britain at The Pitman Press, Bath

The Statements of Basic BASIC

Items in square brackets are optional

line number **DATA** *constant, constant*	**Unit 12**
line number **DEF FN***a (variable) = expression*	**Unit 9**
line number **DIM** *subscripted variable, subscripted variable . . .*	**Unit 10**
line number **END**	**Unit 1**
line number **FOR** *variable = expression* **TO** *expression* [**STEP** *expression*]	**Unit 7**
line number **GOSUB** *line number*	**Unit 13**
line number **GO TO** *line number*	**Unit 4**
line number **IF** *relational expression* **THEN** *line number*	**Unit 5**
line number **INPUT** *variable, variable . . .*	**Unit 3**
line number **LET** *variable = expression*	**Unit 3**
line number **NEXT** *variable*	**Unit 7**
line number **ON** *expression* **GO TO** *line number, line number . . .*	**Unit 5**
line number **PRINT** *quantity delimiter quantity . . .*	**Unit 1, 3, 8**
line number **READ** *variable, variable*	**Unit 12**
line number **REM** *any remark or comment*	**Unit 3**
line number **RESTORE**	**Unit 12**
line number **RETURN**	**Unit 13**
line number **STOP**	**Unit 13**

The Statements of Basic BASIC

Items in square brackets are optional

line number **DATA** *constant, constant*	**Unit 12**
line number **DEF FN***a (variable) = expression*	**Unit 9**
line number **DIM** *subscripted variable, subscripted variable . . .*	**Unit 10**
line number **END**	**Unit 1**
line number **FOR** *variable = expression* **TO** *expression* [**STEP** *expression*]	**Unit 7**
line number **GOSUB** *line number*	**Unit 13**
line number **GO TO** *line number*	**Unit 4**
line number **IF** *relational expression* **THEN** *line number*	**Unit 5**
line number **INPUT** *variable, variable . . .*	**Unit 3**
line number **LET** *variable = expression*	**Unit 3**
line number **NEXT** *variable*	**Unit 7**
line number **ON** *expression* **GO TO** *line number, line number . . .*	**Unit 5**
line number **PRINT** *quantity delimiter quantity . . .*	**Unit 1, 3, 8**
line number **READ** *variable, variable*	**Unit 12**
line number **REM** *any remark or comment*	**Unit 3**
line number **RESTORE**	**Unit 12**
line number **RETURN**	**Unit 13**
line number **STOP**	**Unit 13**

试读结束，需要全本PDF请购买 www.ertongbook.com

PREFACE

Computer programming is something that nearly everyone can do, and many people are now finding this out using BASIC, the language designed to help them learn. This little course gives priority to two objectives which I believe are necessary in the learning process. First of all it is fully structured, which means that I intend it to be followed in order using a computer terminal. Secondly I feel that the level of problems and examples must be neither so banal as to insult the intelligence of the reader, nor so advanced as to bewilder him. I recognize that practical computing cannot be totally non-mathematical - after all it is numbers that are dealt with - and so some of the material is related to secondary school mathematics. There are also some instances of numerical computation which I hope will be regarded from a programming point of view rather than a mathematical one. All the material is intended to be approachable.

My earlier text using the same approach at a more advanced mathematical level has been successful with students in higher education and so I have some basis for hoping that this completely new application of these ideas will meet the needs of schools and private citizens.

I have taken account of the GCE syllabus in Computing Science in choosing how much of BASIC to include, in designing the flow-charts and to some extent in choosing examples and problems. Some of the material leans gently towards the pure and applied mathematics syllabuses as well.

D.M. Monro
Totland Bay
Isle of Wight

1978

CONTENTS

INTRODUCTION

1 Computer Languages

Like any language, BASIC is used by people for the communication of ideas. Unlike English, French, or German the intended recipient of a communication in BASIC is a calculating machine which is somehow equipped to accept instructions written in BASIC. There are many computer languages, some intended for specific uses and others which are said to be general, meaning that any computing task could be expressed using them. BASIC is at the same time both specific and general. The name BASIC stands for Beginners All-purpose Symbolic Instruction Code, and the word 'beginners' is the key to its special use as a language for learning the fundamentals of computation in an easily understood form. At the same time BASIC is a very useful general purpose language with some unique features.

All languages have rules of grammar, and in computing these rules must be precise so that no statement of the language has more than one meaning. However, BASIC has a grammar which is intentionally simplified so that only a few simple rules must be learned before real computations can be performed, as will be seen in Unit 1. But because all the essential facilities for computation are present, the emphasis can be placed on the style and methods of computation. When the techniques of computing have been mastered it is easy to convert to the traditional languages of large scale computation because BASIC strongly resembles them.

2 Computer Systems

A computer system is organized around a fast and powerful calculating machine. This machine has no personality or intelligence of its own; anything it does is a result of human instruction. It has a repertoire of simple orders which it obeys slavishly. A series of these orders would be called a computer program; the concept of a program is an easy one for humans to grasp and develops naturally in this course by example. It is important to realize that the machine cannot tell if its very literal interpretation of the programmed instructions make sense. A computer's mistakes are

nearly always the fault of the program.

3 Batch and Interactive Computing

Originally computer systems were organized to deal with one program at a time, and programs were presented in groups or 'batches' which the machine processed one after the other. The programmer submitted his program to a computing service, usually on punched cards, and collected the result some time later. BASIC like any other language can be run in this way, and the majority of computing is still done in batches. The disadvantage for small programs and for learning is that the 'turnaround' time is unlikely to be less than a few hours and could be measured in days. However batch systems are very widely used particularly for production work for reasons of economy.

Interactive computing puts a programmer into direct communication with the computer, usually through a typewriter terminal. He may have the computer to himself or he may be 'timesharing' with others unseen to him. The time taken to submit a program and receive results is reduced to seconds and so program development and error correction are supported in a convenient manner. The learning process is both shortened and made more thorough because the rapid response and the straightforward nature of BASIC encourage experimentation.

4 How to Learn BASIC

First of all, it is essential to be able to run BASIC programs on a computer, ideally by using an interactive system. If only batch processing is available the course can still be followed but the ability to experiment as suggested by some exercises will be reduced. A source of expert advice is required, not so much about BASIC as about the computer system. To learn BASIC simply follow the units in order and do all the exercises because they provide exploration and clarification which are vital to complete understanding. Problems are provided at the end of each unit. Solutions to problems should be worked out on paper before trying them at a terminal. Nothing is more futile than trying to think out solutions at the keyboard; even the most tentative outline can save hours.

UNIT 1
GETTING STARTED IN BASIC

1 Introduction

An advantage of BASIC is that a small amount of information enables it to be used. This Unit introduces the simplest kind of BASIC program and shows how to create, edit, and run it.

2 A Simple Program in BASIC – the PRINT and END statements

A computer program in BASIC is a series of instructions to the computer which has a natural order and is written as statements using familiar English and mathematical terms. The meaning of the program is clear to the programmer and is also precise for the computer.

The following very simple BASIC program has two lines, each stating an instruction from the programmer to the computer:

```
1Ø PRINT 2+2
2Ø END
```

The program instructs the computer to evaluate the expression 2+2 and to print the result. Two important grammatical rules of BASIC are evident in the example:

(i) Each of the two lines begins with a *line number*. These line numbers dictate the order of events when a program is obeyed. Every line must begin with a line number.

(ii) The program ends with an END statement. Every BASIC program must have an END statement as its highest numbered line.

Two different statements of BASIC appear in this simple example, the PRINT statement and the END statement. The intention of the program is clear; when it is obeyed by a computer at line 10 a sum (2+2) is evaluated and printed, and at line 20 the program ends.

The meaning of complicated programs can often be clarified by the use of a flowchart, which shows diagramatically the steps

试读结束，需要全本PDF请购买 www.ertongbook.com

involved in a computer program and their order. This simple program has a simple flowchart as shown in Fig. 1.1.

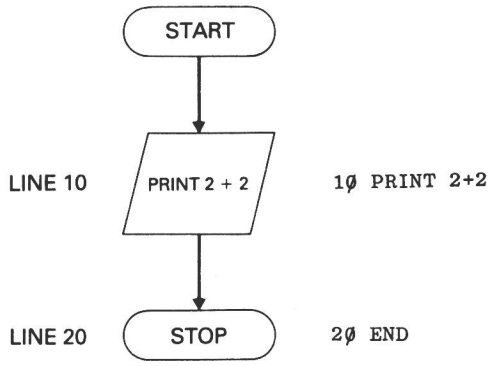

Fig. 1.1. Flowchart of a very simple BASIC program

3 Creating BASIC Programs

The creation of a program consists simply of typing into the terminal the desired lines of BASIC each beginning with a line number. Before this is possible the terminal must be activated and the procedure the computer system has for recognizing permitted users and connecting them to BASIC must be followed. This is where expert help is required the first time.

With the computer system prepared to receive a program, the sample program can be entered by typing it. The keyboard is laid out very much like a typewriter but some of the special symbols such as '+' are in unexpected places and there are some extra keys. Each separate line is entered beginning with its line number and ending with the 'carriage return' or CR key. The first symbols typed must be the line number but after that the number of blank spaces inserted (or left out) is unimportant. The order in which the lines are entered is also unimportant because the line numbers tell BASIC what the real order should be. It is usual to separate the line numbers by 10, since the programmer may later wish to insert extra lines, although any separation is allowed down to consecutive numbers.

EXERCISE With the computer system ready for it, enter the sample program. Be careful that the first symbols in each line are the line number. Do not begin a line with a space, and do not attempt to correct errors at this stage.

4 Listing BASIC Programs – the command LIST

Typing mistakes can easily occur in entering a BASIC program, and errors in transmission between the computer and the terminal are possible. Once a program has been entered it is important to be able to examine it as it is known to the computer. The system command LIST is provided for this purpose, and causes the current version of the program to be printed on the terminal.

EXERCISE Type in the command LIST to obtain a listing of the sample program.

It is important to distinguish between commands like LIST whose first symbol is a letter, and lines of BASIC which begin with a number.

5 Editing BASIC Programs

It is very likely that typing or other mistakes will be made when a program is entered, so that a means of editing is necessary. With BASIC it is possible to change lines, insert new lines, and eliminate unwanted ones. All these procedures are based on the line numbers; when a new line is typed in it becomes part of the program. The procedures are:

(i) To replace or correct a line: type the new line in full and 'carriage return'.

Example: The program reads

```
1Ø PRANT 2+2
2Ø END
```

You type in

```
1Ø PRINT 2+2 and 'carriage return'
```

The corrected program then reads

```
1Ø PRINT 2+2
2Ø END
```

(ii) To insert a line: type in the new line with a suitable line number and 'carriage return'. For example a new line numbered 15 could be inserted between lines 10 and 20 of the sample program.

Example: To the sample program you add a line by typing

```
15 PRINT 5+3
```

The program then reads

```
1Ø PRINT 2+2
15 PRINT 5+3
2Ø END
```

(iii) To eliminate a line: type in only the line number, and 'carriage return'.

Example: To delete the extra line added in (ii) you type

```
15          (just the line number and 'carriage return'
            with no blank spaces)
```

The program then reads

```
1Ø PRINT 2+2
2Ø END
```

Note that to correct a line, the new version has to be typed in full. This is sometimes tedious, so that careful original typing is always worthwhile. Most systems also provide a means of correcting characters in a line while it is being typed. On the keyboard there will be a symbol which deletes the previous one - a letter gobbling key. Usually this is a ← ('back arrow'). An incorrect symbol in any program line or command which is noticed before the line is finished can be removed by typing in enough 'back arrows' to reach the error, and then typing of the line can be continued from the corrected symbol. For example the twice corrected line

```
1Ø PRA←INP 2+2←←←←←T 2+2
```

is the same as

```
1Ø PRINT 2+2
```

The blank between P and 2 counts as a character.

EXERCISE Experiment with the editing facilities to add, change, and delete lines in the sample program. Try correcting errors while typing using 'back arrows'. Finally, restore the program to its given form and list it to be sure.

6 Running BASIC Programs – the command RUN

So far in this unit the sample program has been created, listed, and edited, but it has not been tried. The command RUN is provided to initiate the running of a program. When the RUN command is entered, the computer begins to execute the instructions given by the program. When a program is running the computer has taken control of the session and it is not possible to edit the program or to use other commands until it is finished. The sample program will terminate itself at the END statement.

EXERCISE Run the sample program, by typing in the command RUN.

7 Problems

PROBLEM 1.1 The following BASIC program has been entered. Is it correct?

```
2Ø PRINP 1+3
1Ø END
```

Without using the computer, follow what happens to the program as the following lines are typed in order. (CR) indicates 'carriage return'.

(i) 3Ø END (CR)
(ii) 1Ø (CR)
(iii) 15 (CR)
(iv) 2Ø PRINT 1P3 (CR)
(v) 25 PRINT 5-3 (CR)
(vi) 2Ø PRIYP←←NT 1+3 (CR)
(vii) 2PR←←5 (CR)
(viii) LIST (CR)

Is it now correct? What will happen if the RUN command is entered?

PROBLEM 1.2 Classify the following typed lines as lines of BASIC or commands according to the computer

(i) 1Ø PRINT 3
(ii) LIST
(iii) 2Ø←←END
(iv) RUN←←←3Ø END
(v) PRINT 2+2
(vi) 15 LIST
(vii) 99←←RUN
(viii) LIST←←←4Ø END

Which are correct representations of the facilities described in Unit 1?

UNIT 2
THE ARITHMETIC OF BASIC

1 Introduction

Armed with the ability to prepare simple BASIC programs acquired in Unit 1, it is now possible to introduce the rules of arithmetic as they apply to BASIC. By the end of this unit it will be possible to use BASIC as a programmable calculator for the evaluation of complicated expressions.

2 Addition and Subtraction

The sample program of Unit 1 included the addition facility. An expression including addition is formed simply by placing a plus sign between the numbers to be added. In the expressions considered here the numbers are constants like, 1, 2, 3 etc., and expressions are formed from these. The program

```
1Ø PRINT 2+2
2Ø END
```

contains the *expression* 2+2 formed by the *operation* of addition between the *constants* 2 and 2. A number by itself is an expression as well, such as in

```
1Ø PRINT 35
2Ø END
```

Not surprisingly the operation of subtraction is indicated by a minus sign:

```
1Ø PRINT 4-3
2Ø END
```

Everyone knows that the order of addition is unimportant, that is, 5+1 is the same as 1+5. This is not true for subtraction as 5-1 is not the same as 1-5. This is an obvious example of a fact which will be important, namely that the order of numbers and operators like + and - in an expression is important. In BASIC operations

are done from left to right, and the meaning of a program like

```
10 PRINT 4-3+7
20 END
```

is obvious to the programmer and the computer.

The symbols + and - also have a meaning if they precede a number, for example in

```
10 PRINT -3
20 END
```

This is not true of any other arithmetic symbols.

EXERCISE Ensure that addition and subtraction and their combination is understood, by trying some programs.

3 Multiplication and Division

Multiplication in BASIC is indicated by the symbol * written between the numbers to be multiplied. Like addition, it is of course order-independent.

EXERCISE Try this

```
10 PRINT 4.2*3.7
20 END
```

Here there are *constants* which have decimal points in them. This is always acceptable in BASIC.

In BASIC division occurs when the symbol / is used. Division is order-dependent so that

```
        6/3 is 2
and     3/6 is 0.5
```

EXERCISE Try some divisions. What does BASIC do about decimal points when the result of an integer division is not an integer? What if it is?

EXERCISE Now mix multiplication with division. Do not include addition or subtraction yet.

试读结束，需要全本PDF请购买 www.ertongbook.com