

BASIC

程序设计的理论和习题

[美] B. S. 戈特弗里德 著
农植伟 姚玉洁 张志军 陶思雨 译

高等教育出版社

BASIC 程序设计的 理论和习题

[美] B. S. 戈特弗里德 著
农植伟 姚玉洁 张志军 陶思雨 译

高等教育出版社

SCHAUM'S OUTLINE SERIES
THEORY AND PROBLEMS OF PROGRAMMING WITH BASIC
by
BYRON S. GOTTFRIED
McGRAW-HILL BOOK COMPANY 1975

BASIC 程序设计的理论和习题

[美] B. S. 戈特弗里德 著
农植伟 姚玉洁 张志军 陶思雨 译

高等教育出版社出版
新华书店北京发行所发行
人民教育出版社印刷厂印刷

开本 787×1092 1/32 印张 15 字数 300,000

1981 年 6 月第 1 版, 1984 年 6 月第 2 次印刷

印数 17,501—109,500

书号 15010·0343 定价 1.30 元

译 者 的 话

本书是美国绍姆课程概要丛书中的一本，内容为用BASIC语言进行数字计算机程序设计。全书内容安排从简到繁，循序渐进。上篇在扼要介绍计算机后，即讲述最简单的BASIC程序的编写与运行，然后再讲本语言的其他一些常用特性；下篇则进一步介绍BASIC的一些更专门的特性。本书大部分内容，只要求读者具备高中代数的数学知识。理论讲述比较简要，主要通过大量例题阐明理论及其应用。每章末均有复习题、解答题、补充题和程序设计题，藉以巩固概念、方法并作实际练习。因此，对学习BASIC程序设计的学生和自学者，本书都有一定参考价值。

书中少数例题内容不尽符合我国情况，但其方法仍可参考。

参加本书翻译工作的有姚玉洁(序言和第一、二、三章)，张志军(第四章)，农植伟(第五、六章)，陶思雨(第七、八章)，由农植伟主译并校阅全书。在翻译过程中，对原书个别错误作了订正。由于译者水平有限，错误在所难免，欢迎读者指正。

原 序

在过去二十多年期间研究出的许多种程序设计语言中,没有哪一种比 **BASIC** 语言更容易学。然而, **BASIC** 这种非常简单的语言却包含足够使各方面的人都感兴趣的功能和灵活性。在中等学校和专科学校使用 **BASIC** 语言已很普遍。而且,由于可通过大多数商用分时系统来利用这种语言,使它可更广泛地用于许多商业、技术和科学应用上。

本书讲授如何使用 **BASIC** 语言进行数字计算机程序设计。**BASIC** 语言的全部基本特性都讨论到了。然而,本书的目的在于教给读者怎样编写有效的计算机程序,并强调进行充分的程序设计练习的重要性,本书还介绍 **BASIC** 语言的规则。

有意将本书写成初等的样子,以使得本书对包括从高中学生到从事实际工作的专业人员的广大读者来说都是容易理解的。本书尤其适用于高级中学或初等专科学校,或是作为基本程序设计课程的教科书、分析技术中更综合的课程中的辅助教材,或是作为一种有效的自学入门书。就大部分内容而言,所需的数学知识不超过高级中学代数的水平。

本教材由篇幅大体相等的两篇组成。第一篇——基本 **BASIC**——包括该语言常用的特性。简明的程序设计课程可以只讲这部分内容。第二篇——扩展 **BASIC**——涉及到一些更专门的特性,如子程序、矩阵语句和文件处理等。

这些内容是按能使读者尽快写出虽基本但完整的 **BASIC** 程序这种方式编写的。必须强调的是,读者在读本书的同时,必须自己编写这种程序,并在计算机上执行之。这能大大提高初学程序设计者的自信心,并培养其在此方面的兴趣。(学习计算机程序设计和学习打鼓一样,是不能仅靠读书本就能学会的!)

作为本书的一个组成部分,还包括大量例题。其中有许多综合程序设计问题及通常训练用的练习,另外在多数章后还有一些解答题。读者应象读每章正文内容那样仔细研究这些例题和解答题,并着手编写自己的程序。

各章末还给出复习题、补充题和程序设计题。复习题能帮助读者回忆各章介绍过的内容,并且这些复习题实际上还是各章的小结。多数补充题和程序设计题并不需要专门的数学或科技知识。学生应尽可能多做这些题。(在书末给出补充题的答案。)当把本书作为教科书时,教员最好再补充一些能反映本学科需要的另外布置的程序设计题。

感谢乔安娜·科尔拉夫人,她耐心、辛勤地打印了全部手稿;感谢匹兹堡大学计算中心,它提供了了解大批综合程序设计题所需的计算机时间;还要感谢我的那些早期使用该书手稿的学生们,他们提出了许多宝贵意见。

最后,学完这本书的读者将学到相当多的有关通用计算机程序设计的概念及 **BASIC** 语言的具体规则,他将会完全确信:用 **BASIC** 语言编程序不仅是容易的,而且还是很有意思的。

B.S. 戈特弗里德

目 录

第一篇 基本 BASIC

第一章 概述3	3.3 输入程序39
1.1 计算机的特性3	3.4 校正错误40
1.2 工作方式4	3.5 处理程序41
1.3 BASIC 导论7	3.6 脱机44
第二章 BASIC 语言入门12	3.7 错误诊断44
2.1 数(常量)12	3.8 逻辑调试46
2.2 字符串12	3.9 结束语48
2.3 变量13	第四章 分支和循环51
2.4 算符和公式(表达式)13	4.1 关系符51
2.5 运算的优先级别14	4.2 条件分支——IF-THEN 语句52
2.6 括号的应用14	4.3 多重分支——ON-GO TO 语句56
2.7 有关公式的特定规则15	4.4 STOP 语句57
2.8 赋值——LET 语句.....16	4.5 循环的构成——FOR-TO 语句.....62
2.9 读输入——INPUT 语句.....17	4.6 循环的封闭——NEXT 语句.....63
2.10 打印输出——PRINT 语句18	4.7 嵌套循环67
2.11 END 语句22	第五章 BASIC 的一些其他特性80
2.12 编写完整的 BASIC 程序22	5.1 库存函数80
2.13 程序说明——REM 语句23	5.2 表和表格(阵列)83
2.14 转移控制——GO TO 语句24	5.3 下标变量84
2.15 重复执行程序24	5.4 定义阵列——DIM 语句87
2.16 结束语25	5.5 送进输入数据——READ 和 DATA 语句88
第三章 BASIC 程序运行36	5.6 重读数据——RESTORE 语句.....96
3.1 分时终端36	5.7 结束语102
3.2 请求联机36	

第二篇 扩展 BASIC

第六章 函数和子程序111	6.6 产生随机数——RND 语句.....124
6.1 定义函数——DEF 语句111	6.7 RANDOMIZE 语句125
6.2 引用函数111	6.8 定义子程序129
6.3 多行函数117	6.9 引用子程序——GOSUB 语句130
6.4 数据编码与译码——CHANGE 语句118	6.10 图形输出138
6.5 ASC 函数与 CHR\$ 函数.....120	第七章 向量和矩阵156
	7.1 向量和矩阵运算156

7.2 向量和矩阵的输入和输出	159	8.3 运算期间的文件规定	205
7.3 特殊矩阵	168	附录 A BASIC 语句总表	215
7.4 改变行列数	174	附录 B BASIC 库存函数	217
第八章 数据文件	190	附录 C BASIC 系统指令	218
8.1 顺序数据文件	190	部分补充题的答案	219
8.2 随机数据文件	199		

完整程序的例题

1. 圆的面积和周长(例 1.6)(参看例 3.8, 41 页)	8
2. 二次方程的根(例 2.26 和 2.30)	22, 24
3. 方程的根(例 4.5)	53
4. 计算折旧(例 4.9)	57
5. 求大气污染数据的平均值(例 4.16)	64
6. 斐波纳契数的产生和素数的检索(例 4.18)	68
7. 一个函数表(例 5.5)	82
8. 破译密字码(例 5.9)	84
9. 重排数表(例 5.14)	90
10. 表格处理(例 5.15)	92
11. 求最大值(例 6.6)	113
12. 产生切口隐语的程序(例 6.15)	121
13. 概率游戏(掷双骰)(例 6.20)	126
14. 月工资表(例 6.26)	132
15. 模拟球的弹跳(例 6.28)	140
16. 矩阵运算(例 7.15)	166
17. 联立方程(例 7.19)	171
18. 最小平方曲线拟合(股票市场交易)(例 7.22)	175
19. 学生考试分数的处理(例 8.4)	194
20. 库存控制(例 8.10)	202
21. 数据文件的寻找(例 8.13)	205

第 一 篇

基 本 BASIC

第一章 概 述

本书讲述的是用 BASIC(Beginner's All-purpose Symbolic Instruction Code 初学者通用符号指令代码)程序语言进行计算机程序设计。在本书中可以看到,怎样对原来用语言描述的问题进行分析、怎样概述其要点、最后怎样把此问题变成可供使用的 BASIC 程序。下面将用许多实例来详细阐述这些概念。

1.1 计算机的特性

什么是数字式计算机?它是怎样工作的?

计算机是信息处理机。为使它能工作,对它要完成的每个特定任务,都必须给出一组特有的指令,这种指令序列称为程序。程序存储在计算机的内存储器中,需要存储多久就可存多久。一旦一个完整的程序存入存储器以后,就可执行它,产生一系列要完成的操作。

通常,执行计算机程序可做下列这些事:

1. 把被称为输入数据的所需信息读入计算机,并存入计算机存储器的其他(不存程序的)区内。
2. 处理输入数据以产生需要的结果,这种结果称为输出数据。
3. 最后,把输出数据(也许还有部分输入数据)打印在一张纸上。

■ 例 1.1

假如要用计算机计算半径为 r 的圆的面积。计算公式为

$$A = \pi r^2$$

计算步骤如下:

1. 读半径的数值。
2. 利用上述公式计算面积的值。
3. 打印半径及相应的面积的值。
4. 停止。

每一步骤都可以和 BASIC 程序中的一条指令相对应。当整个程序都执行完了时,就可以从存储器中清除它,也可以继续保存它,并再次对其他半径值进行计算。

可以用图形来表示此整个过程,如图 1.1 所示。这种图称为流程图。流程图能用来帮助读者把程序的逻辑流程形象化。



图 1.1

在下一节再来研究所谓“读”和“打印”是什么意思。现在先注意两个重要的计算机概念:计

算机存储器及其可编程序的能力。现代计算机存储器的容量范围从几千字到几十万字。这里, 一个字可以表示一条指令、一个数值或一组字符。计算机存储器的容量大小常用 $2^{10}=1024$ 字的倍数来表示, 2^{10} 这个值称为 1K。例如, 一个带容量为 48K 存储器的计算机就有 $48 \times 1024 = 49152$ 字。因此, 即使一个小计算机, 其存储器内也能存储大量信息。此外, 对一给定的计算机, 能编写无数的各种程序, 这是使计算机具有通用性的根本原因。

计算机的另一个特点是它的速度极高。象两数相加这样简单的任务, 在几分之一微秒内就可以完成(1 微秒简写成 $1\mu s$, 等于 10^{-6} 秒或百万分之一秒)。按实用情况来说, 只需花约一美元的费用, 就能在几秒的计算机时间内处理完一个规模较大的大学全部学生的期末成绩。

1.2 工作方式

使用数字式计算机有两种不同的方式, 即成批方式和分时方式。两种方式都常用。每种方式各对某类问题有其自己的优点。

成批处理

在成批处理中, 读入计算机的是许多作业, 在计算机内按次序处理这些作业(所谓一个作业, 指的是一个计算机程序和一系列要处理的输入数据组)。通常, 程序和数据是记录在穿孔卡片上的(在穿孔卡片上, 指令或数据项中的每个字符和数值中的每位数字, 都是用一组编码的孔来表示的。每组孔在穿孔卡片上占一列, 一张卡片一般有 80 列。因此, 在一张穿孔卡片上能记录 80 个字符或 80 位数字)。每个作业是通过机械读卡片装置读入计算机的, 而输出是用专用打字机打印出来的。

在成批处理中, 大量信息(程序和数据)能非常迅速地送入计算机或从计算机输出。因而这种工作方式十分适用于需要大量计算机时间的或本身很长的作业。另一方面, 即使处理一个作业本身所需的时间可能仅要一、两秒有效计算机时间, 但用成批处理方式处理该作业, 通常也得需要几小时到一、两天的时间(这是因为必须等待依次轮到它时, 才能读入它、处理它, 并把结果打印出来)。因此, 当需要处理一个简单的作业并要求尽快地报告结果时, 成批处理方式就会不方便了。

■ 例 1.2

某学生有 1000 个不同的圆半径值。他想对每个半径值计算圆面积。为此, 他必须把例 1.1

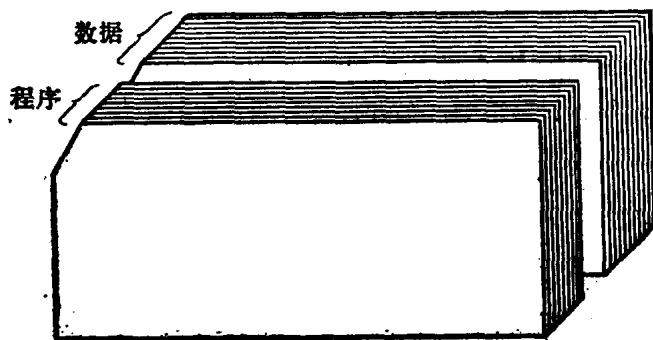


图 1.2

中所述的那样的计算机程序执行 1000 次, 每次对一个半径值。他将用成批处理方式, 因为, 用这种方式可以快速、连续不断地进行全部计算。

为了处理数据, 该学生把一组卡片读入计算机。卡片组的第一部分包含程序, 其中每张卡片上一条指令; 程序后面是 1000 张数据卡片, 每张上有一个半径值。卡片组如图 1.2 所示。

把一组卡片读入计算机后, 要过几个小时之后才执行程序和处理输入数据。计算完之后, 结果打印在一大张纸上。此时, 该学生将得到一张包含 1000 对数的表。每对数表示的是一个半径值和其相应的面积值。

分时

分时就是由几个不同的用户同时使用一个计算机。每个用户都能通过打字机终端和计算机进行联系, 而打字机终端可以用电话线或微波电路连接到计算机上(一种常用的打字机终端如图 1.3 所示)。通常, 终端离计算机很远——也许在几百英里外的地方。由于所有的终端能基本上同时使用计算机, 所以每个用户觉察不到还有别的用户, 就好像整个计算机都由他自己支配一样。

单个的打字机终端既可用做输入设备, 又可用做输出设备。通过键盘把程序和输入数据打入计算机; 把输出数据从计算机送到打字机终端, 在打字机上打印输出。然而, 在打字机终端与计算机之间传送输入或输出数据, 比计算机处理数据要慢得多。处理如用 1 秒, 传送数据可能需 5 分(在成批处理时, 用卡片读出机和打印机传送数据, 这比用打字机终端传送数据快得多)。速度上的这种相对差异, 就使得一个计算机可以同时和几个打字机终端接口。

分时工作方式最适用于处理相对简单的作业, 这类作业不需要传送大批数据或不需要很长计算机时间。中小学校和商业部门中的许多应用计算机的问题就有这个特点。利用分时处理这类应用问题, 迅速、容易, 并且费用最少。

■ 例 1.3

某郊区中学有一套计算机分时设备, 由三个打字机终端组成。这些终端通过电话线连接到附近一所大学的大计算机上。每个终端向计算机传入数据或从计算机传出数据的最高速度是每秒 10 个字符。虽然这三个终端是和一个计算机接口, 但却能同时使用它们。



图 1.3

另外还有 39 个终端连接到该计算机上, 其中 15 个放置在本区域内的另五所中学内, 余下的 24 个配置在该大学本校内各个地方。全部 42 个终端都能 (并且经常是) 被同时使用。因而单个计算机是能够供一个相当大的市区内的几个教育机构作计算使用的。用这种方式共用计算机, 每个学校都能以合理的费用充分利用大计算机。

■ 例 1.4

某个学生写了一个 BASIC 程序, 他想用分时方式执行该程序。为此, 他必须把他的程序打入打字机终端, 一次一条指令, 每条指令打在单独的一行上。指令被送入计算机, 指令一打毕, 就立即存入计算机存储器内。

在把整个程序打完并存入存储器内以后, 该学生只要简单地把一个字 RUN 打入打字机终端, 就可以执行他的程序了。这样就使得能读入输入数据、计算出输出数据, 并将计算结果直接送到打字机终端。

分时方式的一个重要特性是, 在执行程序期间, 计算机和用户能交互作用。即可先执行一小部分程序, 打印出一个信息, 以此信息指明已计算出的初步结果, 并向用户要求补充数据。在供给补充数据之前, 将停止往下执行程序。

在供给所需数据时, 用户可能受早先计算出来的初始结果的影响。例如, 用户可能想在供给补充数据之前先研究输出。此外, 用户供给的某个特殊数据也可能影响执行其余程序的方式。因而, 在某种意义上说, 计算机和用户是彼此可以对话的, 因为由这两个参与者之一供给的信息能影响另一个的下一步操作。

利用这种概念的程序被认为是用对话方式写的。象成三棋、西洋跳棋和国际象棋这样一些已计算机化的比赛, 就是很好的交互式(对话式)程序的例子。

■ 例 1.5

某学生想用分时系统计算已知面积为 100 的圆的半径。可以利用给定半径求圆面积的交互式程序 (注意, 这刚好与学生想求的相反)。因此该学生将用试凑法来处理, 即先假设一个半径值, 然后确定它是否与给定的圆面积相对应, 若不对应, 该学生再假设另一个半径值, 再计算新的面积, 依此类推。这种计算一直要进行到学生求出其相应的面积充分接近 100 的半径值为止。

学生用打字机终端与计算机联系。只要联系一建立并开始执行程序, 就在打字机终端打印出信息

RADIUS=?

然后学生输入一个半径值。现假设他输入的半径值是 5, 打字机终端响应打出

AREA=78.5398

DO YOU WISH TO REPEAT THE CALCULATION?

然后学生打入 YES 或 NO。如果打入 YES, 打字机终端将再打印出信息

RADIUS=?

再重复整个过程; 如果该学生打入 NO, 则在打字机终端上打印出

GOODBYE

便自动地断开和计算机的联系。

在图 1.4 上, 可看到利用上述程序在一个典型的分时期间内打印出来的信息。下面划线的是学生打入的信息。只计算了三次, 就已确定出近似的半径值 $r=5.6$ 。

```
RADIUS=? 5
AREA= 78.5398

DO YOU WISH TO REPEAT THE CALCULATION? YES

RADIUS=? 5
AREA= 113.097

DO YOU WISH TO REPEAT THE CALCULATION? YES

RADIUS=? 5.6
AREA= 98.5204

DO YOU WISH TO REPEAT THE CALCULATION? NO

GOODBYE
```

图 1.4

请注意学生和计算机相互对话的方式。还应注意到, 学生要等到看见计算出的面积之后, 才能判断是否需要再进行另一次计算。如果需要, 学生供给什么样的新半径值是受前一次计算结果影响的。

1.3 BASIC 导论

为给数字式计算机编制程序, 已研究出了许多不同种类的语言。这些语言在难易程度、通用性和预期的用途诸方面的差异相当大。

BASIC 是数字式计算机的一种程序语言, 它的指令很象初等代数公式, 只是增加了一些英文字, 象 LET、GO TO、READ、PRINT、IF、THEN 等等。因此, 和某些其他较隐含的、因而也较难学难用的程序语言比较起来, BASIC 是一种简单的“面向人”的语言。此外, 许多人发现 BASIC 程序是非常有趣的事, 就象有些人解纵横组字谜那样, 可以享受到无穷乐趣。

由于 BASIC 语言和初等代数类似, 它特别适用于解自然科学、数学和工程问题。当然, 这种语言的应用决不是仅限于这些领域。BASIC 还可应用于商业、经济学、心理学、医学、图书馆科学中的各种各样的问题——实际上, 凡需要大量处理数值数据或字符信息的领域, 都可使用 BASIC (所谓字符信息是指非数字符号、字母、单词等等)。在本书的许多程序例子中, 将可以看到 BASIC 的各种初步应用。

BASIC 最初是在二十世纪六十年代中期由约翰·凯梅尼 (John Kemeny) 和托马斯·库尔兹 (Thomas Kurtz) 在达特默思 (Dartmouth) 大学提出来的。由于这种语言简单明瞭, 引起了许多商业性分时服务机构的注意, 不多久就有几家采用了 BASIC 供用户使用。而现在绝大多数分时服务机构都用某种 BASIC 文本, 但是, 在各种 BASIC 文本中, 在比较高级的性能上略有差异。不过, 由于 BASIC 语言已充分标准化了, 以使得学过一种 BASIC 语言文本 (如象本书介绍的这种) 的任何人在适应任何别种 BASIC 语言文本时, 将没有多大困难。

不出所料,通常是把 BASIC 看成分时制语言的。事实上,它是现时可应用的所有分时制语言中应用最广泛的一种语言。本书中,将从分时的观点来介绍这种语言。但是读者应了解,BASIC 也是一种有效的成批处理语言。因此它的用途是对成批处理及分时应用两种情况来考虑的。

BASIC 程序的结构

BASIC 程序中的每一条指令都写成单独的一个语句。因此,一个完整的 BASIC 程序是由一系列语句组成的。除非特意标明“转移”(即控制的转移)之外,这些语句必须按它们将被执行的顺序出现。

下列规则适用于所有的 BASIC 语句:

1. 每个语句必须出现在单独的一行上^①。
2. 语句在长度上不得超出一行(即,不能从一行继续写到下一行)。
3. 每个语句必须以一个正整数开始,该正整数称为语句标号(或行号)。任两个语句都不能具有相同的语句标号。
4. 连续的语句必须有递增的语句标号。
5. 每个语句标号后面必须有 BASIC 关键字,这些字指明要完成的指令的类型。
6. 为使语句更清晰,在需要的地方可以加空格。

在 BASIC 程序中还可能有空行,但每个空行也必须有单独的行号,且行号后至少要有一个空格。(如果行号后紧跟着有回车号,则将从程序中删去这一行。在第三章再进一步讨论这问题。)

■ 例 1.6 圆的面积

图 1.5 介绍一个 给定半径求圆面积的简单 BASIC 程序。为进行计算所用的逻辑早已在例 1.1 中讨论过。然而,这个程序是这样简单,只要简单观察一下就能确定它的逻辑基础。

```
10 INPUT R
20 LET A=3.14159*R↑2
30 PRINT R,A
40 END
```

图 1.5

可以看到,该程序由四个语句组成,每个语句出现在单独的一行。每个语句有自己的语句标号(或行号),从程序的开头(顶上)到末尾(底下),语句标号是递增的。各语句中分别包含 INPUT、LET、PRINT 和 END 等 BASIC 的关键字。

第一个语句(INPUT)的作用是把半径(R)的数值从打字机终端送入计算机;第二个语句(LET)是计算 πR^2 的值,这个值用字母 A 来代表;第三个语句(PRINT)是把 R、A 的数值传送到打字机终端,并在那里打印出来;最后,还需要最后一个语句(END),以指明整个程序的结束。

注意,在行号为 20 的语句中的符号是用来表示算术运算的。乘法用星号(*)来表示;垂直的箭头(↑)则用来将数值自乘(这种运算称为取幂);其他的算术运算,即加、减和除,在 BASIC 中则分别用加号(+)、减号(-)和斜划(/)来表示。

BASIC 程序的执行

只要已写完一个 BASIC 程序,它就能在许多类型和牌号都很不相同的计算机上运行。除

^① 大多数打字机终端上,一行等于 72 个字符,而有些打字机终端允许每行有 132 个字符。

了在各种 BASIC 语言文本之间可能稍有些次要差别之外, 这种语言是与机器无关的。这也就是 BASIC 被广泛采用的一个重要因素。

这种语言与机器无关的原因, 是由于 BASIC 程序实际上分两个步骤来执行。第一步叫做编译过程, 它是把 BASIC 语句转换成计算机自己的机器语言指令系列(机器语言由大量的、十分详细的指令组成。这些指令是和计算机内部的电路有关的。每个计算机都有自己的机器语言。用机器语言编程序比用象 BASIC 这样的高级语言编程序要困难得多。另外, 在型号不同的计算机之间, 机器语言程序不能相互转换)。因此, BASIC 程序称为源程序。在第二步中, 执行已产生的机器语言程序, 这种语言程序称为目标程序, 执行程序也就是按所要求的方式对输入数据进行处理。

通常, 目标程序在被编译成之后立即就被执行。程序员并不能觉察到两个截然分开的步骤, 因为他看到的仅仅是源程序、输入数据和计算出的输出数据。

BASIC 的一些优点

1. BASIC 是“面向人”的。很容易学习, 使用起来也有趣。任何健全的人都能学习用 BASIC 编程序, 并不需要广泛的数学基础知识。
2. 这种语言灵活, 可使程序员毫不费力地修改已有的程序。
3. BASIC 非常适用于分时系统, 而分时系统可使程序员以最少的费用使用大计算机。
4. 除了在各种 BASIC 文本之间有些次要差别之外, 语言是与机器无关的。因此 BASIC 程序能在很多不同的计算机上运行。

复 习 题

- 1.1 能把数字式计算机看做是什么机器?
- 1.2 所谓计算机程序是指什么意思?
- 1.3 当执行计算机程序时, 一般会出现什么情况?
- 1.4 什么是计算机存储器? 哪些类信息存在计算机存储器内?
- 1.5 用什么术语来描述计算机存储器的容量? 一些典型的存储器的容量是多大?
- 1.6 用什么时间单位来表示计算机完成基本任务所用的速度?
- 1.7 所谓成批处理和分时是指什么意思? 叙述每种的特点。
- 1.8 从哪种意义上说计算机和用户是交互作用的? 所谓对话式程序是什么意思?
- 1.9 BASIC 这个词意指什么? BASIC 语言的一般特性是什么?
- 1.10 什么是 BASIC 语句? 在 BASIC 程序中, 语句必须按什么次序出现?
- 1.11 概述适用于所有 BASIC 语句的六条规则。
- 1.12 在 BASIC 中, 用什么样的符号来表示加、减、乘和除?
- 1.13 所谓取幂是指什么意思? 在 BASIC 程序中, 用什么符号来表示取幂?
- 1.14 所谓 BASIC 程序的编译是什么意思? 什么是源程序? 什么是目标程序? 这些概念为什么重要?
- 1.15 概述 BASIC 语言的基本特点。

解 答 题

- 1.16 下面介绍几个初级的 BASIC 程序。试解释每个程序的用途。


```
(a) 10 INPUT L, W
    20 LET A=L*W
    30 PRINT L, W, A
    40 END
```

已知长和宽, 计算矩形的面积。

```
(b) 10 INPUT A, B, C, D, E
    20 LET S=A+B+C+D+E
    30 PRINT A, B, C, D, E
    40 PRINT S
    50 END
```

计算五个数的和。注意, 把五个数打印在一行上, 而把计算出的和打印在下一行上(每个 PRINT 语句从新的一行开始)。

```
(c) 10 INPUT A, B, C
    20 LET X1=(-B+(B↑2-4*A*C)↑.5)/(2*A)
    30 LET X2=(-B-(B↑2-4*A*C)↑.5)/(2*A)
    40 PRINT A, B, C, X1, X2
    50 END
```

根据计算公式

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

$$x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

计算 x_1, x_2 的值, 其中 a, b 和 c 的值已给定。

1.17 对下述每种情况写出一个初级的 BASIC 程序。

(a) 已知圆的面积, 求半径(参看例 1.6)。

因为 $A = \pi r^2$, 可以解出 r 为

$$r = \sqrt{A/\pi}$$

因而所求的程序是

```
10 INPUT A
20 LET R=(A/3.141593)↑.5
30 PRINT A, R
40 END
```

(b) 已知矩形的面积和宽, 求矩形的长[参看题 1.16(a)]。

```
10 INPUT A, W
20 LET L=A/W
30 PRINT A, W, L
40 END
```

(c) 计算五个已知数的积[参看题 1.16(b)]。

```
10 INPUT A, B, C, D, E
20 LET P=A*B*C*D*E
30 PRINT A, B, C, D, E
40 PRINT P
50 END
```