

BASIC语言

丰富多彩的88例

〔西德〕克劳斯·门采尔 著 纪有奎 徐震宇 译 漆永新 王郑南 校

人民邮电出版社

BASIC 语言

丰富多彩的88例

[西德] 克劳斯·门采尔 著

纪有奎 徐震宇 译 漆永新 王郑南 校

人民邮电出版社

译者序

目前,我国虽已出版了不少有关BASIC的初级读物,但尚未见一本把BASIC的基本知识和程序集融于一体的书。本书巧妙地将两者结合起来,使初学者能够学以致用、深刻理解,也可使有基础的读者受到启迪。

全书共六章,分两大部分。前五章介绍BASIC语言的基本知识,依次叙述了BASIC程序的基本结构、该语言的基本成分、文字处理和扩展BASIC等。其特点是重点突出,取材精炼。第六章为程序集部分,是全书的重点。该程序集原有100个例题,中译本精选了其中富于知识性和趣味性的丰富多彩的88个。这些例题既包含小学的四则运算、中学的代数运算等较浅的内容,又有级数、数论、数制转换、排序、数值逼近等较深一些的知识。此外,还包含一部分引人入胜的游戏程序、模拟程序和其他应用实例。为了便于读者自学和理解,作者对每个例题都给出了详尽的问题描述、求解方法、说明、程序清单和运行结果。每道例题之后还给出了一道习题,便于读者举一反三来加深理解。

门采尔教授长期从事计算机教育工作,具有丰富的教学经验。他1981年在德意志联邦共和国出版的这一书,受到了普遍的欢迎。译者特在此将其译出,深信会得到我国读者的喜爱。

书中的大部分程序,曾由中国科技大学计算机系冀翊同志协助调试,并改正了原书中的某些错误。有关的专有名词请教了德意志联邦共和国专家乌塔·埃特尔(Uta·Ertl)。对于他们的热心帮助,译者在此一并表示感谢。由于水平有限,错误之处难免,欢迎读者批评指正。

1986.7

前 言

为什么要写一本关于 BASIC 的书呢？

今天，凡是想与遍及全球的台式计算机上的 BASIC 语言打交道的人，都会找到大量可供参考的德文文献。既然如此，那么为什么还要写一本关于 BASIC 的书呢？

这是因为，使用 BASIC 就如同开汽车一样需要实践！而理论则是开车和使用 BASIC 时避免事故的前提。

有关 BASIC 的大多数入门书都侧重于这种程序设计语言的理论描述，所举的例子有多有少，但仅仅是理论描述的点缀。本书的出发点与之不同，主要部分是取材于众多应用领域的 100 个例题^①，每个例题都附有完整的 BASIC 程序，从而使理论叙述非常紧凑，书中有一些内容甚至直接从实例开始介绍。

为了迅速入 BASIC 之门，读者不妨按下述办法一试。

可以花几小时上机练习，以便掌握 BASIC 的要点。当然，进一步的深化需要时间和经验。人们只有通过实例才能获得经验。

经历过最初困难的初学者，很快会对 BASIC 的有趣应用产生强烈的兴趣。本书可以做到这一点，书中的 100 题，除常用的初等数学内容外，还包括许多游戏以及日常练习。

如果 BASIC 用户能独立地阐明他的题目并开发他自己的程序，那是最好不过了。但是，要做到这一点，他必须从理论上弄通全部问题，并做到无错误地设计程序。而对初学者来说，这是要花时间的。久而久之，也许就使他对学习使用台式计算机完全扫兴了。

① 中译本仅选择了其中的 88 个例题。——译者注

书中提供的例子和程序并不能代替独立的思考和解答问题，而只不过是解决各种具体问题起到抛砖引玉的作用而已。

直接采用书中给出的程序例题可能会产生什么结果呢？

1) 如果你的问题恰好就是书中的具体例题，那么，你可以不必花大量的时间进行理论推导而直接利用答案。

2) 如果你不在于为某个答案所吸引，而是对解题方法感兴趣，那么，你可以借助该程序来试验，从而研究并改进这种方法。

3) 除解题方法外，如果你还对程序设计技巧感兴趣，那你可以改进这些程序。但是，不要把希望寄托在尽量节省 BASIC 语句上，而应想办法使程序变得更“精巧”，也就是说，寻找最简单的解题方法。

有些读者会觉得，本书中的解题方法以及各个例子所用的技巧缺乏理论基础。出于篇幅的原因，本书无法满足这些读者的愿望，他们必须求助于有关的专业文献。我们着重考虑的是提供尽可能广泛的应用。要指出的是，除了纯粹的解答公式的程序设计之外，解题方法往往也可以根据程序清单予以再现。为此，读者最好把 BASIC 程序全部或部分译成框图来研究解题方法。

6.1 节到 6.3 节给出了程序实例。

欢迎读者指出本书逻辑思维、事实陈述、程序设计以及印刷上的错误。

克劳斯·门采尔

1981 年春

目 录

第一章 导言	1
第二章 BASIC程序的结构	3
2.1 基本结构	3
2.2 五个基本成分	5
2.3 框图	7
2.4 标准函数	9
2.5 控制命令	11
第三章 BASIC的基本成分	14
3.1 输出(PRINT)	14
3.2 输入(INPUT, GET, READ)	16
3.3 赋值(=, LET)	18
3.4 转移(IF...THEN, GOTO)	20
3.5 循环(FOR/NEXT)	21
第四章 BASIC中的文字处理	23
4.1 字符串	23
4.2 输出/输入	24
4.3 赋值	25
4.4 转移	27
4.5 加工	28
第五章 扩展BASIC	31
5.1 数组	31
5.2 子程序	32
5.3 转移表	34
5.4 自定义函数	35

5.5 逻辑表达式	36
-----------------	----

第六章 BASIC 88 例	38
-----------------------------	-----------

6.1 数学例题	39
----------------	----

[例 1] 数的整除	39
[例 2] 用 INT 函数判断可整除性	41
[例 3] 带余数的除法	42
[例 4] 数的四舍五入	44
[例 5] 数的约数	45
[例 6] 某数的约数集合	47
[例 7] N 是素数吗?	48
[例 8] 求 M 到 N 之间的素数	50
[例 9] 分解成素数因子	51
[例 10] 欧几里得算法	53
[例 11] 最大公约数	54
[例 12] 最小公倍数	56
[例 13] 分数化简	57
[例 14] 分数化小数	59
[例 15] 求 M 到 N 之间的完全数	60
[例 16] 求 M 到 N 之间差值为 2 的一对相邻素数	61
[例 17] 求某数各位数字之和	63
[例 18] 求多位数字总和的规律	64
[例 19] 把十进制数转换成 G 进制数	65
[例 20] 将自然数 N 转换成基数为 G 的形式	67
[例 21] 把 G 进制数转换成十进制数	68
[例 22] 把十进制小数转换成 G 进制小数	70
[例 23] 求多项式的值	71
[例 24] 求多项式的根	74
[例 25] 求一元二次方程的根	74
[例 26] 三次方程	75
[例 27] 二元一次线性方程组	77
[例 28] 求平方根的近似值	79

[例 29]	用区间夹套法求平方根	81
[例 30]	插值法	83
[例 31]	调和级数	84
[例 32]	π 的近似值	85
6.2	游戏和模拟	88
[例 33]	掷骰子(取 1 ~ N 间的整数)	88
[例 34]	掷骰子试验	89
[例 35]	数字彩票	91
[例 36]	俄罗斯轮盘赌	92
[例 37]	超人智力	94
[例 38]	17 加 4 纸牌游戏	97
[例 39]	取火柴游戏	101
[例 40]	占方游戏	104
[例 41]	猜数	107
[例 42]	小星星	109
[例 43]	区间寻找法	110
[例 44]	捉小精灵	113
[例 45]	字母排序游戏	116
[例 46]	报数出列	118
[例 47]	轮盘赌	119
[例 48]	九柱戏(保龄球)	123
[例 49]	记字母	127
[例 50]	找数字	129
[例 51]	难题	133
[例 52]	掷双么游戏	136
[例 53]	击沉舰队	139
[例 54]	奇还是偶?	142
[例 55]	Z 大夫	145
[例 56]	不参与经营的股东	148
[例 57]	连珠棋	151
[例 58]	插棍游戏	154
[例 59]	阿瓦利——古老的非洲游戏	157

[例 60]	放射性碎片	160
[例 61]	登月飞行	163
[例 62]	看谁被吊起来	170
[例 63]	八皇后问题	173
[例 64]	图案设计	177
[例 65]	梵塔	180
6.3	其他实例	185
[例 66]	表中最小元素	185
[例 67]	排序和查找	187
[例 68]	气泡法排序	190
[例 69]	交换排序	193
[例 70]	快速排序	195
[例 71]	两个表的合并	198
[例 72]	平均值和方差	201
[例 73]	回归和相关	202
[例 74]	降雨(随机事件发生器)	204
[例 75]	在立方体上迷路	208
[例 76]	迷宫里的老鼠	211
[例 77]	生日	215
[例 78]	逆波兰表示法	216
[例 79]	按德·翁德方法分配席位	219
[例 80]	追捕	221
[例 81]	设计食谱	224
[例 82]	高效物质或有害物质的浓缩	227
[例 83]	命中目标	229
[例 84]	光电效应	230
[例 85]	元素周期表	233
[例 86]	用爱因斯坦系数求孪生晶体的参数	234
[例 87]	兔子和狐狸随时间的演变规律	236
[例 88]	生命演化	238

BASIC 语句和标准函数在例题中的应用索引	247
------------------------------	-----

第一章 导言

为什么偏偏选中了 BASIC 呢？

BASIC 是世界上最流行的程序设计语言，它不仅用来作中小学教材，而且在迅速增长的家用电脑和业余爱好者的计算机上得到了广泛的应用。

拥有一台配备了标准 BASIC 语言的家用电脑的人，实际上并不存在语言问题。原则上说，他可以从一个台式仿真计算机或邻居的台式计算机上运行 BASIC 程序，过渡到在他自己的计算机上运行程序。这是 BASIC 和其他“面向问题”的语言所共有的一个优点。

仅仅这一点还不能说明 BASIC 高度普及的原因，更重要的还在于，使用 BASIC 语言的计算机费用相当低，并且只要很少的存储容量就够用了。还有一点就是 BASIC 的通用性。一般来说，任何算法的答案都可以用 BASIC 表达。此外，在游戏和模拟等应用中，操作员和 BASIC 之间能灵活进行会话。在程序运行期间，BASIC 用户不仅可以输入数据，而且还可以询问数据。

特别是在教育领域里，BASIC 的易学性应视为主要优点。但 BASIC 也有一个主要的缺陷，这不足为奇。BASIC 的这一先天性缺陷，就是不合乎目前人们倾向的所谓“结构程序设计”的思想。结构程序设计的原理在于逐步“求精”解题方法，故特别需要“分程序结构”，而 BASIC 却不具备这种结构。BASIC 的这一先天性缺陷是不可避免的，因为如果要弥补这一缺陷，就必然会丧失其他一些优点，例如良好的会话能力和易学性等。

虽然如此，BASIC 这种格式仍然很快地得到了人们的承认。它也许会在较长时间内占领微型计算机舞台，特别是在未来十年的联邦共和国中小学教育领域里会是这样。这是由中小学教育领域的具体情况决定的。目前，价廉物美的单板机系统只有 BASIC 才能使用。同样，供学校使用的大一些的程序库也只有用 BASIC 语言才能存在下去。使用超越 BASIC 程序水平的程序设计语言之所以失败，主要是由于教师们缺乏基础知识和进修能力，以及由于计算机系统的价格过高。

对于业余程序爱好者来说，使用面向机器的汇编语言，在节省时间和存储量方面是特别有利的。但这种语言的一个主要缺点是，缺乏从一种计算机系统到另一种计算机系统的转换能力（缺乏互换性）。由于目前大多数微型计算机不但性能优良而且价格低廉，所以在购置计算机时就不大考虑费用问题了。可惜的是，几乎所有的厂商现在都不顾及标准 BASIC，这也许会导致在不同的计算机系统之间运行同一 BASIC 程序时发生困难。

无论是学校师生还是业余程序设计爱好者，本书提供的程序将大有帮助。人们不用花费很多的时间，就可以打下从 BASIC 程序通向标准化问题的基础。

至此为止，可以说已经回答了本章一开头提出的问题：为什么一定要选用 BASIC 作程序设计语言。为了能够描述各种算法，BASIC 包括了五个基本成分（见第三章）。除这些基本成分之外，还约定了其他一些语言成分（见第五章），其中最重要的是“文字处理”。这些应用现在已远远地超过了传统的应用。

第二章 BASIC 程序的结构

2.1 基本结构

任何 BASIC 程序都是由一系列语句构成的，单个 BASIC 语句由一个行号（有的也称作语句标号）开始，以按 RETURN（回车）键结束^①：

```
10 INPUT  A
20 PRINT  A
```

其中，10、20 是程序的行号，行号确定了 BASIC 语句的排列顺序。为了便于修改程序，在必要时可以插入其他语句，行号间距选择为 10 比较合适（10，20，30…）：

```
10 INPUT  A
15 PRINT
20 PRINT  A
```

BASIC 语句可以不按“正确”的行号顺序输入，计算机会自动地按行号由低到高顺序执行。行号在程序转移中起着重要的作用。

就象大家都说同一种语言能互相听懂一样，一个 BASIC 计算机能懂得 BASIC 语言并能处理 BASIC 程序。

由于 BASIC 语言主要由一些简单易懂的英语单词构成，故你可以在几小时内学会这种语言的基本语句。而此后你能否编出“好”

① 为了清晰起见，译文中将原书程序语句后面的 RETURN 一律省去。

——译者注

的程序，则完全取决于你是否弄懂了所要解决的问题并且正确解答了它。

注意：错误并不是计算机造成的，而是程序造成的；错误并不是程序造成的，而是程序员造成的！

可惜许多 BASIC 计算机并不是在 BASIC 命令输入时能立即报告简单的书写错误，而是要等待程序执行到某一时刻才报告错误，这是一个较为不利的因素。因为书写错误易于识别和改正，而在程序运行时发现的错误则较难排除。

改正一条错误语句的简单方法是将其用同一行号予以重写。

现在，我们来看看程序执行时会发生什么情况？

在发出直接命令 RUN 后，计算机就从行号最小（通常是 10）的语句开始执行 BASIC 程序。计算机每次要确定：是应该执行一个它可以直接执行的语句（输入，输出，赋值，运算）呢，还是处理一个转移语句（IF...THEN, GOTO）。如果应该转移，就不继续执行该语句后面的下一个语句，而是“跳”到另一个语句：

```
100   IF   A=B   THEN   200
```

```
110   GOTO   300
```

行号为 110 的语句意思很明确：在任何情况下，程序都无条件地转移到行号为 300 的语句。而行号为 100 的语句则首先判定 A 和 B 的值是否相等，如果相等则转移到语句 200；如果条件不满足，就继续执行下一个语句（此处为语句 110）。由此可见，在没有任何外界干预的情况下，程序总是可以运行的。

任何一个程序的工作效率在于是否可以频繁地重复运行指定的程序段。现代电子数据处理技术，就是基于所谓循环结构（冯·诺伊曼思想）这一程序控制原理。

BASIC 的基本结构简单而清晰，这既是 BASIC 的优点，也是 BASIC 的弱点。之所以是优点，是因为人们能够很快地学会它，而且在计算机上实现时所付出的代价相当少。对于初学计算机的中小学学生和业余爱好者来说，这两个优点都是很重要的。BASIC

基本结构的主要缺点是缺少“结构化”程序设计的先决条件。使用结构化程序设计可以使算法逐步求精。BASIC 却因没有参数调用和所谓分程序结构的过程说明功能而无法进行逐步求精。

2.2 五个基本成分

和其他程序设计语言一样，BASIC 应能完成五项任务 输出，输入，赋值，转移，运算。

有关 BASIC 语句的含义，将在第三章和第四章详细阐述。任何一个解算问题的程序都可以由这五部分组成。这就是说，如果不考虑实际的运行时间，那么，一个算法问题的某种解，总是可以用 BASIC 来描述的。

其他语言成分只能起方便程序员的作用。但是太多的语言成分会破坏程序的清晰性和可移植性。因此，初学者不应在程序设计上钻牛角尖。基本而简练的方法是最合适的。

在介绍 BASIC 的五个成分之前，让我们先来谈谈“数据的结构”。简单地说，谈一谈 BASIC 程序加工处理的“原理”是什么？

在 BASIC 中我们必须区分两类变量，即 两类存储空间的占有者。

(1) 数值变量 每个数值变量代表一个可用来进行算术运算的数值。在 BASIC 中，每个数值变量至多用两个字符表示，其中第一个字符必须为字母，第二个字符可以是字母，也可以是数字。例如：A，B，X1，YZ。

(2) 非数值变量 每个非数值变量代表一个“字符串”，通常它最多有 256 个字符。在 BASIC 中，字符串变量用附在变量名之后的符号“\$”来标识。例如：A\$，X1\$，YZ\$。

注意：BASIC 系统能自动区分数值变量和非数值变量，允许人们在同一个程序中同时使用 A 和 A\$。

第一个成分：输出

没有输出语句的程序实际上是无意义的，因为我们不可能知道运行情况和结果。在 BASIC 中，数值变量和非数值变量的输出可使用同一语句

PRINT

来实现。该语句看上去非常简单，但实际使用时会涉及一系列问题（详见 3.1 节）。

第二个成分：输入

在大多数情况下，了解题需要使用不同的输入数据，但输入的不一定都是数字。例如，在游戏中也可输入文字。与输出不同，没有输入的程序也可能有实际意义。例如，列出从 1 到 100 所有素数的清单。除 READ 语句外，最重要的 BASIC 输入语句是

INPUT

语句。在每个这样的语句之后，计算机停止程序的运行，等待输入数据（参见 3.2 节）。

第三个成分：赋值

我们把数据（数值，文字）赋给变量的第一种可能性就是使用已经学过的输入语句。此外，数值变量通常是通过计算赋值，非数值变量则通过字符串的文字处理赋值。这些都是由赋值语句

(LET) =

实现的。该语句的作用是将赋值号右边的实际值或字符串赋给左边的变量。在所有 BASIC 语句中，变量的值可以保持到由重新赋值或输入改变它为止。通常，赋值语句前面的定义符“LET”可以省略（上述用圆括号括起来即表此意）。但请注意，赋值并不是数学上的相等（详见 3.3 节）。

第四个成分：转移

利用到目前为止学过的输出、输入、赋值成分，我们已能写求解问题的 BASIC 程序了。不过，要使用计算机还缺乏一个重要的成分。即，应该允许根据某一条件反复运行 BASIC 程序的某一部分（循环），或者跳过 BASIC 程序的某一部分（经判断而选择执行程序另一部分）。这种条件转移在 BASIC 中用

IF...THEN

语句实现。当计算机接到这样的命令时，就判断其条件是否成立：如果 IF 后面的条件满足（为真），则转移到 THEN 后的语句行号指定的语句；如果条件不满足，则运行下一个 BASIC 语句。

在 BASIC 中，还可以无条件地转移到一个语句。无条件转移由

GOTO

语句实现。在这种情况下，无条件地转移到 GOTO 后面的行号指定的语句继续执行。

第五个成分：运算与比较

进行数值处理时，常常需要在两个数值变量之间执行+、-、*和/等基本运算。

有时，还必须能比较两个数值变量或非数值变量的“大小”。即，在两个这样的变量之间用=、<或>等符号进行比较。例如对非数值变量（文字）而言， $A\$ < B\$$ 表示字符串 A\$按字母顺序位于字符串 B\$之前（详见 4.1 节）。

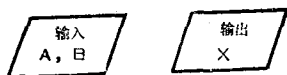
2.3 框图

书写和调试 BASIC 程序是解答问题的最后步骤。编写 BASIC

程序时，首先应有思路，根据数学解直接编写程序乃是最常见的手段。用框图描写解题方法是编写 BASIC 程序的一种辅助手段。这种框图虽然本身并不包含 BASIC 语句，但它可以既方便而又准确地改写成 BASIC 程序。当一个 BASIC 程序用框图描述时，就使人很容易理解。

我们可以通过建立框图的单元与 BASIC 五大成分（参见 2.2 节）之间的一一对应关系，来展示框图与 BASIC 之间的“亲缘关系”。

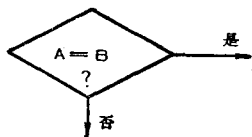
(1) 输出/输入 用如下的平行四边形表示。



(2) 赋值及运算 用如下的矩形符号表示。



(3) 条件转移 用如下的菱形符号表示。



其余部分不再一一说明。有了上述对应关系，就可以把框图符号直接翻译成 BASIC 程序。为了说明各符号间的关系，我们把一个求 A、B 二者最大值的例子，分别用框图和 BASIC 程序描述如下（图中 VON 表示“值”）：