

高等学校教材

# 计算方法 与算法语言

张德荣 王新民 高安民 编

(第二版) 下册 (算法语言)



高等教育出版社

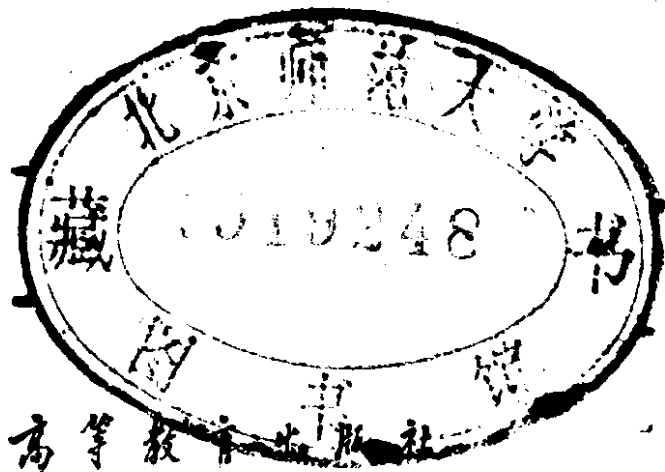
711-551-5  
高等学校教材

# 计算方法与算法语言

(第二版)

下册(算法语言)

张德荣 王新民 高安民 编



## 内 容 提 要

本书是在1981年出版的高等学校试用教材《计算方法与算法语言》的基础上修改再版的。全书分上、下两册出版。上册为计算方法，下册为算法语言。这次再版，删去了原书第一篇计算方法的少量内容，也增加了不少材料，并对不少内容进行了改写；对第二篇算法语言作了彻底改写，改为讲述BASIC语言的最新版本 True BASIC，并着重介绍结构化程序设计方法。内容选取的范围和深度恰当，推理比较严密，对算法作了较充分的说明。

本书可作为师范院校的教材，也可作为其它院校有关课程的教材或参考书。

本书下册配有软盘一张(西文一张或汉化一张)，内容包括: True BASIC 系统程序、各章例题的源程序、部分习题答案(源程序)，有需要者，可与高等教育出版社软件编辑室联系。

高等学校教材

### 计算方法与算法语言

(第二版)

下册(算法语言)

张德荣 王新民 高安民 编

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装

开本 850×1168 1/32 印张 7.25 字数 180 000

1981年12月第1版 1989年8月第2版 1989年8月第1次印刷

印数 0 001—3,310

ISBN7-04-002399-7/TP·54

定价1.80元

## 再版说明

这次再版,删去了原书第一篇计算方法部分的少量内容;根据教与学的需要,改写了不少内容;还增加了一些选学材料。

对于原书第二篇算法语言部分,根据计算数学及其应用软件编审组会议的意见,作了彻底改写。把讲述算法语言 ALGOL60 改为讲述结构式语言 True BASIC,介绍 True BASIC 语言的主要内容和编写结构程序的基本方法。

目前常用的 BASIC 语言的版本不少,它们之间,既有共同之处,也有差别,但使用它们编写程序的基本方法大致相同。考虑到这种情况,书中还介绍了常用 BASIC 版本的一些情况,希望本书的适应范围较广。

书中所列程序的数量和种类都不算少,有些内容有一定难度,希望同志们酌情取舍。算法和程序的关系极为密切,不少问题,用较多的篇幅先介绍算法。希望读者不要在对算法不甚了解的情况下,企图熟悉解决问题的程序。

为了使用方便,本书分上(计算方法)、下(算法语言)两册出版。北京师范大学陈公宁教授审阅了上册的全部内容,北京自动化工程学院谭浩强教授审阅了下册的全部内容,均提出了不少宝贵的建议,编者谨致以诚挚的感谢。

编者对计算数学及其应用软件编审组的全体同志,以及兄弟院校的同行人,也致以诚挚的谢意,感谢他们所给予的支持和帮助。希望继续提出批评和建议。

编 者

1988年6月于陕西师范大学

## 初版说明

本书是按照高等师范院校《计算方法与算法语言》教学大纲的要求编写的。为了适应不同情况的需要，书中列出了少量选学内容。这些内容，都在标题中加了星号“\*”，使用时可酌情取舍。

阅读本书仅需《高等代数》和《数学分析》中的基础知识。对一些不甚常见的内容，书中作了概括地介绍。

为了书写简便，用“定理 2.6.2”表示第二章 §6 定理 2；用“定理 2.1”表示本章 §2 定理 1；关于引理和公式，也有类似的记法。此外，符号“ $\triangleq$ ”表示“定义为”，即此符号左右两端是同一事物的两种不同表示方法。

本书试用过几次，使用的总学时是：计算方法 72 学时，算法语言 30 学时，上机实习 12 学时。各章的学时分配，比大纲的相应部分略有增减。

西北大学数学系刘国良同志阅读了计算方法手稿，并试用过几次，还有些兄弟院校也试用了本教材或者阅读了交流讲义，他们都提出了修改意见，特此致谢。

吉林大学李荣华同志、北京师范大学刘贵贤同志和七机部二院二十六所袁兆鼎同志审阅了本书，都提出了不少宝贵的意见，谨致以衷心的感谢。

我系系主任魏庚人老师对本书的编写予以鼓励和支持，谨于此致谢。

本书第一篇计算方法由张德荣同志编写，第二篇算法语言由王新民、高安民二同志编写。

近年来,计算数学这一学科发展很快,编者对大纲内容的深度和广度掌握不够,且限于水平,选材不当及错漏之处在所难免,恳请兄弟院校的老师 and 读者批评指正.

编 者

一九八一年六月于陕西师范大学

## 下 册 目 录

<b>第一章</b>	<b>引言</b> .....	<b>1</b>
§ 1	从用户观点看计算机.....	1
§ 2	结构化 BASIC 语言.....	2
§ 3	True BASIC用法概况.....	4
§ 4	编辑源程序.....	7
§ 5	及时把程序存入磁盘.....	10
<b>第二章</b>	<b>简单语句</b> .....	<b>12</b>
§ 1	数据的基本类型.....	12
§ 2	表达式和赋值语句.....	14
§ 3	源程序格式.....	18
§ 4	PRINT 语句和 INPUT 语句.....	21
§ 5	DATA, READ 和 RESTORE 语句.....	27
<b>第三章</b>	<b>判断结构</b> .....	<b>30</b>
§ 1	逻辑表达式.....	30
§ 2	IF 语句.....	34
§ 3	SELECT CASE 语句.....	44
<b>第四章</b>	<b>循环结构</b> .....	<b>49</b>
§ 1	FOR 循环.....	49
§ 2	DO 循环.....	58
§ 3	PRINT USING 语句.....	74
<b>第五章</b>	<b>数组</b> .....	<b>79</b>
§ 1	一维数组.....	79
§ 2	多维数组.....	89
§ 3	矩阵语句.....	92
§ 4	矩阵运算和表达式.....	97
<b>第六章</b>	<b>函数和子程序</b> .....	<b>101</b>

§ 1	子程序	101
§ 2	函数	116
§ 3	递归调用	123
§ 4	应用举例	132
§ 5	数组排序	143
<b>第七章</b>	<b>文件和程序库</b>	<b>158</b>
§ 1	概述	158
§ 2	正文文件	163
§ 3	记录文件	167
§ 4	字节文件	174
§ 5	通道参数	176
§ 6	CHAIN 语句和库文件	179
<b>第八章</b>	<b>绘图</b>	<b>184</b>
§ 1	基本绘图语句	184
§ 2	绘图子程序	193
§ 3	窗口	197
§ 4	应用举例	200
<b>第九章</b>	<b>转向语句</b>	<b>211</b>
§ 1	转向语句	211
§ 2	GOSUB 语句	215
<b>习题</b>		<b>217</b>
<b>附录</b>	<b>程序索引</b>	<b>222</b>
<b>参考书</b>		<b>224</b>



# 第一章 引言

## §1 从用户观点看计算机

电子数字计算机由控制器、运算器、存储器和输入输出设备等部件构成。前三者是计算机的主体,构成主机。运算器与控制器是计算机的核心部分,称为中央处理机,简称CPU (Central Processing Unit)。

通常使用的微型计算机,起码包括一台主机、一个键盘和一个显示器。还可以扩大存储器的容量、配备打印机绘图仪等输出设备,以及配备既可作为输出设备又可作为输入设备的磁盘驱动器。

主机能执行算术运算和逻辑运算,能传递信息,为进行复杂的科学计算和数据处理提供了良好的基础。

计算机的组成部件,包括电子的、磁性的以及机械的,统称为机器系统或硬件。硬件只能执行专门的十分繁琐的机器语言。为了发挥机器的作用,让更多的人能够相当方便的使用计算机,多种多样的系统程序,应运而生。这些在计算机硬件和用户之间起桥梁作用的系统程序称为软件。硬件和软件统称为计算机系统。

一般计算机用户,大多不使用机器语言,而是通过软件间接地使用计算机。久而久之,说到计算机,往往指的是包括硬件软件在内的计算机系统。不必对硬件的结构和软件的内容有详细的了解,只要熟悉它们的功能和用法,就可以有效地使用计算机系统。

为了科学计算和数据处理的需要,人们设计出各种高级语言。用高级语言编写的程序,计算机都不能直接执行,只有把它先翻译成机器指令才能使计算机执行它。将用高级语言编写的程序(源程

序)翻译成机器语言程序(目标程序),采用两种不同的方式:编译方式,是将源程序全部翻译成机器指令后,再将机器指令组成的目标程序交计算机执行;解释方式,它不是产生出目标程序后再交计算机执行,而是边翻译边执行,不产生整个目标程序.将高级语言程序转换成机器语言程序的翻译程序称为编译程序或解释程序.无论是编译程序还是解释程序,都在用户和计算机硬件之间起桥梁作用.

既然算法语言是用户和计算机硬件的中间环节,设计算法语言时,既要考虑到易读易写,使用方便,又要考虑到便于编译,充分发挥机器效能,还要考虑到可以在多种不同的机器上实现.从这一点看,算法语言又是用户和计算机系统之间的协议书.

编译程序或解释程序都和硬件有关,按某种算法语言规则所写的源程序则和机器无关.这就大大减轻了用户的负担,使得用户只要了解算法语言的各项规定,就可以使用计算机系统.

算法语言的规则十分严格,只要有错误存在,就必然得不到正确的结果.

## §2 结构化BASIC语言

BASIC是Beginner's All-purpose Symbolic Instruction Code(初学者通用符号指令代码)的缩写.除了这一种算法语言外,常用的还有ALGOL(ALGOrithmic Lanquage),PASCAL(以数学家命名)等算法语言.这些算法语言,各有特点.BASIC语言曾以其简单易学而广为流行.为适应新的情况,它也不断地向前发展.

各种算法语言,都是以便于编出质量较高的源程序为目的而设计的.从形式和功能看,程序是解决某一类实际问题的语句序列.随着实践的发展,人们很快认识到编写容易阅读,容易修改并且便于证明程序的正确性十分重要.于是提出了结构化程序设计

的概念。

解决任何一个复杂问题,都归结为按照一定的逻辑条件,解决一系列较为简单的任务来完成。把解决某一特定任务的语句序列称为模块,程序又可以看成模块的序列。于是程序设计可以用积木化的方式完成。

从序列的观点看,只反映了程序的书写顺序,是静态的固定不变的。结构化程序还应当反映各模块的执行顺序,每一模块中各语句的执行顺序,清晰地表现出动态结构。

要写出结构化程序,不仅依赖于用户的努力,程序设计语言本身还应提供有利的条件,便于写出结构化程序。

由BASIC 创始人 John G. Kemeny 和 Thomas E. Kurtz 等人于 1985 年推出的 True BASIC 是一种典型的结构化程序设计语言。本书介绍 True BASIC 的大部分内容和编写结构化程序的基本方法。

在当前情况下,把结构化程序设计语言作为“普及推广计算机应用的启蒙语言”,不仅可能,而且还是提高程序设计教学水平的有效措施。

为了养成编写结构化程序的良好习惯,本书严格限制 GOTO 语句的使用。学过其它非结构化 BASIC 语言的读者,最初可能不习惯这种限制,但是经过一段时间以后,就会体会到结构化程序具有易读、易写、易改等优点,会自觉地使用结构化程序设计方法写出结构化程序。

一般用户学习算法语言的目的,大多是打算用它作为工具,解决一些实际问题。学会结构化程序设计方法,不仅有利于进一步学习计算机科学理论,而且有利于解决实际问题。

任何一种算法语言,都不会使用很多年,但它的概念将在多年内仍为设计算法语言的人们继续采用。掌握了算法语言基本精神

以后,不仅能使用一种算法语言,解决实际问题,查阅用户手册,还能使用各种类似的算法语言。只有这样,才能不断前进。

### §3 True BASIC 用法概况

本书主要介绍 IBM PC 机 True BASIC 语言, True BASIC 语言与旧有的各种 BASIC 语言有良好的互通性,和其它各种结构化 BASIC 语言,如 Better BASIC, Turbo BASIC,大同小异。

使用 True BASIC 语言要求如下:

内存储器至少 192 K;

至少一个双面磁盘驱动器;

一台显示器。

本书在标题前用星号“\*”表示选学内容,用两个星号“\*\*”表示其它 BASIC 语言和 True BASIC 的主要差别,着重介绍 IBM PC 机上流行的几种 BASIC 语言和 True BASIC 语言的差别,介绍使用它们编写结构化程序设计的方法。

#### 1. 学习的出发点

使用 True BASIC 语言,要有一台配有 True BASIC 语言编译程序的计算机,例如 IBM PC 及其兼容机。当屏幕上显示

A>

的时候,就可以开始工作。

上述信息告诉用户,磁盘操作系统正在等待用户下达命令。磁盘操作系统(DOS, Disk Operating System 的缩写)能完成的任务很多,可以复制文件、复制整个软盘,可以执行批命令文件和执行各种编译程序等。

这时,如果你有一张软盘,其中有 True BASIC 编译程序和有关的文件,而且编译程序的名称为 HELLO,那末你就在键盘上打入 hello (DOS 对于大小写字母不加区别),则在屏幕上有一条水

平直线把屏幕分为两部分,上一部分 17 行,没有字符或图形;下一部分 7 行,前 5 行内容如下.

True BASIC here.

Version 1.0

Copyright (c) 1985 by True BASIC, Inc.

Published by Addison-Wesley Publishing Company, Inc.

OK.

其中第 5 行的“OK.”表示 True BASIC 编译程序正在准备为您服务,可以下达命令.从下一小节起,将陆续介绍各种命令和源程序的编辑方法.

如果你用的计算机有硬盘 C,在此硬盘中有名为 TRUEBAS 的子目录,有关文件已经复制到此子目录内,并且硬盘的根目录内有名为 EXETRUE.BAT 的批命令文件,此文件的内容为:

```
CD C:\TRUEBAS
```

```
HELLO
```

则只要键入 EXETRUE 也可以使 True BASIC 编译程序为你服务.

和 DOS 有关的内容请参阅有关资料.暂时不学习这些内容,不影响以后的学习.

## 2. 窗口和光标

屏幕的作用是让用户了解计算机的运行情况和有关信息.一般屏幕显示英文时只能显示 25 行,每行 80 字符.把屏幕分为两部分,上一部分称为编辑窗口,显示源程序,供输入源程序或修改源程序时使用;下一部分称为记事窗口,记录你打入的命令,程序运行情况,以及程序运行过程的输出结果.

光标是显示在屏幕上的一个记号,只要在键盘上打一个或几个字符,就会看到屏幕上不但出现了新的字符,而且有一个记号随

着新出现的字符陆续移动,它就是光标。

现在若光标位于记事窗口,试按一次或几次标有“F1”或“F2”的键,再交替地按这两个键,观察光标的移动情况。

**F1** 使光标位于编辑窗口

**F2** 使光标位于记事窗口

### 3. 一个简单的例子

按F1键,使光标位于编辑窗口。输入 **print maxnum** 以后,再按一次←键(称为换行键),再输入 **end**,最后按F9键,则在编辑窗口出现

```
print maxnum
```

```
end
```

两行新的内容,在记事窗口出现一个浮点数,例如

```
3.59539e+308
```

```
OK.
```

如果确实如此,就说明你已经正确地编好了一个完整的程序,并且已经得到了结果。 如果不然,就要根据记事窗口提供的信息,改正源程序中的错误,获得正确的结果。

现在解释上面各项操作的用意。首先,要输入源程序。例子中的源程序,只有两行,每行都是一个语句。第一个语句为 **print maxnum**,其中 **maxnum** 是在同一台机器上不变的常数,表示计算机的最大数, **print** 表示输出到屏幕,整个第一个语句的功能是,把计算机能够表示的最大浮点数在屏幕上显示出来。第二个语句 **end** 表示程序结束。在有些 BASIC 语言中, **end** 可有可无, True BASIC 则规定必须以这一语句来标志程序的结束。其次, F9 键的作用如下。

**F9** 运行当前的程序

最后,请注意,程序的运行结果显示在记事窗口,如果要保存运行

结果,可以使用以后介绍的文件.

#### 4 命令

记事窗口的最后一行称为命令行. 这一行已经由编译程序显示了 OK., 如果光标在 OK. 的右侧, 则先输入 `run`, 再输入字符“换行”(按“换行”键), 就会得到和输入 F9 一个字符同样的结果. `run` 是一条命令.

有两种下达运行命令的方法: 一种是输入 F9; 另一种是输入 `run` 和“换行”键. 前一种比较方便, 但 F1 至 F10 这些键(称为功能键)各有不同的功能, 如果按错了键, 会得到意外的结果; 后一种虽然比较复杂, 但在输入了 `run` 之后, 可以检查一下, 命令 `run` 是否正确(包括是否在命令行, 命令行是否还有上面没有说过的其它符号), 然后再按“换行”键. 用英文下达任何命令必须以“换行”键结束, 计算机才开始执行命令, 不按“换行”键, 编译程序处于待命状态. 这时, 如果发现命令有错误, 例如把 `run` 错输入成 `rum`, 就不要按“换行”键, 先按位于“换行”键上面标有 ← 的退格键, 删去 `m`, 再输入 `n`, 最后确认命令正确, 才按“换行”键.

退格键能删去光标左侧的字符.

要结束 True BASIC 返回到 DOS, 先输入结束命令 `bye`, 再按“换行”键, 就返回到 DOS, 等待 DOS 命令.

### § 4 编辑源程序

输入源程序时, 可能发生错误; 初步运行以后, 也可能发现错误. 所以必须有比较方便的修改源程序的措施. True BASIC 的源程序, 是可以显示或打印的字符组成的文件, 能用许多字符处理软件来输入和修改. True BASIC 系统提供了输入和修改源程序的功能. 本节介绍使用这种方法.

#### 1 移动光标

如上节所说,先输入 **hello**, 再按 **F1** 键, 使光标位于编辑窗口, 任意输入几行英文. 随着字符的输入, 光标跟着也不断地移动. 在键盘右侧, 有四个标有箭头的键, 使用它们, 可以移动光标.

←使光标向左移动一个字符;

→使光标向右移动一个字符;

↑使光标移到上一行;

↓使光标移到下一行.

使用上列四个键, 就会把光标移到指定的位置.

## 2 插入和改写

当首次运行 True BASIC 时, 光标是一个短划, 这时, 试把光标移到一个英文字, 例如 **word** 的字母 **d** 上, 再按一次 **l** 键, **word** 变成了 **world**, 这一个单字后面的字虽然没有变化, 但都向右侧移动了一个字符. 连续多次输入字母, 会使屏幕上的一些字符移出屏幕, 但光标却不会移出屏幕. 现在按一次键盘右下方标有 **INS** 的键, 光标的位置不变, 但形状变了, 由一划变成了矩形小块. 这时再输入英文字母, 情况变了, 新输入的字符代替了原来位于光标处的旧字符. 这两种状态, 分别称为插入和改写.

**INS** 键可以把插入状态变为改写状态,

也可以把改写状态变为插入状态.

学会了上述移动光标键、**INS** 键以及上一节介绍的退格键的用法, 就会修改源程序, 进行增加、删除和修改. 为了提高修改效率, 下面还要介绍一些方法.

## \*3 快速移动光标

用 **Shift & tab** 表示先按下 **Shift** 键 (两个 **Shift** 中的任一个), 再按下 **Tab** 键, 等放开 **Tab** 键以后, 再放开 **Shift** 键; 同样, 用 **Ctrl & ←** (或 **→**) 键表示先按下 **Ctrl** 键, 再按 **←** (或 **→**) 键, 等放开



了←(或→)键以后,再放开 Ctrl 键。下面也是一些移动光标的方法。

<b>Home</b>	使光标位于当前窗口的开始;
<b>End</b>	使光标位于当前窗口的结尾;
<b>PgUp</b>	使光标位于当前窗口的前一页;
<b>PgDn</b>	使光标位于当前窗口的下一页;
<b>tab</b>	使光标位于下一字或下一个数的开始处;
<b>Shift &amp; tab</b>	使光标位于前一个字或前一个数的开始处;
<b>Ctrl &amp; ←</b>	使光标位于当前行的开头;
<b>Ctrl &amp; →</b>	使光标位于当前行的结尾。

上述移动光标、插入和删除,对两个窗口都适用。当前窗口指光标所在的窗口。其余各项细节,请通过实践来了解。

#### \*4 删除

除了退格键能删除光标右侧的字符外,还有下列删除的方法。

<b>Del</b>	删除光标处的字符;
	当光标在行标记上时,删去当前行;
<b>Ctrl &amp; Home</b>	删除光标左侧一个单词;
<b>Ctrl &amp; PgUp</b>	删除光标右侧一个单词;
<b>ESC</b>	由光标直到行首都删去;
<b>Ctrl &amp; End</b>	由光标直到行尾都删去;

其中“行标记”指的是每行开始的矩形小块;用光标移动键,可以使光标位于此处。

#### 5 恢复

如果删错了某一行的一部分内容,按 F7 键,能恢复删去的内容。

**F7** 恢复删去的内容。