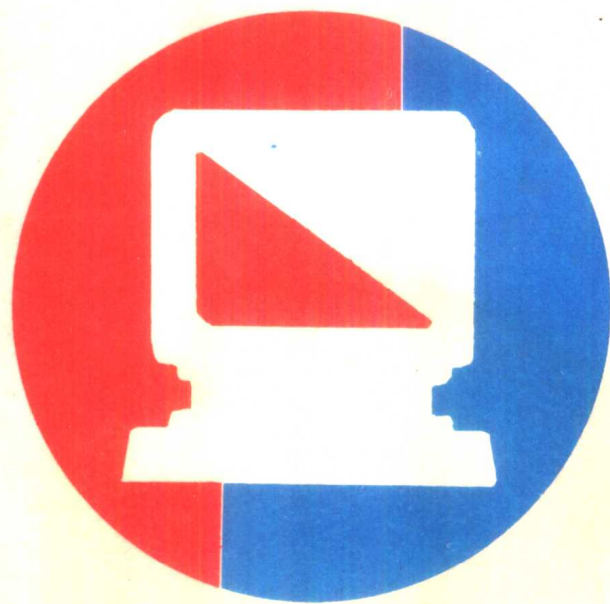


高等学校非计算机专业计算机基础课程教材
高等学校非计算机专业计算机等级考试用书

True BASIC 语言程序设计

朱学勤 主编



中国科学技术出版社

73.87424

C122

932262

阅 8 清

高等学校非计算机专业计算机基础课程教材
高等学校非计算机专业计算机等级考试用书



10364806

True BASIC 语言程序设计

朱学勤 主编

中国科学技术出版社

· 北 京 ·

(京)新登字 175 号

图书在版编目(CIP)数据

True BASIC 语言程序设计/朱学勤主编. —北京:中国
科学技术出版社,1994
ISBN 7-5046-1695-8

I. T… I. 朱… II. BASIC 语言—程序设计—教材
IV. TP312BA
中国版本图书馆 CIP 数据核字(94)第 05935 号

中国科学技术出版社出版
北京海淀区白石桥路 32 号 邮政编码:100081
新华书店北京发行所发行 各地新华书店经销
北京市燕山联营印刷厂印刷

*

开本:787×1092 毫米 1/16 印张:24.75 字数:540 千字
1995 年 1 月第 1 版 1995 年 4 月第 3 次印刷
印数:10001-15000 定价:22.00 元

前 言

本书是安徽省高等学校计算机基础课程系列教材之一,是为适应当前非计算机专业的教学实际需要,以《安徽高等学校计算机水平考试大纲(二级 True BASIC)》为依据而编写的。

True BASIC 是 BASIC 的两位创始人于 1985 年推出的结构化版本。它以其突出的特点,令人信服的功能,已赢得了广大用户的信赖,也成为一种学习高级语言程序设计的理想语言。

本书内容丰富,特色鲜明,既考虑到知识的系统性,又注意到重点突出;既注意概念的严格与准确,又力求叙述深入浅出、通俗易懂;为了提高读者编程的能力,本书从实用出发,选取了大量的例题,并进行透彻的分析,不少例题还做到知识性与趣味性共融,引人入胜。

全书正文部分共分十章。第一章概括介绍 True BASIC 基础知识及上机操作的基本方法。第二章介绍顺序结构程序设计,重点讲述常用的输入、输出语句。第三、第四章介绍选择结构和循环结构程序设计,这是程序设计的重要基础,也是本书的重点内容之一。第五章为数组,第六章为函数与子程序,这两章是结构化程序设计的重要组成部分,也是对实际问题进行编程训练的重要内容。第七章介绍字符串及其应用。第八章较系统地介绍 True BASIC 的文件及其操作,是本书的又一重点内容。第九章简要介绍图形功能。第十章是上机实习指导,较详细地介绍了上机操作方法,还提供了具体的上机实习内容,这也是本书的特色之一。附录部分,考虑到读者的实际需要,收集了一些很有实用价值的资料,相信会给读者带来不少方便。

因此,本书既可作为高等学校非计算机专业有关课程的教材,也是自学、培训的理想用书。

本书由朱学勤编写第一、第三、第六章及附录;楼诗风编写第二、第八章;潘瑜编写第四、第五章;孙强编写第七、第九、第十章。本书由朱学勤任主编,佟英泰教授担任主审。

在编写过程中得到安徽省教委高教一处领导的大力支持和帮助,琚仲达、袁振发同志为组织、指导编写本书做了大量的工作,在此一并表示真诚的感谢。

由于编者水平有限,书中错误和不妥之处在所难免,恳请同行专家不吝指教及广大读者批评指正。

编 者

1994 年 6 月

目 录

前言	(1)
第一章 True BASIC 基础	(1)
1 True BASIC 概述	(1)
2 True BASIC 程序结构	(4)
3 True BASIC 的基本操作	(5)
4 True BASIC 字符集及符号名	(9)
5 常量与变量	(10)
6 标准函数	(13)
7 表达式	(17)
8 例题	(19)
小结	(20)
习题一	(21)
第二章 顺序结构程序设计	(23)
1 程序设计序言	(23)
2 提供数据的语句(LET, INPUT, READ/DATA)	(24)
3 输出语句(PRINT)	(32)
4 注释语句(REM)、中止语句(STOP)和结束语句(END)	(36)
5 应用举例	(37)
小结	(38)
习题二	(38)
第三章 选择结构程序设计	(41)
1 结构化程序设计简介	(41)
2 关系表达式和逻辑表达式	(46)
3 简单分支选择结构及 IF 语句	(49)
4 多路分支选择结构及 SELECT 语句	(60)
5 应用举例	(61)
小结	(72)
习题三	(73)
第四章 循环结构程序设计	(78)
1 循环的基本概念	(78)
2 计数循环及 FOR—NEXT 语句	(80)
3 条件型循环及 DO—LOOP 语句	(87)
4 循环的嵌套——多重循环	(101)
5 循环结构应用举例	(110)
小结	(120)

习题四	(121)
第五章 数组	(125)
1 数组和下标变量的概念	(125)
2 一维数组及其应用举例	(128)
3 二维数组及其应用举例	(136)
4 矩阵语句(MAT)及矩阵运算	(143)
5 数组应用程序实例	(167)
小结	(183)
习题五	(183)
第六章 函数与子程序	(188)
1 自定义函数	(188)
2 子程序	(194)
3 内部函数(子程序)与外部函数(子程序)	(203)
4 函数与子程序的嵌套	(216)
5 库文件	(218)
6 程序间的链接	(221)
7 应用程序实例	(223)
小结	(234)
习题六	(235)
第七章 字符串	(239)
1 字符串常量与变量	(239)
2 字符串表达式	(242)
3 格式化输入	(246)
4 格式化输出	(249)
5 字符串函数	(255)
6 字符串数组	(257)
7 应用实例	(258)
小结	(264)
习题七	(264)
第八章 文件	(266)
1 文件概述	(266)
2 常用文件操作	(269)
3 正文文件的存取及应用实例	(274)
4 记录文件的存取及应用实例	(278)
5 字节文件的存取及其应用实例	(284)
小结	(295)
习题八	(296)
第九章 图形	(297)
1 工作方式、屏幕坐标和图形窗口	(297)
2 基本图形功能	(300)

3 图形着色	(308)
4 多重窗口	(311)
5 动画技术	(313)
6 图画	(317)
小结	(323)
习题九	(325)
第十章 True BASIC 上机实习指导	(327)
1 怎样使用 True BASIC	(327)
2 IBM-PC 编辑键和 True BASIC 系统命令	(328)
3 True BASIC 程序的编辑与运行	(331)
4 True BASIC 语言上机步骤	(333)
5 True BASIC 上机实习内容	(334)
附录	(1)
附录 I True BASIC 语法一览表	(1)
附录 II True BASIC 常见出错信息及解释	(9)
附录 III True BASIC 2.03 版	(20)
附录 IV EXE 文件生成软件——汇集程序	(28)
附录 V True BASIC 与 DOS 系统接口方法	(31)
附录 VI True BASIC 与其他可执行程序(.EXE)间的接口方法	(32)
附录 VII DO 命令的使用与生成	(34)

第一章 True BASIC 基础

1985年,在 BASIC 语言第一个版本推出整 20 年,而且正风行于世界时,它的两位设计者 J. G. Kemeny 和 T. E. Kurtz 却大声疾呼:“到了用一完美设计的现代版本来取代那些陈旧而且在执行上非常糟糕的 BASIC 的时候了!”一种新的 BASIC 版本以全新的面貌诞生了,那就是 True BASIC! 十年来的实践表明:它的确给 BASIC 带来新的生命力,从而开辟了 BASIC 的新纪元。

1 True BASIC 概述

1.1 从 BASIC 到 True BASIC

BASIC 是英文 Beginner's All-purpose Symbolic Instruction Code 的缩写,它的中文意思为初学者通用符号指令代码。它是由美国达特默思(Dartmouth)学院数学系的两位教授根据这所学院曾做过的一项承诺:计算机对于所有学生而言将变为非常容易掌握的工具,于 1964 年开发成功的一种计算机高级语言。它是当时十分流行的高级语言 FORTRAN 和 ALGOL 的精髓的混合体。正是出于设计 BASIC 语言的初衷,BASIC 以其良好的用户界面,易学易用的突出优点,受到了广大初学者的热烈欢迎,并使之很快冲出校园,被广泛应用于社会的各个领域。1971 年,微处理器问世,微型计算机异军突起,如雨后春笋蓬勃发展。这对于 BASIC 来说也真可谓天赐良缘!由于 BASIC 对硬件要求较低,大受微机厂商的青睐,几乎所有的微型计算机都毫不例外地首先配置了 BASIC 语言,有的甚至把它固化在内存中。这样,伴随着计算机的迅猛发展,BASIC 语言也走向世界,得到了迅速的推广和应用,也对我国计算机事业的普及和发展产生了巨大的作用和深远影响。但与此同时,由于微型计算机产品的激烈竞争,各厂商都争先在自己的微机上开发出多种多样的版本。这样做的结果,一方面使 BASIC 语言的功能不断得到强化,但另一方面也给它本身带来了各种问题,主要表现在以下三个方面:

(1) 方言性。如上所述,BASIC 的版本十分庞杂,以至到了“同一个 BASIC 程序将无法在不同的个人计算机上运行,甚至也不能在同一厂家生产的两个不同型号的机器上运行”的地步。因而被称为“Street BASIC”(大路 BASIC)——这是给漂亮语言所起的一个可怕绰号。BASIC 的方言性影响了它的通用性和可移植性,也给标准化造成了很大困难。

(2) 早期的 BASIC 语言版本多为解释方式。这种执行方式对初学者来说无疑有不少方便之处,然而它的执行速度较慢,而且对于硬件的依赖性也大。这也给 BASIC 的推广带来严重的影响。

(3) 很长时间里,BASIC 是非结构化的。这一缺点使之在大规模或复杂的程序设计中存在很大的局限性。

对 BASIC 的这些缺点,人们已从多方面对它进行了改进,如制定 BASIC 标准、推出编译 BASIC 版本、改良并增加结构化的控制结构等,但这些零打碎敲的改良,并未从根本上解决问题。于是,终于使它的两位创始人下定了本书一开始所援引的使它真正回到 BASIC 上来的决心。1985 年 11 月,Kemeny 和 Kurtz 推出了集 BASIC、FORTRAN 与 PASCAL 等语言的优

点之大成的新一代 BASIC 语言,并把它称为“真正的”BASIC——True BASIC”。

1.2 True BASIC 的主要特点

True BASIC 是一种结构化、模块化的语言。它是先进的程序设计语言,不仅保留了原 BASIC 简单易学、通俗易懂和适应性强等优点,又在功能上作了重大改进,使之具有结构模块清晰、作图功能强、编译紧凑、程序设计风格优美和可移植性好等特点,同原 BASIC 相比,具有质的差异。

归纳起来, True BASIC 语言有如下主要特点:

(1) True BASIC 是结构化程序设计语言。结构化程序设计是 60 年代后期,人们为对付日益严重的“软件危机”而由荷兰学者 Edsger W. Dijkstra 于 1965 年提出的一种程序设计方法。它的要点是程序设计者应当按照工程的方法,遵循一种规范化方法进行程序设计,纠正那种无章可循,使程序成为“个人工艺品”的状况。结构化程序设计方法认为一个好的程序应具备以下条件:运行结果正确,符合题目要求;有良好的结构,易读易懂;尽可能少的运行时间;运行时所占的存贮器空间应压缩在合理的范围之内等,并指出造成程序质量差、编程及维护效率低的主要原因是程序中滥用 GOTO 语句。结构化程序设计方法的提出是程序设计方法的一场革命,经过激烈的论争,它的理论和技术日益完善,已为人们所普遍接受,成为目前程序设计中的主流。

原有 BASIC 语言版本中,也正因为大量地使用 GOTO、GOSUB、ON—GOTO(GOSUB)等转向语句,使程序失去了良好的结构,导致了程序质量的降低。这也就是原有 BASIC 不适于编制较复杂软件的主要原因之一。True BASIC 舍弃了 GOTO 语句(但还允许保留),取消了行号而充实了选择结构和循环结构。在选择结构中,IF—THEN—ELSE 可用于多语句块,并可以嵌套,形成了多层二分支结构,还有 SELECT CASE 语句提供多分支选择。循环结构除了 FOR—NEXT 外,还可以用 DO—LOOP,并可方便地选择 WHILE 与 UNTIL 子句,使得循环结构十分灵活。

此外,为了适应模块化程序设计的需要, True BASIC 的函数和子程序除有内部的,还有外部的,而且设置了库(库文件)来存放外部的有用的函数和子程序,在编程时方便地予以调用。

综上所述,正如 True BASIC 的发明者所概括的那样:“在符合大多数计算机科学家所支持的‘结构化程序设计’原则的意义上说,它是现代化的。”

(2) 很强的图形功能。在原有的 BASIC 作图功能的基础上, True BASIC 作了很大改进,其中最重要的改进是:取消了坐标点与象素之间的换算,使用户在用它来作图时,可以象在纸上作图一样,根据自己需要确定屏幕坐标,而不必考虑屏幕上的象素点。

True BASIC 提供了图形子程序(PICTURES),并可装入图形库中,需要的时候可随时调用它,并可作移位、改变宽度、放大缩小、旋转和倾斜等五种变换。True BASIC 允许用户把屏幕开设成几个窗口,并根据需要进行窗口间的切换。在一个窗口里文字和图形可以并用,使用快速绘图语句 BOX 还可以实现简单的动画技术。

(3) 较强的编辑功能。True BASIC 在操作时将屏幕分为可调整的上下两个窗口,上面的窗口称为编辑窗口,用于显示所编写或使用的现行程序;下面的窗口称为背景(历史)窗口,可显示输入的所有命令和程序运行的结果。

True BASIC 的全屏幕编辑功能比原有的 BASIC 更强。它不仅能逐行对程序编辑,而且还可以对标定的模块,包括单行、多行组、一个函数、一个子程序或一个图形进行删除、复制或移

动的操作。

利用 FIND、CHANGE 和 TRY 命令,可以查找、统一更改或逐一更改所指定的字或字符串。对于调试好的程序,为使程序压缩得更简洁并采用标准格式,可键入命令"DO FOR-MAT"来实现。

用户掌握了 True BASIC 的这些编辑命令,可以更加得心应手,运用自如。

(4) 良好的可移植性。True BASIC 是严格按照美国国家 BASIC 标准而设计的,因此它有良好的可移植性而不依赖于硬件。True BASIC 程序可以不作任何改动就能在各种流行的微型计算机上运行,而且它们可以与能运用美国国家标准局的标准 BASIC 的较大系统兼容,甚至使用图形和窗口的程序也是可移植的。当个人计算机的差异很大时,为什么能做到这样呢?回答是语言的实现把从一台计算机转换到另一台计算机的一切繁重工作都承担下来了。

(5) 解释与编译方式相融。大多数通常使用的 BASIC 版本都是解释方式。解释方式固然给初学者带来许多方便,但它的缺点也是明显的,尤为突出的是对于大的程序的执行速度慢,且不能及时给用户提供清楚的出错信息。形象地说,在解释系统检测出一个非法语句以前,一个程序完全有可能已运行了 10 分钟之久了!而相反,编译方式就可以在编译阶段检测出所有的非法结构或调用的不合法错误,提供清楚而易于理解的出错信息,只有在修改合格后才去投入运行,而程序运行的速度则比解释方式要快得多。True BASIC 具有解释执行和编译执行两种方式,既有利于初学者的交互式对话、调试,又可用于生成快速执行的软件。True BASIC 提供的编译方法简单易行,只需用一条 COMPILE 命令,就能迅速编译成功,编译效率高。

(6) 充分利用内存。原有 BASIC 限制内存的使用在 64K 以内,致使较大规模的程序不能运行。True BASIC 则允许以绝对地址方式使用整个内存,从而充分利用了内存容量。

此外, True BASIC 还有较强的乐谱和声音功能,加上很强的图形、动画功能,适用于开发引人入胜的程序,这无论对于初学者还是有经验的程序员都具有很强的吸引力。True BASIC 还提供了矩阵(MAT)语句和各种矩阵运算,从而大大提高了矩阵处理及运算的效率,使这类处理的编程大为简化。

以上所述只是 True BASIC 的主要特点, True BASIC 的实际功能远比上述的更为丰富。True BASIC 语言对不同层次的用户都是一种很受欢迎的语言,可使用户得心应手地开发功能较强、结构良好的应用软件,而不必去另寻其它语言。一度在我国出现的对 BASIC 冷淡以至责难都将消失。

1.3 BASIC 展望

应当指出,在 BASIC 语言上进行结构化努力而形成的结构化版本并非只有 True BASIC。在 80 年代后期还有 TURBO BASIC 和 QUICK BASIC。它们相互补充各具特色。经过不断的推陈出新,已经发展成为一种功能齐全的集编辑、调试、编译、连接于一身的现代集成软件开发环境。它所具有的时钟功能、日期功能、运算功能、字符串处理功能、绘图功能、动画功能、音响功能、读写内存功能、文件处理功能、菜单功能、多窗口功能及多模块调用功能等等,与任何一种高级语言相比是毫不逊色的。1987 年,当时已经有 True BASIC, TURBO BASIC, QUICK BASIC 三种结构化版本,美国波士顿公司计算机学会对其成员作了使用计算机语言的调查,结果 BASIC 占 80%—90%之多。

进入 90 年代以来,软件界有两大趋势:一是 OOP (Object-Oriented Programming) 面向对象程序设计的兴起;一是 MICROSOFT 公司的 WINDOWS 获得空前的成功。伴随着 OOP 和 WINDOWS 的飞速发展, BASIC 也迅速发展,功能越变越强,在软件世界中的地位越变越

重要。最为典型的是 GFA—BASIC 和 Visual Basic 的出现。Visual Basic 是继承了 BASIC 语言的传统,用以前版本编写的应用程序,几乎可以不经变动或只需稍加修改即可在 Visual Basic 环境中运行,而且 Visual Basic 增加了事件驱动编程、可视界面设计等功能,Visual Basic 分为 WINDOWS 和 DOS 两种类型,在这两种类型中又分为标准版和专业版,其中专业版提供了更高级的专业式工具包功能。

世界上最大的软件公司 MICROSOFT 的缔造者现任公司董事长兼总经理 Bill Gates,这位杰出的计算机科学家曾在 1989 年预言:BASIC 的寿命超过我们这些人(他当时 33 岁),事实已证明他的预见的正确性。因此,结论应当是:BASIC 程序员根本不必担心 BASIC 语言的发展前景,把精力耗费在改用其它语言和移植程序的工作上,还是继续充满信心地使用 BASIC 系列语言编程吧!

2 True BASIC 程序结构

程序是解题过程(也称算法)的具体描述。程序设计是以高级语言为工具利用计算机解题的必经步骤,True BASIC 当然也不例外。下面是处理两个简单问题的 True BASIC 程序例子。

2.1 引例

例 1-1 计算半径为 40 厘米的圆的面积。

```
5 REM 计算圆面积
10 LET R=40          ! 给定圆的半径
20 LET P=3.14159      ! 给定圆周率  $\pi$  的值
30 LET S=P * R * R    ! 根据圆面积计算公式计算出圆的面积.
40 PRINT "面积为:";S  ! 输出结果
50 END               ! 结束程序
```

例 1-2 求 $\sum_{n=1}^{100} n$ (即 $1+2+3+\cdots+100$)

```
! Compute : SUM=1+2+3+...+100
let sum=0
for n=1 to 100 step 1
let sum=sum+n
next n
print "sum:";sum
end
```

2.2 True BASIC 程序的结构

从上述两个例子,我们可以看出:True BASIC 程序是由若干语句行组成;一个程序的最后一行应为 END,它是表示该程序的结束。

(1) 语句行的格式。True BASIC 程序的语句行的格式为:

[<语句行号>] <语句关键字> <语句体>

语句行中,语句行号、语句关键字和语句体三部分之间用“空格”来分隔。“空格”在计算机中是一个特殊的字符,若在键盘上按一下空格键,则屏幕上的反应是光标向右移动一个字符。

位置,但无任何内容显示。有时为了书写的方便常用“_”来表示。

(2) 行号用以决定语句的执行次序。在 True BASIC 中一般不使用它,只是考虑与非结构化 BASIC 版本的兼容才予以保留。在有行号的情况下,可使用旧的 BASIC 控制结构,如 GOTO、GOSUB、ON 等。但规定,只要在一个程序中的一个语句行使用行号,那么所有语句行都必须有行号,且严格按递增的次序编号,其取值范围为 1—999999 之间的整数。

(3) 空格和大、小写字母。空格除了不能置于关键字、变量名、数字、复合运算符(如 $>=$)的中间和行号的前面外,可以出现在其它任何地方,其作用是用以增强程序的可读性。例如 `let a = a+1` 和 `let a=a+1` 都是合法的,但 `leta=a+1` 是非法的。另外,如有行号,则行号后面至少应有一个空格。

True BASIC 不区分字母的大小写(字符串常量除外)。因此,它认为“LET”与“let”是同一关键字。在语句、文件名、变量名、子程序中大写、小写或大小写混合都是允许的。

(4) 关于注释。注释是对程序、程序段或语句行的功能的一种备注,用以提高它的可读性。注释不会影响程序的执行速度,因此没有必要去省略。

注释的方法是一行的开头或末尾,用感叹号(“!”)打头,例如上面引例中所示,也可以用 REM 语句来写注释,注意 REM 后面应插入一个空格。不允许把注释放在一行的中间。

(5) 语句行的长度。True BASIC 允许一个语句行内最多可有 30767 个字符,但每行只允许写一条语句(注意这是与通常的 BASIC 版本有很大的差别!)

3 True BASIC 的基本操作

True BASIC 源程序的输入、编辑及运行与通常的 BASIC 版本有较大的差别。在本节中仅介绍它的基本操作,以便读者能边学习边实践。有关上机操作及上机实习的详细内容集中在本书的第十章,读者随时可以参考。特别是在进行教学时,应根据进度适时穿插上机操作的内容。

3.1 上机操作步骤

3.1.1 进入 True BASIC

在开机进入 DOS 操作系统之后,在屏幕上显示 A>(硬盘启动则显示 C>)插入 True BASIC 系统盘,并键入命令:

A>HELLO

稍等一会,在屏幕的下半部显示所使用系统的版本信息,并最后出现“ok”字样。这表示 True BASIC 已启动成功,进入了 True BASIC 环境,允许用户进行 True BASIC 的操作。

3.1.2 屏幕窗口

True BASIC 与一般 BASIC 版本不同的是:它把屏幕划分为上下两个区域(窗口),中间用一条横线隔开,如图 1-1 所示。

True BASIC 在屏幕上部的窗口叫编辑窗口(editing window),供用户编辑输入源程序;下部的窗口叫背景窗口(history window),供用户输入 True BASIC 命令或显示程序运行结果。由于它可以“记忆”最近执行的命令或程序运行结果的历史痕迹,故也叫历史窗口。鉴于 True BASIC 操作上的这一特点,读者应当熟悉对窗口的操作。

(1) 窗口的转换。在进入 True BASIC 环境时,光标位于背景窗口内,这时用户可输入各种 True BASIC 命令。如果用户要编辑输入程序,就必须将光标转换到编辑窗口。转换的办法

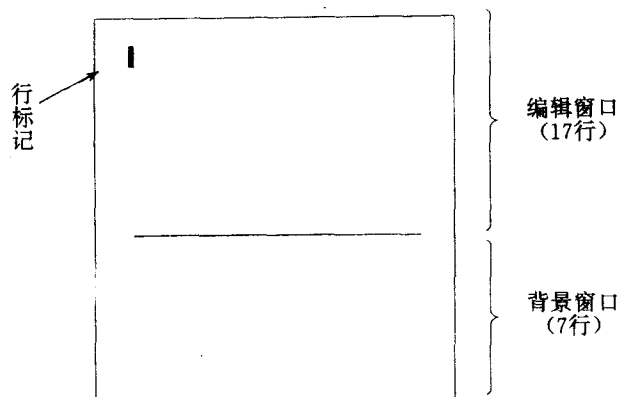


图1-1 True BASIC窗口

是:按一下F1键,光标就移至编辑窗口;按一下F2键,光标又从编辑窗口返回到背景窗口。反复按动这两个键,就可以在两个窗口之间不断转换。用户必须注意:输入命令必须在背景窗口,编辑程序则要在编辑窗口。

(2) 窗口大小的调整。进入 True BASIC 时,窗口的大小是系统默认的大小:编辑窗口占 17 行,背景窗占 7 行,中间横线占一行。由于执行程序的命令及程序的运行结果显示在背景窗口,如果运行结果超过了 7 行,在屏幕上就不能一次看到完整的全部运行结果。这时,可以调整两个窗口的宽度:减小编辑窗口,增大背景窗口。实现这一调整的命令为:

SPLIT n

其中:n 为编辑窗口所占的行数,其范围为 $0 \leq n \leq 24$,例如:

SPLIT 4

此时,屏幕立即就会将编辑窗口缩小至 4 行,背景窗口则扩大至 20 行

(3) 窗口的前景与后景颜色的设置。如果你使用的显示器不是单色的而是一台彩色显示器,那么你还可以使用组合控制命令设置屏幕的前景或后景的颜色。具体操作方法如下:

① 当光标位于编辑窗口内,按 $\wedge F$ (CTRL+F) 改变编辑窗口前景的颜色;按 $\wedge B$ 则改变编辑窗口的后景的颜色。若连续按键,则连续循环改变。

② 当光标位于背景窗口,按 $\wedge F$ (CTRL+F) 改变背景窗口前景的颜色;按 $\wedge B$ 则改变背景窗口的后景的颜色。若连续按键,则连续循环改变。

③ 改变边缘颜色。使用 $\wedge E$ 可改变边缘颜色,若连续按键,可以连续改变颜色,当选到一种合适的颜色时就停止按键。

3.1.3 编辑源程序

编辑源程序在编辑窗口内进行。编辑时先将光标移至编辑窗口,逐个字符逐行输入源程序,每输完一行,按回车键,光标自动移至下一行的起始处。程序最后以 END 结束。如图 1-2 所示。

<pre> FOR I=1 TO 10 PRINT I, NEXT I END </pre>
ok.

图 1-2

3.1.4 修改源程序

由于 True BASIC 对于编辑窗口中的源程序是进行全屏幕编辑的,因此,只要利用表 1-1 中编辑键即可对源程序进行各种修改。

表 1-1

键名	功能
→	光标右移一个字符
←	光标左移一个字符
↑	光标上移一行
↓	光标下移一行
HOME 或 END	光标移至文件首部或尾部
PGUP 或 PGDN	使程序向前或向后翻页
^←	光标移至本行行首
^→	光标移至本行行末
Backspace	删除光标左边的一个字符
Del	删除光标处一个字符
Del(光标在行首)	删除光标所在一行
^HOME	删除光标左边一个单词
^PGUP	删除光标处单词的右边部分
ESC	删除光标的左边部分
^END	删除光标的右边部分
F7	恢复当前删去的一行
TAB	光标左移,右移一个单词(上下档)
ENTER	光标右边部分另起一行
INS	插入字符的开关键

True BASIC 具有很强的全屏幕编辑功能,除了上表中的编辑键所能完成的基本编辑功能外,还具备更加灵活、方便的编辑方法,详见第十章。

下面以输入、编辑、修改一个源程序为例来说明以上介绍的操作。

首先在 True BASIC 环境下键入 NEW 命令:

ok. NEW (在背景窗口下操作)

此时编辑窗口清屏。用 F1 将光标移到编辑窗口第一列,即可开始输入源程序如下:

正确的源程序为:

```
! Find the maximum from 3 numbers
!
INPUT prompt "Enter 3 numbers:":a,b,c
IF a>b THEN LET max=a ELSE LET max=b
IF c>max THEN LET max=c
PRINT "Maximum number is :";max
END
```

如果在编辑窗口输入程序时由于误操作,将源程序输入成:

```
! Find the maxmum from 3 nobers
!
INPUT promtp "Enter 3 numbers:":a b c
IF a>b THEN max=a ELSE max=b
IF c>max THEN max=b
PRINT "Maxmium number is: ";max
END
```

输入上述程序操作中有多处输入错误,现用编辑键进行修改。先用 F1 键使光标回到编辑窗口,然后按以下步骤操作:

(1)用光标右移键将光标移至 maxmum 的中间一个 "m" 处(横线标出),在插入状态下键入 "i", "i" 即插入到 "m" 之前;

(2)用光标右移键将光标移至 nobers 的 "o" 处,按 DEL 键删去 "o",光标停在 "b" 处。在插入的状态下键入 "u" 及 "m",得到 numbers;

(3)下移光标键至第3行 promtp 的 "t" 处,按 INS 键使之关闭插入而处在 "改写" (替换)状态,依次键入 "p"、"t",即用 "pt" 替换 "tp" 再按 INS 键,回到插入状态。

(4)将光标右移至本行 abc 的 "b" 处,键入 ",", 再移至 "c" 处,再键入 ",", 使之改正为 a,b,c;

(5)将光标移至第4行的 max=a 的 "m" 处,在插入状态下按 "LET", 按空格键;再移至同行 max=b 的 "m" 处以及下一行的 max=c 处的 "m" 处,用同样的方法插入 "L", "E", "T" 和空格;

(6)最后将光标移至 print 语句中的 Maxmium 的中间一个 "m" 处插入 "i", 成为 Maximium, 然后将光标右移至 "i" 处(改成 Maximum), 按 DEL 键删去字符 "i", 使之成为 Maximum, 从而完成修改。

完成了全部修改,将用 F2 键将光标移至背景窗口,即可运行该程序。

3.1.5 源程序的运行

运行源程序,首先要用 F2 键将光标移至背景窗口,一般停留在 ok. 的后面,然后键入

"RUN"(运行命令)。如果程序已无语法错误,则执行程序,例如运行上面已修改的程序:

```
ok. RUN
Enter 3 number :? 7,8,11
Maximum number is 11
ok.
```

3.2 退出 True BASIC 环境

当光标处于背景窗口时,只要键入"BYE"命令即可退出 True BASIC 环境,返回到操作系统状态。例如:

```
ok. BYE↵
则显示
```

A>

这表明机器已退出了 True BASIC 返回 DOS 状态,用户可根据需要进行另外的操作或关机。

4. True BASIC 字符集及符号名

4.1 True BASIC 字符集

字符集是指计算机语言规定允许使用的字符的集合。True BASIC 允许使用任何机器能识别的字符。它通常是指 ASCII 码符号(American Standard Code for Information Interchange 美国信息交换标准代码,详见附录 I)。True BASIC 的基本字符包括:

- (1) 字母。包括英文26个大写字母(A—Z)、26个小写字母(a—z),共52个。
- (2) 数字。数字包括0—9共十个。
- (3) 专用字符。True BASIC 使用的有确定含义的专用字符如表1-2所示。

表1-2

符号	名称	符号	名称	符号	名称	符号	名称
+	加号	(	左括号	"	双引号		竖直线
-	减号	)	右括号	'	单引号	\	右斜线
*	乘号	.	句号或小数点	&	合并号	[	左方括号
/	除号	,	逗号	\$	美元符号	]	右方括号
	乘方号	:	分号	#	#字符号	{	左花括号
=	等于号	:	冒号	%	百分号	}	右花括号
>	大于号	!	感叹号	@	圈a		
<	小于号	?	问号	_	下划线		

此外, True BASIC 还允许使用其它一些专用字符,它们虽无明确含义,但也可以显示或打印出来。如果你使用的 True BASIC 版本已经汉化了,那么汉字也将包括在它的字符集内。

因此,上机时应注意你使用的 True BASIC 版本。为了提高可读性,本书大部分程序中使用了汉字。如果你使用的 True BASIC 版本未经汉化,那么在输入时,应将汉字部分改为英文或拼音,也可以略去。

应当特别注意的是,在 True BASIC 程序中允许字符一定的组合,例如 $\geq$ (代表大于等于), $\lt \gt$ (代表不等于),AND(与运算符)等等,但决不允许出现字符集外的任何字符,诸如希腊字母 π 、 α 、 β ,等以及一些常用的数学符号如: $\geq$, $\neq$,等等。

4.2 符号名

符号名泛指在程序中使用的变量名、数组名、自定义函数名、子程序名及文件名等。True BASIC 语言规定符号名用英文字母(大小写均可)开头,后面跟字母、数字、下划线的一串字符来表示,其最大长度为31个字符。规定名字时,应尽量与其含义能一致,例如 SUM(求和的意思)、NAME 或 XM(姓名的意思)等等。这样也可以提高程序的可读性。

在 True BASIC 系统中已规定为保留字的,由于系统已赋予它们专门的含义,而不允许程序员去改变它的含义。因此,保留字的整个字作为不可分割的符号使用,不得再作符号名使用。True BASIC 保留字见表1-3。

表1-3 True BASIC 保留字

关键字	标准函数名	标准子程序名
ELSE	DATE MAXNUM	
ELSEIF	DATE\$ PI	DIVIDE
IF	DET RND	PACKB
NOT	EXLINE\$ TIME	POKE
PRINT	EXTTEXT TIME\$	
REM	EXTYPE	

5 常量与变量

一个程序,如果从组织形式上看,可以象第2节归纳的那样;然而,如果从数据的角度来看,则可以分为输入数据、处理数据和输出数据三个部分。简而言之,数据是程序处理的对象。

在程序中,数据通常有两种形态,即常量与变量。

5.1 常量

常量(Constand)是程序运行过程中始终保持不变的数据。一般来说,常量是在程序执行前就是已知的而且在程序执行过程中始终不发生变化的。

常量是 True BASIC 语言加工处理的最基本的对象,它是组成 True BASIC 语言的主要部分之一。

在 True BASIC 中,常量一般可以分为两大类:数值常量和字符串常量。还有一类数据称为逻辑值("真"和"假"),因为它涉及到关系运算和逻辑运算,将在第三章中再作详细介绍。下面着重讨论 True BASIC 的数值常量和字符串常量。

5.1.1 数值常量

数值常量也叫常数。常数是由一个正号或负号、一个数值部分和一个指数部分组成。其中,正号通常可以省略;数值部分可包括1位或多位数字,以小数点为界分为左右两边,若小数点右边无数字,则小数点可以省略;指数部分以字母"e"(或"E")开头,后面跟一个正号或负号,还