

计算机语言实用程序 与编程技巧

True
Quick BASIC
Turbo
语言

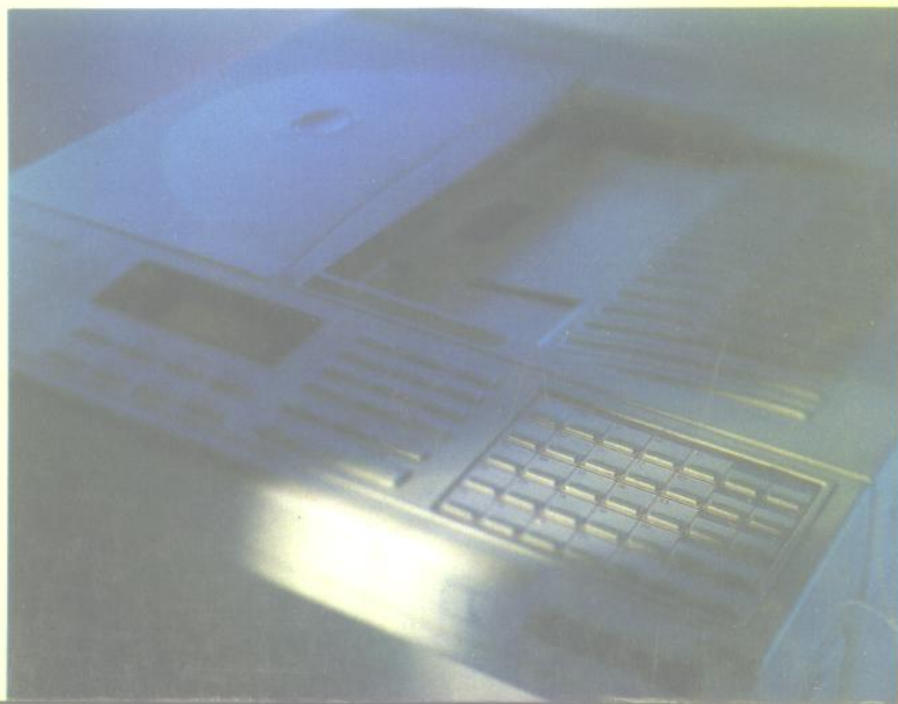


张后苏 向南平 等 编 著
黄健柏 刘兴权

王云宜 审 定



中南工业大学出版社



计算机语言实用程序与编程技巧

**True
Quick
Turbo**

BASIC 语言

张后苏 向南平 等编著
黄健柏 刘兴权
王云宜 审定

中南工业大学出版社

【湘】新登字 010 号

计算机语言实用程序与编程技巧

BASIC 语言

张后苏 向朝平 董理柏 刘永权等编著

王云宜 审定

责任编辑：肖艳高

中南工业大学出版社出版发行

中南工业大学出版社印刷厂印装

新华书店总店北京发行所经销

*

开本：787×1092 1/16 印张：11.5 字数：284 千字

1994 年 11 月第 1 版 1994 年 11 月第 1 次印刷

印数：0001—7000

*

ISBN 7-81020-690-7/TP·041

定价：9.00 元

本书如有印装质量问题，请直接与生产厂家联系解决

《计算机语言实用程序与编程技巧》丛书

编委会成员

编委会主 任: 王云宜

编委会副主任: 田荣璋

编委会成 员 (以姓氏笔画为序):

王云宜 毛先成 田荣璋 向南平

任朝阳 陈福盈 杨路明 肖梓高

吴耀斌 张后苏 崔秀梅 彭先定

雷方桂

前言 一

《计算机语言实用程序与编程技巧》丛书暂包括六个分册。它们是:《C 语言》;《Turbo Pascal 语言》;《80386 汇编语言》;《BASIC 语言》;《FOXBASE+ 数据库语言》;《FOXPro 数据库语言》。

编写这套丛书的目的是为了帮助已经掌握了以上语言基本编程方法的读者进一步提高程序设计技术,学习高级程序设计的方法和技巧。在应用程序设计工作中,借鉴于已有的、经过优化的现有实用程序与实用程序段,使用到应用程序中去,节省编程周期,不断提高应用程序的质量和效果。它是大专生及相关人员最好的教学参考书和工具书。

近年来,各种程序设计语言,从 BASIC 到 C 都在不断的完善和丰富自己,版本不断更新,各种语言在保持自己特色的同时,功能上均在相互接近,互相渗透,取长补短,不断改进。掌握好任意一门程序设计语言都能满足绝大部分应用程序设计的需要。本丛书选材正是从各种语言的最新版本功能出发,叙述各种语言的高级程序设计技术和实用程序。使读者缩短掌握语言新功能的途径。

《C 语言》不仅叙述了基本的静态、动态数据结构,还系统地阐述了内存管理、视频系统及应用、图形、资源辨认、常驻内存程序、数值计算等方面高级程序设计方法,有很多都是作者最新科研成果的总结。

《Turbo Pascal 语言》由于其结构化、模块化特点,不仅是大专学生的基础课程,也是当今中学生奥林匹克竞赛的规定语言之一。本书不仅适合大专学生提高的需要,也适合竞赛需要,适合中学生提高 PASCAL 语言程序能力的参考书。

《80386 汇编语言》则以 8088/8086 为基础,重点阐述 80386、80486 和 80387、80487 的程序设计技术,供程序员参考使用。

《BASIC 语言》是传播最早、最广泛的计算机语言。它对 Quick Basic 和 Ture Basic 功能作了详细叙述,其矩阵语句、图形功能、动画能力、计算能力、I/O 功能及人机图形、窗口界面丝毫不弱于其它任何语言,编程者可以设计出具有各种特色的应用程序。

《FOXBASE+ 数据库语言》总结了目前最常用和实用的程序设计技术,在菜单设计、窗口设计、查询搜索、工具应用等方面都进行了系统的阐述,还有不同类型管理系统的实用程序示例,供读者直接引用或模仿。它是数据库管理系统开发人员和应用程序设计人员最好的一本参考书。

参加本书编写的人员都是长期从事教学和科研工作的教师,他们具有扎实的基础理论和实践能力。全书所有程序均经过上机验证,不少是科研成果的总结汇编而成,写法上注意了先进性、实用性,尽量符合读者学习的思维过程,因而可读性好,极具实用价值。

由于计算机软件的发展一日千里,丛书不可能全部包含当前应用中最新内容,不当之处,尚望批评指正,并愿为广大读者提供咨询服务。

编委会

1994 年 10 月

前 言 二

BASIC 语言是国际上广泛使用的一种计算机高级语言，它的全称是 Beginners All purpose symbolic InstructionCode，即初学者通用符号指令代码语言。BASIC 语言清晰、易懂、易学，自 60 年代起在我国得到了广泛的应用，在推动我国计算机应用与普及方面起到了重要的作用，目前，它仍是计算机入门的主要学习语言之一。但是随着计算机科学的飞速发展，它自身的弱点也日益明显。这些弱点主要表现在：它是一种非结构化程序设计语言，难以实现程序的结构化和模块化，不利于初学者养成良好的程序设计风格。

1985 年，BASIC 语言创始人 John G.Kemenz 和 Thomas E.Kurtz 等人根据美国国家 BASIC 标准 (1984)设计了一个典型的结构化程序设计语言 True BASIC。它保留了 BASIC 易懂、易学的特点，屏弃了非结构化 BASIC 版本中陈腐有害的内容，成为一个适合于大型软件设计，具有较高效率和完整性的结构化程序设计语言。

1987 年底，美国 Microsoft 公司又推出了 BASIC 语言新版本 Quick BASIC。Quick BASIC 基本上具备 True BASIC 结构化程序设计标准，同时也有自己的鲜明特色。它集解释型与编译型于一体，既具有解释型的简单易学与调试检测的方便性，又具有编译型的快速与通用性。它还提供了十分完善的程序开发环境和丰富的标准函数和语句，大大地扩大了 BASIC 语言的功能。

1987 年 4 月，美国 Borland International 公司也推出了一个 BASIC 语言新版本 Turbo BASIC。它在功能上大致与 Quick BASIC 相当，亦得到广泛的应用。

上述三种 BASIC 语言新版本，对原有 BASIC 语言版本在保留其优点的基础上作了革命性的变革，使 BASIC 语言跟上了当今计算机技术的发展，基本上具有和 Pascal、C 语言并驾齐驱的完美程度。本书综合介绍了这三个新版本的主要内容和特点，并以实际例子来说明三种 BASIC 的编程技巧。

True BASIC、Quick BASIC 和 Turbo BASIC 各具有不同的编程环境和特点，概括起来，可综述如下：

(1)结构控制语句丰富、灵活，保证了程序的结构化和模块化，便于程序的分析验证。对程序中的变量，根据其作用范围分为局部变量、静态变量和共享变量、增强了数据的隐蔽性，这些都有助于软件的编制和调试，提高软件的可维护性。

(2)具有很强的图形、音乐和动画功能，有助于开发出具有友善的用户界面的软件，对初学者和有经验的程序员都有很强的吸引力。

(3)可快速生成 DOS 状态下直接运行的可执行文件(EXE)，运行起来比解释 BASIC 程序快数倍到数十倍。

(4)具有很强的库管理功能，可直接调用外部子程序或库文件；可随心所欲地建立自己的子程序库，增强了软件开发的灵活性。

(5)可充分利用内存，不受 64K 区段的限制，能以绝对地址的方式对主机的全部内存进行存取。

(6)提供了与 DOS 操作系统的简练而方便的接口方式, 可在程序中直接调用 DOS 系统的命令和功能, 并可与外部的可执行程序连接。

(7)有灵活的联机求助功能, 随时可对 BASIC 语句、命令、ASCII 码等提供帮助信息, 可在屏幕上的特定窗口直接测试程序的任何语句。

(8)与已普及的 BASIC 版本有较大的兼容性, 原有 BASIC 程序一般不必做大的修改, 便可在新系统中运行, 有的还提供了自动的转换程序。

(9)系统工作时解释与编译并行, 将解释型 BASIC 的方便性和编译型 BASIC 的快速性合为一体, 自动地把编译、运行和调试有机地结合起来。

(10)Quick BASIC 和 Turbo BASIC 具有功能很强的集成开发环境。它们采用下拉式菜单, 对话方式操作。

(11)True BASIC 的矩阵语句, 使求解线性代数问题的编程变得容易; Quick BASIC 容许用户自定义新的数据类型(记录类型), 它可以将若干个不同类型的变量组合成一种用户定义的类型, 类似于 C 语言中的“结构”和 PASCAL 语言中的“记录”, 利用该语句可使随机文件的输入/输出得以简化。

由于上述特点, 使得 BASIC 语言无论对初学者还是有经验的程序员都仍然是一种很受欢迎的语言, 可以用它得心应手地设计功能强、结构完善的应用软件。

本书共七章和五个附录, 第一章由张后苏和刘兴权共同编写, 第二章由张后苏和黄健柏共同编写, 第三章由刘兴权编写, 第四、五章由向南平编写, 第六章由向南平和刘兴权共同编写, 第七章由黄健柏编写, 张后苏担任主编并最后修改定稿, 中南工业大学的雷春和陈家参加了编写工作。

由于编者水平有限, 错误之处在所难免, 恳请专家和读者批评指正。

作者

1994 年 11 月

目 录

1	TRUE BASIC	(1)
1.1	TRUE BASIC 环境	(1)
1.2	基本语句	(5)
1.3	结构语句	(7)
1.4	数组与矩阵	(11)
1.5	自定义函数、子程序、模块与库	(13)
1.6	文件	(20)
1.7	图形与音乐	(24)
1.8	出错处理	(30)
1.9	常用命令、语句和函数	(31)
2	QUICK BASIC 4.5	(43)
2.1	QUICK BASIC 环境	(43)
2.2	数据类型及作用域	(48)
2.3	基本语句	(52)
2.4	结构控制语句	(54)
2.5	子程序、过程与库	(56)
2.6	文件	(59)
2.7	图形	(61)
2.8	音响和音乐语句	(66)
2.9	出错与事件处理	(70)
2.10	常用命令、语句和函数	(72)
3	TURBO BASIC	(79)
3.1	TURBO BASIC 环境	(79)
3.2	基本语句	(82)
3.3	结构语句	(85)
3.4	函数和过程	(88)
3.5	数据文件	(91)
3.6	图形	(94)
3.7	常用命令、语句和函数	(97)
4	操作系统的基本概念	(103)
4.1	磁盘的基本知识	(103)
4.2	文件和文件管理的概念	(105)
4.3	DOS 的启动	(106)
4.4	操作系统的基本命令	(108)
4.5	汉字操作系统的功能与应用	(112)

4.6 在三种 BASIC 语言系统中使用汉字	(114)
5 结构化程序设计	(115)
5.1 结构化程序设计思想	(115)
5.2 结构化程序设计方法	(115)
5.3 结构化程序设计的风格	(117)
5.4 程序中的算法设计	(118)
6 TURE BASIC 语言程序实例	(120)
7 QUICK BASIC 程序实例	(133)
附录 1 TRUE BASIC 错误信息表	(152)
附录 2 QUICK BASIC 错误信息表	(156)
附录 3 TURBO BASIC 错误信息表	(162)
附录 4 高级 BASICA 的命令、语句和函数	(166)
附录 5 常用 DOS 命令	(172)

1 TRUE BASIC

1.1 TRUE BASIC 环境

(一) TRUE BASIC 语言的特点

True BASIC 是 BASIC 语言的创始人 John G.Kemeny 和 Thomas E.Kurtz 等人于 1985 年推出的集 BASIC、FORTRAN 与 PASCAL 等语言优点的新的一代 BASIC 语言。它严格符合美国国家标准，是一种结构化、模块化的语言系统，且同时提供解释方式与编译方式，在保持 BASIC 原来的优点的基础上，进一步提高了程序的清晰度，有了很好的可移植性并在功能上作了较大的扩充，能够提供比普通 BASIC 大得多的用户空间，并且有了很强的编辑功能。原有的 BASIC 程序一般不必作很大的改进，便可在 True BASIC 环境中运行，运用 True BASIC 的附加软件包，可容易的将原 BASIC 程序转换为 True BASIC 程序。

(二) TRUE BASIC 语言的启动与退出

True BASIC 语言系统完全包含在一个称为 HELLO.EXE 的文件中，要启动 TRUE BASIC，只需将装有该文件的盘片插入 A 或 B 驱动器中，或使用磁盘操作系统中的 COPY 命令将该文件拷贝到硬盘的任意目录或子目录中，然后在 DOS 命令状态下执行该文件，即在 DOS 提示符后打入 HELLO 并按回车，稍等片刻就进入了 True BASIC 语言环境，此时，屏幕显示如下：

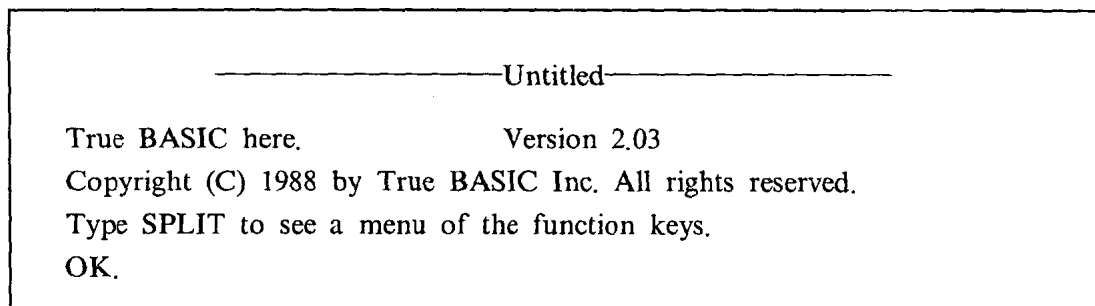


图 1-1 True BASIC 窗口

不同的 True BASIC 版本，英文说明稍有不同，图 1-1 为 2.03 版的屏幕显示。“OK”是 True BASIC 的提示符，它告诉用户系统已处于命令状态，此时可发布任何 True BASIC 命令。

在命令状态下键入命令“BYE”，便会退出 True BASIC，返回到 DOS 状态。

(三) TRUE BASIC 窗口

窗口是在屏幕上开设的一块显示区域。在 True BASIC 命令状态下，屏幕被分成两个

窗口，如图 1-1 所示。上面的窗口叫做编辑窗口(Editing Window)，可供用户书写、显示或编辑 True BASIC 程序，在编辑窗中每个输入行的最左边有一个实心长方标志，称为行标志，每输入一行都会产生一个行标志；下面的叫做背景窗口或命令窗口(History Window)，供用户键入命令和显示程序执行结果，可记录最近执行的命令或程序执行结果的历史痕迹。

光标停在哪个窗口，哪个窗口即为当前工作窗，用 F1 和 F2 键来改变当前工作窗。按 F1 键，光标移到编辑窗，按 F2 键，光标移到命令窗。

两窗口之间的分界线中间，是系统设置的文件名监视口，当用户编辑未命名文件时，其中显示 Untitled，当用户存入或调出一个已知文件名的程序时，上述显示变成当前存入或调出的实际文件名。

(四) 编辑窗口的编辑命令用法

程序的编辑修改是程序设计中经常进行的工作。True BASIC 的编辑功能是在全屏幕状态下进行的，即在需要插入或删除某些字符时，先用键盘操作把光标移到要插入或修改处，然后在光标所在处进行所需的编辑。表 1-1 是光标移动键的名称和功能，表 1-2 是编辑键的名称和功能。

表 1-1 光标移动键及其功能

键 名	功 能	键 名	功 能
→	右移一个字符	PgUp	向上翻页
←	左移一个字符	PgDn	向下翻页
↑	上移一行	Ctrl+→	移至行尾
↓	下移一行	Ctrl+←	移至行首(行标志)
Home	移至程序首行	Tab	右移一个字
End	移至程序末尾	Shift+Tab	左移一个字

表 1-2 编辑键的名称及其功能

键 名	功 能	键 名	功 能
F4	标记行模块，再按一次取消	Ctrl+Home	删除光标左边一个字(词)
F6	移动带标记的行模块	Ctrl+PgUp	删除当前一个字(词)
Esc	删除光标左边的行内容	Ctrl+End	删除光标右边的行内容
>	使标记行模块右移一格	Ins	插入或覆盖状态转换开关
<	使标记行模块左移一格	F8	设置断点
Shift+F1	垂直向上在结构层上移光标	Shift+F3	建立标笺行
Shift+F2	垂直向下在结构层上移光标	Shift+F4	寻找标笺行
Backspace	删除光标左边一个字符，光标处于标志处时，将当前行连接到上一行末	Del	当光标处于行标志处时，删除当前行，否则，删除光标处的一个字符。
F5	在编辑窗口，复制带标记的行模块到光标所在行的紧后面；在命令窗口，重复上一条命令。	F7	取消当前行的修改，恢复当前行内容；将最近删除的行块恢复于光标所在行的上面

(五) TRUE BASIC 语言常用命令及其用法

1. **SPLIT 命令** 在启动 True BASIC 时, 编辑窗为 17 行, 命令窗为 8 行。若要改变编辑窗的大小, 可用该命令。其用法为:

SPLIT [n]

其中: n 为具体的数字, 范围为 0~24。

在 2.03 版本中, 无数字 n 时, 可在两窗分界处显示功能键的功能提示。

2. **OLD 命令** 用于由磁盘读入一个 True BASIC 程序, 命令格式为:

OLD [D:][PATH]<文件名>[.扩展名]

其中: D 为指定驱动器名, PATH 为文件路径名。

若文件在当前盘和当前目录, 可省略路径。若是 True BASIC 程序, 可省略扩展名。编辑窗口中的原文件若修改后没有存盘, 系统在命令窗中提示:

“Do you want to save this file?”

若用户回答 YES 或 Y 并按回车键, 则自动存入原文件名中, 如果是未命名文件, 还要输入文件名; 若用户回答 NO 或 N, 系统舍弃此文件, 执行所给命令。

用户退出 True BASIC 系统, 即执行“BYE”命令时, 情况类似。

3. **NEW 命令** 清除编辑窗中的已有程序, 以便编写新的程序。

编辑窗口中的原文件若修改后没有存盘, 系统提示及用户回答同 OLD 命令。

4. **RUN 命令** 打入该命令并按回车键时, 系统开始执行编辑窗中的程序。

该命令也可用功能键 F9 产生, 即按 F9 可代替输入 RUN 及回车键。

5. **FILES 命令** 用于查找磁盘文件目录。命令格式为:

FILES [D:][PATH][文件名][.扩展名], length]

无扩展名选择项时, 只显示指定目录中所有以 TRU 为扩展名的文件目录。有选择项 length(能缩写为它的前三个字符 len)时, 将显示出以字节为单位的文件长度。这是 2.03 版本新增的功能。

6. **SAVE 命令** 用于将编辑窗中的程序存入磁盘文件。命令格式为:

SAVE [D:][PATH]<文件名>[.扩展名]

文件名是用户命名并要存入的文件名, 它应符合 DOS 系统对文件名的规定。如果省略扩展名, 系统默认扩展名为 TRU。如果命令中的文件名在指定目录中已经存在, 则系统在命令窗中提示:

“File already exists”

若仍要以该文件名存盘时, 可改用 REPLACE 命令。

7. **REPLACE 命令** 以新内容代替原文件的内容并存入磁盘文件, 文件名不变。若在命令中给出一个文件名, 则系统将程序写入指定文件名的文件中。

8. **CHANGE 命令** 从当前光标位置开始, 将编辑程序中的所有指定字词(老字词)改为要更换的字词(新字词), 而不管大小写。命令格式为:

CHANGE <老字词>, <新字词>

9. **TRY 命令** 该命令功能类似于 CHANGE 命令, 只是在每次找到一个要替换的字词后, 给出一个选择的机会, 决定是否要替换。命令格式为:

TRY <老字词>, <新字词>

从光标位置开始，逐个寻找老字词，每找到一个后，系统提示：

“change ?”

键入 Y 及回车键或直接按回车键，执行替换；键入 N 及回车键，放弃替换；键入 Q 及回车键时，系统自动停止查找并退出 TRY 命令。

10. EDIT 命令 该命令用于集中编辑某一段程序。先将要编辑的程序段作上标记，然后键入 EDIT 并按回车键，系统将给出信息：

“Now editing marked region (现在编辑标记的行模块)”

并清除该行模块以外的编辑窗口中的其它内容。当编辑完这个行模块后，先消除行模块标记，再执行该命令，则在编辑窗口中恢复全部程序行，并在命令窗中给出信息：

“Now editing whole file (现在编辑整个文件)”

11. KEEP 命令 用于由一个大程序中剪裁出一个程序段。该命令有两种用法：

①先对所需行模块打标记，然后在命令窗中键入 KEEP 并打回车键，这时编辑窗中便只留下打标记的行模块。

②若要剪裁一个完整的子程序，则可用 KEEP 命令加子程序名参数的方法进行剪裁。例如，源程序中有一个名叫 subprog1 的子程序，可在命令窗中打入“keep subprog1”并按回车键，则系统自动删除该子程序以外的程序行，只保留子程序 subprog1 的内容。

剪裁后可以重新指定文件名并存盘。

12. INCLUDE 命令 该命令用于由磁盘中读入一个 True BASIC 程序，插入到编辑窗中光标所在行之后。命令格式为：

INCLUDE [D:][PATH]<文件名>

命令中参数含义同 OLD 命令。

13. LIST 命令 用于将编辑窗中的程序向打印机输出，既可输出整个程序，也可输出程序中的部分行模块。要输出全部程序行，只需在命令窗中输入 LIST 并按回车键。若要输出部分行时，其使用方法有以下几种：

①先给要输出的行模块打标记，再执行 LIST 命令。

②在 LIST 命令中指定要输出的子程序名。

③在 LIST 命令中指定要输出的行号。例如，要输出第 1 行到 50 行的文件内容，可打入如下命令并回车：

LIST 1-50

14. UNSAVE 命令 用于删除盘上的已有文件。命令格式为：

“UNSAVE [D:][PATH]<文件名.扩展名>”

15. RENAME 命令 用于改变一个文件名字。有两种使用格式：

①RENAME <newname> 用于改变当前文件名字，并不影响磁盘上的任何文件。

②RENAME <oldname>, <newname> 用于将指定盘和路径的磁盘文件 oldname 改名为 newname，文件名必须包含扩展名，它不影响当前正在编辑的文件。然而，不能用该命令来重新命名一个文件到不同的磁盘。

16. ENTER 或 CD 命令 该命令用于改变当前目录。命令格式为：

ENTER / CD <directory>

若指定的目录不存在，则显示“No such directory”。

1.2 基本语句

(一) 程序行格式

True BASIC 程序由一系列简单语句行和控制语句行组成。每一个简单语句用一个关键字开始, 并占一行。控制语句也由一个关键字开始, 但可以占据多行。有关规定如下:

①行的长度。一行最多可有 32767 个字符, 每行中只容许有一个语句。

②行号。行号是任选的, 在有行号的情况下, 可使用旧的 BASIC 控制结构, 如 GOTO、GOSUB、ON。若在程序中有一个语句行有行号, 则所有语句都要有行号, 且严格按递增的次序编号, 行号必须是 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(如大于等于)的中间和行号的前面外, 可以出现在其它任何地方。也建议这样做, 以增加程序的可读性。除字符串常量外, 不区分字符的大小写。

④注释。只能放在某一行的开头或末尾, 并以 REM 或!打头。

(二) 简单语句

1. !(或 REM)语句 在!或 REM 后的任何内容均被认为是注释语句的内容, 而不被执行。其中!既可放在一语句行的开头, 又可放在一语句行的末尾, 而 REM 只能放在语句行的开头。

2. LET 语句 使用该语句给变量赋值。语句格式为:

LET <变量名> = <表达式>

其中: LET 不可省略, 一语句单独占一行。

3. PRINT 语句 该语句用于显示常量、变量和表达式的值。语句格式为:

PRINT <变量 1> [, <变量 2>] [: <变量 3>] [,], [:]

或 PRINT [USING format\$:] <变量 1> [, <变量 2>] [: <变量 3>] [, ...], [:]

用该语句输出数值时, 其前面留一个符号位, 后面留一个空格, 输出字符串时, 前后都不留空格。

多个显示项之间既可用逗号相隔, 也可用分号相隔, 或交替用逗号和分号相隔。前者称分区格式, 后者称紧凑格式。采用分区格式时, 一行 5 个区, 每区 16 个字符宽, 每一个显示项占一个区。采用紧凑格式时, 每个显示项后部留有一个空格, 紧接着显示另一个。若最后一个显示项的末尾带有逗号或分号, 则在 PRINT 语句显示后不换行, 后续 PRINT 语句继续在该行以分号区格式或紧凑格式显示, 直至 80 列被占满时自动换行。

PRINT 后没有显示项时, 显示一个空行。

选用 USING format\$ 项时, 用以定义输出格式。format\$ 格式串中的格式符有以下二类。

①数字的格式符号。

#—每个#代表一位数字, 数值前置的零由空格符来表示。如: "#####"格式代码显示 45 时, 其结果为" 45";

%—每个%代表一位数字, 数值前置的零由0来表示, 如: "%%%%"格式代码显示 45 时, 其结果为"0045";

* — 每个 * 代表一位数字，数值前置的零由 * 来表示。

如: " * * * * " 格式代码显示 45 时，其结果为 " * * 45 "。

在格式符号的首部，可以加以下符号：

+ — 数值前面可显示 "+" 或 "-"；

- — 数值前面可显示 "-"，若是正数时则不显示；

\$ — 数值前显示 "\$ "。

此外，在格式代码中间还可插入以下符号：

, — 将逗号插入数值中适当的位置，如: " ###, #### " 格式代码显示 10000 时，其结果为 " 10, 000 "；

. — 小数点定位显示；

^ — 数值科学表示法中显示指数部分。

② 字符串的格式符号。字符串的输出格式，由 # 组成的格式符号来定义，当字符串小于格式字符串所规定的长度时，则显示的字符串显示在中间位置，两边由空格充填，如果加有首部字符 "<" 或 ">" 时，则显示结果将靠左对齐或靠右对齐。如: "< ##### " 格式代码显示 "XD" 时，其结果为 "XD "。

4. INPUT 语句 使用该语句从键盘输入一个或多个值赋给某变量或多个变量。语句格式为：

INPUT [PROMPT <"提示">] <变量名 1> [, <变量名 2>]

输入的数据类型必须与变量类型一一对应。若少了，会要求再输入，多了，忽略后面多余数据，程序继续运行。

5. DATA/ READ 语句与 RESTORE 语句

DATA 称置数语句，其格式为：

DATA <数据 1>, <数据 2>,

使用 DATA 语句应注意以下几点：

① 语句可放在程序的任何位置，但必须放在 END 语句之前；

② 可用单一语句书写多个变量，也可将数据分散在多个语句中；

③ 语句中的数据次序不是随意的，读数时从第一个开始读；

④ 语句中只容许是常量，不能有变量或表达式或函数；

⑤ 语句中有字符串常量时，可不带双引号，但首尾有空格或中间有逗号时则必须带双引号。

READ 称读数语句，其格式为：

READ <变量 1>, <变量 2>,

其中的变量个数不能多于 DATA 语句中的数据个数，且变量类型必须与 DATA 语句中的数据类型一一对应。可利用系统提供的两个测试函数来判断 DATA 语句中的数据是否读完，这两个语句是 MORE DATA 和 END DATA。

RESTORE 称恢复数据区语句，用于将数据指针重新指向第一个 DATA 语句中的第一个数据。

6. END 语句 END 是程序的结束标志，每一个完整的 True BASIC 程序，在其末尾必须有且只能有一个 END 语句。程序执行到该语句时便停止下来。

1.3 结构语句

(一) 条件表达式

条件表达式是由关系运算符和逻辑运算符组成的表达式子，表达式的值是逻辑值，即真或假。表达式的运算符有关系运算符和逻辑运算符两种。

用关系运算符连接起来的式子称关系表达式，关系表达式的值是一个逻辑值，而非一个数值。当关系成立时，其值为真，否则为假。常用的关系运算符及其含义见表 1-3 所示。

表 1-3 关系运算符及其含义

关系运算符	含义	相反运算符	关系运算符	含义	相反运算符
=	等于	< >	< =	小于等于	>
<	小于	> =	> =	大于等于	<
>	大于	< =	< >	不等于	=

在关系表达式中可以出现算术运算，在对关系表达式求值时，先进行算术运算，后进行关系运算。

用逻辑运算符连接起来的式子称逻辑表达式。逻辑运算的对象是逻辑值，即关系表达式。在 IF 语句的条件表达式中，常常需要将几个关系表达式构成一个复合表达式，这时就需要用到逻辑运算符。True BASIC 可以提供三种逻辑运算，即 NOT(非)、AND(与)、OR(或)，其优先顺序依次为：NOT、AND 和 OR。三种逻辑运算的运算规则及示例见表 1-4。

表 1-4 逻辑运算规则及示例

逻辑运算	结果	示例	逻辑运算	结果	示例
NOT T	F	NOT 2<5	F AND F	F	1<1 AND 3>5
NOT F	T	NOT 2>5	T OR T	T	1=1 OR 2=2
T AND T	T	1=1 AND 2>1	T OR F	T	1=1 OR 2<2
T AND F	F	1=1 AND 2<1	F OR F	F	1<1 OR 2<2

TRUE BASIC 中没有逻辑常数，因此不能在表达式中出现“T”或“F”。

(二) IF-THEN 选择结构语句

1. 单行分支语句 语句格式为：

IF <条件表达式> THEN <语句 1> [ELSE <语句 2>]

其执行逻辑是：若条件表达式的值为真，执行语句 1，否则执行语句 2。省略 ELSE 选项时，执行 IF 语句的下一行语句。

【例 1】从键盘输入两个数，按由小到大的顺序打印出来。

```
! exam1-1
input prompt "input a, b = " : a, b
if a > b then print b, a else print a, b
```

end

2. 多行分支语句 在单行分支语句中, THEN 或 ELSE 后只能执行一个语句。而在实际应用中往往要求按条件分别执行一些语句(语句块), 这时就要用到多条件选择结构。多行分支语句是指解决某个问题时要根据多个条件判断, 从各种情况中选择一种。在编写多行分支的 True BASIC 程序时, 可采用 ELSE IF 语句, 以实现多行分支, 其一般形式为:

```
IF <条件表达式 1> THEN
    <语句块 1>
ELSE IF <条件表达式 2> THEN
    <语句块 2>
ELSE IF <条件表达式 3> THEN
    ...
    ...
    ...
ELSE
    <语句块 n>
END IF
```

使用多行分支语句时, 必须注意以下两点:

① IF、ELSE IF、ELSE 和 END IF 必须独自占一行, 且 IF 和 ELSE IF 中不可缺少 THEN, THEN 后面不可再有其它语句;

② IF 和 END IF 必须成对使用, 若在 ELSE 与 END IF 之间无语句块时, 可省略 ELSE。

【例 2】输入 a、b、c 三个值后, 判断能否构成三角形, 若能构成三角形, 计算该三角形的面积, 并显示出来。

```
REM exam1-2 calculate area of triangle
input prompt "input there data : " : a, b, c
if a+b < c then
    print a; " "; b; " "; c; "it is not a triangle"
else if abs(a-b) > c then
    print a; " "; b; " "; c; "it is not a triangle"
else
    let p = (a+b+c) / 2
    let s = aqr(p*(p-1)*(p-b)*(p-c))
    print a, b, c, s
end if
end
```

(三) SELECT CASE 选择控制结构

从上所述, 在编写多行分支程序时, 尽管可使用多重 IF 语句来实现, 但程序结构复杂, 容易产生错误。此时, 可使用 SELECT CASE 条件语句。其语句格式为: