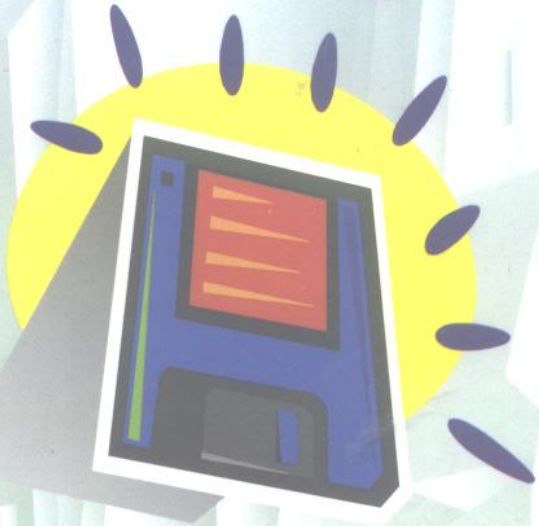
电脑学步丛书之五

程序设计语言 QBASIC

秦戈 编著



电子科技大学出版社

程序设计语言 QBASIC 秦戈 编著

电脑学步丛书之五

程序设计语言

——QBASIC

秦戈　编著

编委　罗筱泠　安钟利　刘勇　冉小兵

电子科技大学出版社

内容提要

本书为《电脑学步丛书》第五册，该书详细介绍了QBASIC这种全结构化和模块化的计算机高级语言，通过大量精选的例题阐述了利用QBASIC语言进行程序设计的基本方法与技巧。本书概念清晰，语言通俗易懂，尤其适于没有程序设计经验的人学习。在介绍程序设计的同时，本书着重突出了算法在程序设计中的重要性。培养读者的算法设计能力及其在程序设计中的应用是本书的一大特点。

本书适用于广大电脑初学者及电脑爱好者，也可作为中专和各类计算机培训班的教材。

程序设计语言——QBASIC

秦戈　编著

出　　版：电子科技大学出版社　（成都建设北路二段四号，邮编：610054）
责任编辑：文　利
发　　行：新华书店经销
印　　刷：峨眉电影制片厂印刷厂
开　　本：787×1092　1/16　　印张　8.375　　字数　230千字
版　　次：1998年1月第一版
印　　次：1998年11月第二次
书　　号：ISBN 7—81043—831—X/TP·348
印　　数：4001—8000
定　　价：9.80元

前　言

计算机科学的崛起，使世界发生了深刻的变化。计算机技术与计算机应用已经深入到社会生活中的每一个角落，了解并掌握计算机的有关知识成为当今社会每个人所必备的一项基本技能。本书正是在这种情况下，为加快计算机知识普及的步伐而编写的。

随着计算机技术的发展，各种各样的应用软件层出不穷，功能也越来越强大，人们几乎可以不需懂有关计算机的任何知识就可以使用这些软件。这样虽然很方便，但同时也带来了一个弊病：那就是使计算机蒙上了一层愈来愈厚重的面纱，人们变得更难了解计算机内部究竟是怎样工作的了。我们建议：在掌握计算机应用的同时，应该学习一些有关计算机的基本知识。而程序设计是计算机应用人员的基本训练和基本功，只有打好了这个基础，我们才能明白计算机是怎样工作的，各种应用软件是怎样设计出来的。

程序设计语言伴随着计算机的发展，也呈现出多样化的趋势，现今得到广泛应用的计算机高级语言已有十几种。其中，BASIC 语言是最适合于广大计算机初学者学习的一种语言，这已被国内外多年的教学实践所证明。

BASIC 语言从 1964 年诞生至今已有三十多年的发展历史了，其本身也演绎出了多种版本。像早先流行的 BASICA 和 GWBASIC 为 80 年代中国大多数的计算机爱好者所熟悉。作为 BASICA 和 GWBASIC 的更新版本，QBASIC 具备了当代计算机高级语言的许多特征：它是完全结构化的语言；区分了全局变量和局部变量，使子程序和函数成为真正独立的模块；提供了良好的编辑环境，同时支持键盘和鼠标的操作；提供了“分步执行”和“跟踪”等程序调试工具；提供了内容丰富的联机“帮助”系统。因而，QBASIC 作为第三代的 BASIC 语言，已经得到了广泛的承认和推广。

本书共分八章。第一章介绍了程序设计的基本概念和方法以及怎样进入 QBASIC 环境；第二章介绍了 QBASIC 语言的基本规则；第三、四、五章分别介绍了顺序、选择、循环这三种基本控制结构的程序设计方法；第六章介绍了函数和子程序的概念及其应用；第七章介绍了数组这一重要的数据结构及应用时的一些技巧；第八章介绍了字符串的处理方法及一些应用实例。

我们希望广大读者在通过本书的学习后，不仅能掌握 QBASIC 语言的程序设计，而且能对“算法”在程序设计中的重要性有所体会，并且可以举一反三，在今后学习其它计算机高级语言时能有触类旁通的感觉。

作者

1997 年 12 月

目　录

第一章 认识 QBASIC

☆ 语言与程序
☆ 流程与算法
☆ QBASIC 环境简介

1.1 关于计算机语言

1.1.1 计算机与程序

20 世纪 40 年代，世界上第一台电子计算机“ ENIAC ”诞生了，全机用了电子管 18 000 个，继电器 1 500 个，耗电 150kW，运算速度每秒钟 5 000 次，占地达 1 800 平方英尺。从第一台计算机问世至今，短短五十年的时间，电子计算机的发展已经从最初的每秒钟运算几千次达到了现今的运算速度超过每秒钟十亿次，体积也从最初的占地达几个房间减小到现在的可以像笔记本一样地装入口袋中随身携带。大规模以及超大规模集成电路的广泛应用加速了电脑的发展步伐，同时也极大地降低了电脑的生产成本，以至于现在微型电子计算机，也就是俗称的个人电脑，已经像日常的家用电器一样深入到了社会生活中的每一个角落。

电子计算机发展得如此迅猛，那么它究竟能够为人类干些什么，又是怎样去干的呢？简单地讲，人类大多数的不具创造性的工作都可以由电子计算机来完成。完成这些工作，计算机依靠的是人们事先编制好并输入机内的程序。

人们做任何事情都有一定的步骤。利用计算机完成一项工作也有一定的步骤：事先对问题进行分析→确定解决问题的方法与步骤（即算法）→根据算法编制出计算机程序→让计算机执行这个程序，得出最后的结果。计算机操作的每一步都是在相应的命令（这里称为指令）操纵下进行的。不同的指令序列，可以操纵机器完成不同的工作。我们把指令的有序集合就称为程序。当用电子计算器进行数学计算时，程序就存在于人们的大脑之中，大脑发出一条指令，计算器就执行一步相应的操作。而要使电子计算机实现自动操作，就必须把程序从人脑中分离出来，让计算机记住，并按程序的规定有顺序地执行一系列指令。这就是通常所说的程序控制，现在所有的计算机都是在程序控制下进行工作的。

1.1.2 计算机的编程语言

既然计算机是在人们预先编制的程序下进行有目的的工作，那么，人们是用什么形式把程序“告诉”计算机的呢？

通常把表示和交流信息的符号系统及规则称为“语言”，并把给计算机编制程序的符号系统及规则称为计算机的编程语言，简称程序语言或计算机语言。程序语言随着计算机

的发展，由低级到高级不断完善、逐步简化和方便。目前计算机的语言大致可分为机器语言、汇编语言和高级语言三大类。

机器语言和汇编语言都是低级语言，是一种面向机器的语言。用这些语言编制的程序具有占用内存少、执行速度快等优点。然而他们的直观性和可读性较差，而且受具体机器指令系统的束缚，编程效率很低。随着计算机技术的高速发展，人们设想能够寻求一种可摆脱机器指令系统的束缚、通用性强、又比较接近人的语言习惯的计算机编程语言。高级语言的开发成功使人们的这种设想变成了现实。高级语言是一种面向过程或是面向对象的语言，使用这种语言编程，基本上摆脱了机器指令系统的约束，程序员无须再花很大的气力来了解机器内部的逻辑结构，可以集中精力去推敲解题的思路。但高级语言与低级语言相比，占用的内存较大，执行速度较慢，某些情况下还不能代替低级语言，这与计算机硬件的发展不同，编程语言的发展不是高一代取代低一代，而是多代共存的。

高级语言从 50 年代中后期发展到今天，全世界大约已经有了一千多种，并且新的语言还在不断出现。但这些语言中大多数都是一些专用语言，真正在国际上得到广泛应用的只有十几种，如 BASIC 、 FORTRAN 、 COBOL 、 PASCAL 、 C 、 ALGOL 68 、 LISP 及 PROLOG 等等。

高级语言编写的程序只有经过翻译才能执行，根据翻译的方式，可分为编译型和解释型两类。无论是编译方式还是解释方式，都有一个编译程序或解释程序被事先放在机器中。编译方式是源程序输入计算机后，编译程序将它们全部编译成目标程序后才交机器执行；解释方式是源程序输入后，解释程序对源程序进行逐句翻译，翻译一句交机器执行一句。解释方式比编译方式占用内存少，但执行速度较慢。

1.1.3 算法的概念

人们做任何一件事情都有一定的步骤。同样，计算机也是按程序所规定的内容和步骤进行工作的。因而，在计算机上为解决一个问题而进行程序设计时，至少应该具备以下两个条件：

- 首先，必须掌握解决问题的方法和步骤。也就是说，在拿到一个需要求解的问题后，经思考先确定其解决方法，然后将其分解成一系列可实施的操作步骤；
- 其次，应至少掌握一门高级语言，以使能将所确定的解决问题的方法和步骤通过程序设计来告诉计算机。

为解决一个问题而采取的方法和步骤，就是我们通常所说的“算法”。广义地说，处理任何问题都有一个“算法”问题。解决一个问题的过程就是实现一个算法的过程。例如，一个菜谱就是一个“算法”，厨师炒菜就是实现这个“算法”。所以，利用计算机解决一个问题时，实际上就包括了算法的设计和算法的实现两部分工作。可以说，程序设计的灵魂就是算法的设计，而语言只是表达算法的一种形式。有了正确的算法，可以利用任何一种语言来编写程序，使计算机工作，得出正确的结果。

对同一个问题，往往有不同的解题方法。例如要计算：

$1+2+3+\cdots+100$，即 $\sum_{n=1}^{100} n$ 的值，就有几种不同的方法。可以先进行 1+2，再加 3，再加 4，一直加到 100，得到结果 5050。也可以采取另外的办法，$\sum_{n=1}^{100} n = 100+(99+1)+(98+2)+\cdots+(51+49)+50 = 100+49\times100+50 = 5050$。当然，还可以有其它的许多方法。人们希望采用好的算法，即方法简单、运算步骤少，又能迅速得出正确结果的

算法。因此，为了有效地解决问题，不仅需要保证算法正确，还要考虑算法的质量。用计算机解决问题，则要选择适合计算机的算法。请本书的读者把注意力重点放在算法的设计上，而不是局限在某一种具体的计算机语言上。掌握了算法的设计，无论用哪一种语言编写程序都不会太困难。

计算机算法可分为两大类：数值运算算法和非数值运算算法。数值运算算法的目的是求数值解。例如求方程的根、求函数的定积分等，都属数值运算范围。而非数值运算算法包括的范围则很广，最常见的是用于管理领域，如人事档案管理、图书情报资料检索、航空调度等。很多算法现在已经研究得比较成熟，特别是数值运算方面的算法。而非数值运算的算法由于种类比较多，难以规范化，因此往往需要使用者参考已有的类似算法重新设计解决特定问题的专门算法。

1.1.4 N-S 结构流程图

设计一种算法，必然要用一种方式将其表示出来，N-S 结构流程图则是表示算法的一种较为理想的形式。它是在 1973 年由美国学者 I.Nassi 和 B.Shneiderman 提出的。在这种流程图中，全部算法写在一个大矩形框中，在该框内还可以包含一些从属于它的小矩形框。换句话说，它是由一些基本的矩形框组成的一个大的矩形框，而且只有三种基本的元素框，分别表示算法的三种基本结构。

1966 年，Bohra 和 Jacopini 提出了算法的三种基本结构：

（1）顺序结构

（2）选择结构（或称分支结构）

（3）循环结构（或称重复结构）

三种基本结构是一个良好算法的基本单元。已经证明，由以上三种基本结构顺序组成的算法，可以解决任何复杂的问题。由基本结构构成的算法属于“结构化”的算法，它不存在无规律的转移，只有在本结构内才允许存在分支或向前向后的跳转。这种由基本结构顺序组成的算法便于阅读和修改，从而使算法的可靠性和可维护性得到了保证。

在 N-S 结构流程图中用以下的一些流程符号来表示算法的三种基本结构。

1、顺序结构

图 1.1 是用 N-S 流程图表示的顺序结构，其中 A 和 B 两个框内的操作是顺序执行的。

2、选择结构

如图 1.2 所示，当 P 条件成立时执行 A 操作，P 不成立时则执行 B 操作。注意，图 1.2 所示的是一个整体，代表一个基本结构。

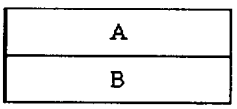


图 1.1 选择结构

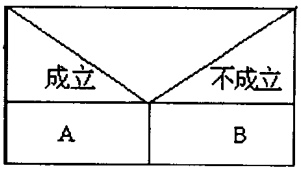


图 1.2 选择结构

3、循环结构

循环结构的作用是反复执行某一部分操作。它有两类循环结构：当型循环和直到型循

环。

（1）当型循环（WHILE 型循环） 当型循环的特点是：当指定的条件满足时，就执行循环体，否则就不执行。当型循环又分两种形式：

①“前测试”型 先测试条件，后执行循环体。如图 1.3（a）所示。在执行如图所示的循环结构时，先检查 P1 条件是否成立，如果成立就执行循环体 A。然后再检查 P1 是否成立，若成立再执行 A，如此反复直到某一次 P1 条件不成立时，就不再执行 A，循环结束。

② “后测试”型 先执行循环体，然后再测试条件是否成立，如图 1.3（b）所示。从图中可以看出：先执行循环体 A，然后才测试条件 P1 是否成立，如果 P1 不成立，则不再返回执行 A。由此可知，后测试当型循环至少执行循环体一次（前测试的当型循环，若开始时条件 P1 不成立，则一次也不执行循环体）。这是前测试型和后测试型的区别。

当型循环可以概括为：“当 P1 条件满足时，反复执行循环体”。

（2）直到型循环（UNTIL 型循环） 直到型循环的特点是：执行循环体直到指定的条件满足，此时就不再执行循环。

直到型循环同样分为“前测试”型和“后测试”型。

①“前测试”型先测试条件，后执行循环体，如图 1.3（c）所示。从图中可以看到：执行此循环结构时，先测试条件 P2 是否成立，如成立，就不执行循环体 A，如不成立就执行循环体 A。

②“后测试”型 先执行循环体，然后再测试条件是否成立，如图 1.3（d）所示。从图可知其执行过程为：先执行一次循环体 A，然后再检查条件 P2 是否满足，如果不满足就重复执行循环体 A，然后再对 P2 条件作测试，如果 P2 仍不满足，又执行循环体 A……直到 P2 条件满足，就结束循环过程。

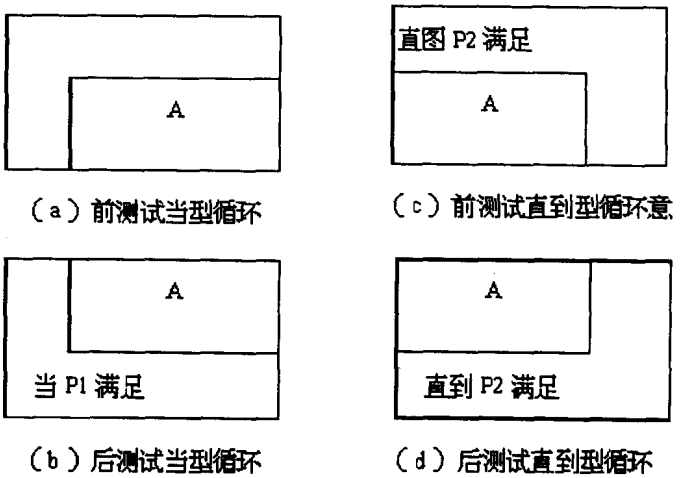


图 1.3

直到型循环可以概括为：“反复执行循环体，直到 P2 条件满足。”

图 1.1 至图 1.3 中的“A”或“B”可以是一种简单的操作，也可以是三种基本结构之一。基本结构之间是可以互相嵌套的，在一个基本结构中又可以包含一个或多个基本结构，例如在循环结构中可以包含一个选择结构。

后测试直到型循环 N-S 流程图尤其适合于表示一个结构化的算法，用于结构化的程序

设计。

[例 1-1] 求 1 × 2 × 3 × 4 ×…× n，即 n!，直到 n!≥ 1000 为止。用 N-S 流程图表示该题的算法。

分析该题，设一个变量 t 用来存放求出来的阶乘值，在累乘的过程中 t 的值是变化的。另设一个变量 i，用来存放累乘数 1，2，3…算法可用图 1.4 表示。可以看出，这是一个后测试直到型循环：先执行循环体，然后检查“t ≥ 1000”条件是否满足，满足就循环。它可以概括为：执行阶乘操作，直到条件“t ≥ 1000”成立为止。这个算法由三个基本结构顺序构成，分别是顺序结构 A →循环结构 B →顺序结构 C 三个部分。在 N-S 流程图中，基本元素框在流程图中的上下顺序就是执行时的顺序，也就是:图中位置在上面的先执行，位置在下面的后执行。写算法和看算法只需按从上到下的顺序进行就可以了，十分方便。在本书后几章中，基本上都采用 N-S 流程图表示算法。

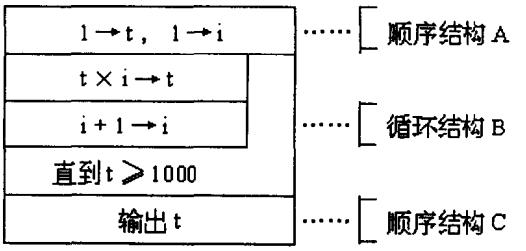


图 1.4

1.2 BASIC 语言及其基本特点

BASIC 是 Beginner's All-purpose Symbolic Instruction Code（初学者通用符号指令代码）的缩写，最初的 BASIC 版本是由美国新罕布什尔州达特蒙斯（DARTMOUTH）学院的 J.基米尼（John G Kemeny）和 T.库尔茨（Thomas E Kurth）两位教授于 1964 年提出的。经过 30 年来的扩充和改进，BASIC 语言已成为国际上广泛使用的一种高级语言。

尽管不同版本的 BASIC 语言互有差异，但其基本特点是共同的：

• 简单易懂。BASIC 语言最初是为教学目的而从 FORTRAN 语言中提炼出来供初学者使用的一种计算机编程语言。它的语法关系简单，核心语句只有十几句。它所使用的命令、语句中的词汇、符号与英文单词及数学中的符号差不多，书写规则也不复杂，容易理解，便于记忆。

• 具有交互性。BASIC 语言是一种会话式语言，采用了解释的方式来对源程序进行翻译。当用户把源程序输入计算机并命令机器执行程序时，解释程序便会逐句地顺序检查程序。一旦发现语法错误便给出相应的错误信息，指出错误的类型和位置，提示用户在机器上边执行边修改，直到得出正确而满意的结果为止。这种人机对话的方式，称为 BASIC 语言的交互性。它能为初学者提供良好的学习环境。

占用的内存空间小。解释方式带来的另一个特点就是占用内存空间小，特别适合于内存量较小的微型计算机。由于解释方式工作时机器的执行速度较慢，所以 BASIC 语言对机器本身的运行速度就没有太高的要求，在普通的 386 机型上运行小型 BASIC 程序与在速度更高的机型上相比，没有明显的差别。也正是因为这个原因，BASIC 语言不适合解太大或速度要求较高的题目。不过它对于普通科学、技术和事务中的大量问题还是够用的。

现在许多 BASIC 语言的新版本中都增加了字符处理、图形、音乐、文件以及调用汇编语言子程序等功能，使 BASIC 语言的应用更加广泛。从不同角度所作的改进，致使 BASIC 语言形成了多种不同规模、不同风格的版本。本书以 IBM PC 系列微机上使用的 QBASIC 为蓝本，介绍 BASIC 语言程序设计的基本方法和基本技巧。

1.3 进入 QBASIC 的世界

QBASIC 语言是 Microsoft 公司开发的，作为 MS-DOS 5.0 及以上版本的一个组成部分免费提供给用户的。也就是说，如果计算机上已经安装了 DOS 5.0 或更高的版本，就可以自由地使用 QBASIC 软件了，不需再另外购买此软件。

QBASIC 是 BASIC 语言发展到近阶段的产物。它是 BASICA 和 GWBASIC 的升级版，并与之保持了高度的兼容性，用 BASICA 和 GWBASIC 语言编写的源程序可不加修改（或仅作很少的修改）就能在 QBASIC 环境下运行。

QBASIC 是现代化和结构化的 BASIC 语言。与 BASICA 和 GWBASIC 相比，它在以下几个方面扩充了功能：

- 扩充了变量和常量的类型；
- 提供了新的选择结构，增加了块 IF 语句和实现多分支选择结构的 SELECT CASE 语句；
- 改进了循环结构，增加了 WHILE 型循环和 DO 循环；
- 区分了全局变量和局部变量，使子程序和函数成为真正独立的模块；
- 不需要行号，语句按其排列的顺序执行。但 QBASIC 也允许使用行号，作为转移指向的标志，但行号不反映执行顺序；
- 提供良好的编辑环境，同时支持键盘和鼠标的操作；
- 提供“联机”帮助，在编程过程中可以要求计算机随时提供“帮助”信息；
- 提供“分步执行”和“跟踪”等程序调试工具；
- 采用了先进的解释执行方式，使程序运行比一般 BASIC 快得多。

1.3.1 QBASIC 的启动

如果计算机上已安装了 MS-DOS 5.0 或更高的 DOS 版本，那么在子目录 DOS 中就已包含了 QBASIC.EXE 和 QBASIC.HLP 两个文件，这就是启动 QBASIC 环境所需的两个支持文件。开机后，使当前盘为 C 盘，然后按以下步骤启动 QBASIC：

C:\>CD\DOS ↵　　（进入 DOS 子目录）

C:\DOS>QBASIC ↵（将 QBASIC 调入内存）

划线部分是从键盘输入的内容，“↵”符号代表键盘上的“Enter”键。注意，在上面的显示中，执行完第一步操作后，计算机显示出当前的子目录为 C:\DOS>，这样使用户能够清楚地知道自己是处在哪一层目录下工作。但有些用户使用的计算机系统上未作“显示路径”的设置，因此不显示出子目录名，而只显示出当前盘符：

C:>

但用户的操作是一样的，只是必须自己记住是工作在磁盘的哪一层目录下。

在进入了 DOS 子目录后，可以用 DIR 命令检查一下是否已有 QBASIC 系统。

C:\DOS>DIR Q*.*　　（列出所有以字母 Q 打头的文件目录清单）

此时在屏幕上将显示出 DOS 子目录下所有以字母 Q 打头的文件的目录清单（包括文件名、文件的大小及文件的建立时间），其中应该有两个 QBASIC 的支持文件：QBASIC.EXE 和 QBASIC.HLP，显示形式如下：

```
QBASIC    EXE    194309   05-10-93 6:00a
QBASIC    HLP    130881   05-10-93 6:00a
```

其中 QBASIC.EXE 是 QBASIC 解释程序的主体（执行文件），由它实现 QBASIC 的解释执行功能。QBASIC.HLP 是联机帮助文件，它向用户提供“帮助”信息，说明 QBASIC 语言的语法规则以及 QBASIC 编辑环境的使用方法。

如果读者使用的计算机上没有 QBASIC 的这两个支持文件，可以自己将这两个文件复制到所用的硬盘上。最好先建立一个子目录，目录名可以任取，例如就用“ BASIC ”，把 QBASIC 文件放在此子目录下，以免与其它的应用文件相混淆。假如 A 盘上已装有 QBASIC 的两个文件，可按下面的步骤将 QBASIC 复制到 C 盘上：

C:\>　　（设当前为 C 盘根目录状态）
C:\>MD BASIC ↵　　（在 C 盘根目录上建立 BASIC 子目录）
C:\>CD BASIC ↵　　（进入 BASIC 子目录）
C:\BASIC>COPY A:QBASIC.* ↵　　（将 A 盘上以 QBASIC 作文件名的所有文件复制到 C 盘的当前目录下）
C:\BASIC>QBASIC ↵　　（执行 QBASIC.EXE 程序，进入 QBASIC 环境）

此时屏幕上将出现 QBASIC 的工作窗口。

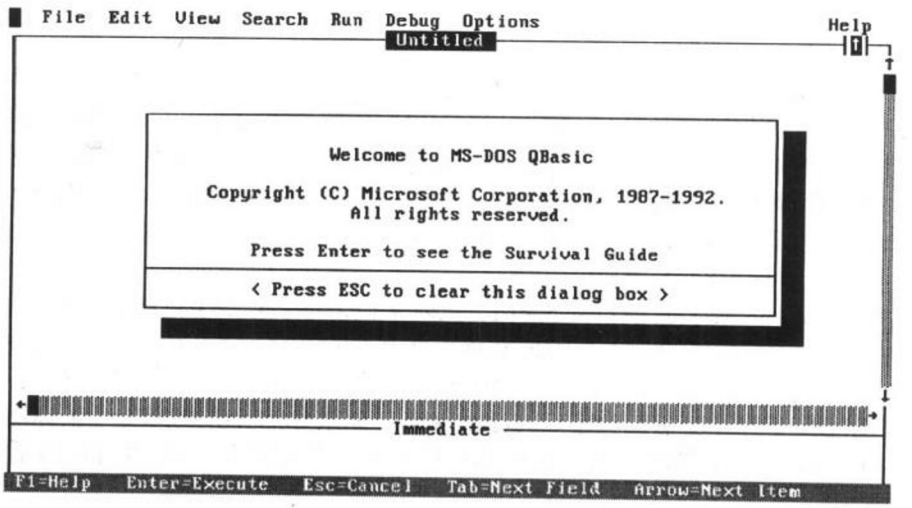


图　1.5

1.3.2　QBASIC 环境的组成

在进入 QBASIC 环境后屏幕上将出现如图 1.5 所示的画面。

这时有两个选择：

- 按 Esc 键清屏，进入正常的程序编辑环境；
- 按 Enter 键调用联机“帮助”系统，从中可知道一些有关 QBASIC 的使用说明。

现在按 Esc 键，屏幕显示如图 1.6 所示。

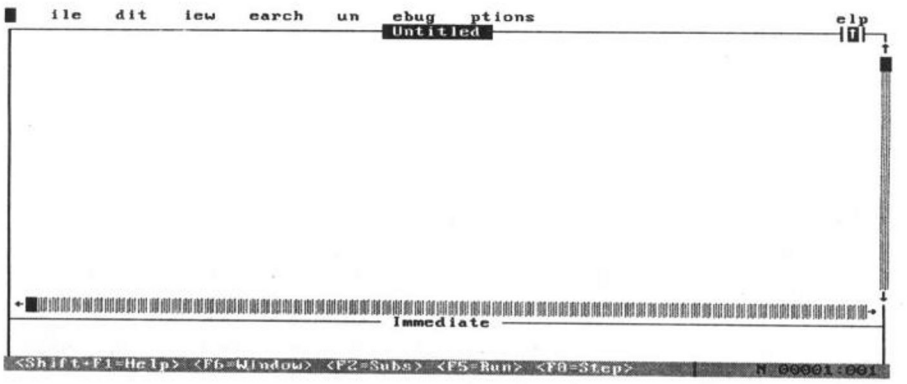


图 1.6

从图 1.6 可以看到， QBASIC 在正常的程序编辑环境下提供了两个工作窗口。上窗口为“程序窗口”，用来编辑和运行程序，在 QBASIC 未从磁盘调入源程序或新程序尚未起名时，程序窗口的标题为“ Untitled ”(意为无标题)，反之则为当前源程序的文件名“*.bas ”。下窗口称为“命令窗口”或“直接窗口”，用来直接执行单个的命令。下窗口的标题是“ Immediate ”（意为立即）。例如在下窗口中键入命令“ PRINT 2*5 ”，按 Enter 键后马上执行此命令，并输出结果 10 。

任何时候都不能同时使用这两个窗口，也就是说，两个窗口中只能有一个是被激活的。判断当前窗口是否被激活的标志是闪烁光标处在那个窗口或该窗口的标题部分被加亮，只有在被激活的窗口才能输入信息，无法对未被激活的窗口进行操作。

使用功能键 F6 可以切换两窗口的激活状态。如果你的计算机配备了鼠标器，那么直接用鼠标在两窗口的有效输入区域内任意点取亦可很方便地完成对两窗口激活状态的切换。

下面介绍一下整个 QBASIC 工作窗口内的各项显示信息。

• 光标　处于激活状态的窗口中有一闪烁的短线，它用于标识即将输入的字符出现的位置。每输入一个字符，光标自动右移一个位置。可以用键盘上的 4 个方向键（→、←、↑、↓）或是鼠标直接点取的方式来任意移动光标，使其在整个工作窗口的有效范围内移动。

• 光标的位置显示　在整个 QBASIC 工作窗口的右下角（图 1.6 的右下角）有两个数字，代表光标的当前位置，第一个数字代表光标所在的行号，第二个数字代表光标所在的列号。它使能够清楚地知道当前将要输入的字符在屏幕上的确切位置。

• 菜单显示区　从图 1.6 中可以看到，在 QBASIC 工作窗口的最上端有一条被加亮的“菜单条”，在该菜单条内列出了所有 QBASIC 提供的菜单项： File（文件操作菜单项）、Edit（编辑操作菜单项）、 View（观察设置菜单项）、 Search（搜寻操作菜单项）、 Run（运行操作菜单项）、 Debug （调试操作菜单项）、 Options （选项设置菜单项）。所有菜单项均为下拉式菜单，用鼠标直接点取或按 Alt 键均可将其激活。

• 信息显示区　在如图 1.6 所示屏幕的最下部有一被加亮的信息显示条，其中显示了各功能键的使用说明或是各命令的操作说明。注意，信息显示区的内容随当前工作窗口的状态不同或是当前所选取的命令的不同而不同，请读者在上机操作时注意比较。

• 程序窗口的缩放　在如图 1.6 所示程序窗口的右上角有一被加亮的向上箭头，操作此箭头可使程序窗口放大至满屏而覆盖命令窗口，此时，向上的箭头变成了双向箭头，如

图 1.7 所示。在键盘上连续按 Ctrl+F10 （即同时按住 Ctrl 键和 F10 键）或是单击鼠标左钮连续点取该箭头均可完成程序窗口缩放的切换。

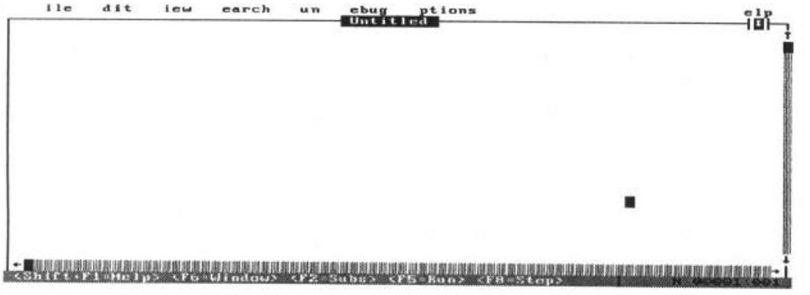


图　1.7

1.3.3　一段简单的程序

下面输入和运行一个简单的 QBASIC 程序，以使能够对 QBASIC 的使用环境和 QBASIC 程序的组成有一个具体的认识。

进入 QBASIC 环境后，激活程序窗口，并使光标处于第一行第一列的位置。按照下列的格式输入以下程序：

```
let a=2 ↵
let b=5 ↵
let c=a*b ↵
print “ C=”； c ↵
end ↵
```

输入完成后，屏幕显示如图 1.8 所示。

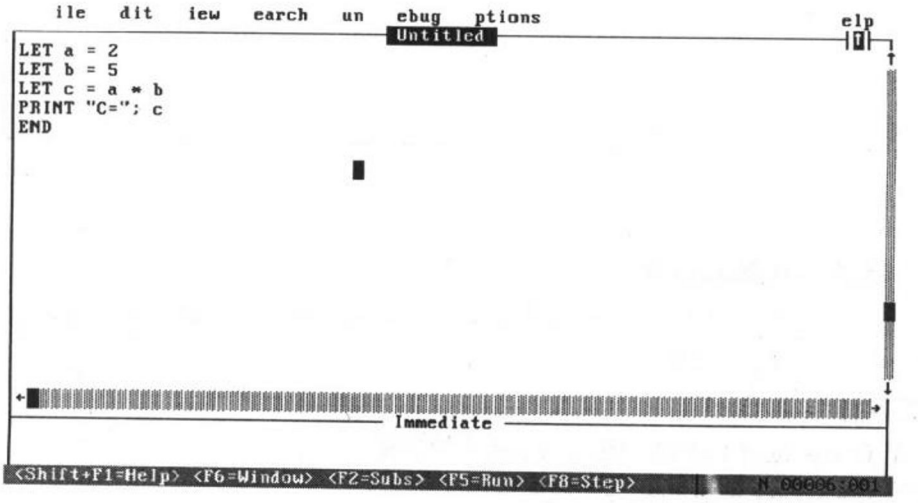


图　1.8

用户是否已注意到，当你输完一行并按 Enter 键时，所有小写的“ let ”、“ print ”和“ end ”都自动变成了大写的“ LET ”、“ PRINT ”和“ END ”，并且连着写的“ a=2 ”，“ b=5 ”和“ c=a*b ”之间都自动地加上了空格，使语句看起来更加简洁清楚，便于阅读。