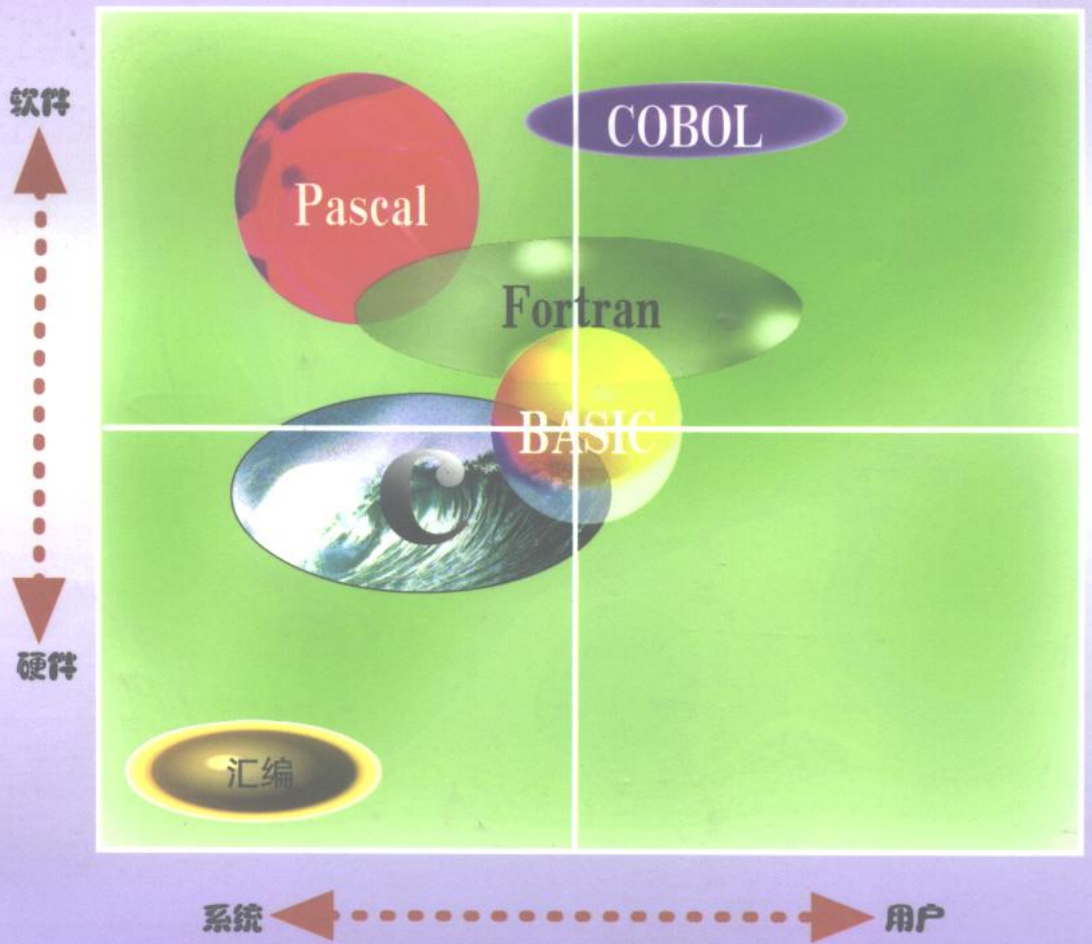
按国际语言教学新经验编写的

半 双 语 系 列 教 材

Semi-Bilingual Series Course

从BASIC跃到C

潘正伯 瞿燕 编著



利用你已有的BASIC知识

帮 助 你 更 快 地 掌 握 C

北京大学出版社

内 容 简 介

本书是为有 BASIC 基础的读者编写的一套学习程序设计语言的教材中的一本(另外两本是《从 BASIC 跃到 Pascal》,《从 BASIC 跃到 Fortran》)。本书详细讲解 C 语言的基本知识,在写法上采用类似外语教学中的双语教学,通过对比方式让已会 BASIC 语言的读者学会把用 BASIC 语言编写的程序转换成 C 语言的形式。比较两种语言编写的程序之间的差异,引导读者进一步思考如何发挥 C 语言的特长和优势,从而更快捷、更深刻地掌握 C 语言,达到能类旁通的效果。

本书是作者总结多年教学经验所作的教学改革的尝试,适合作为大专院校程序设计课教材,也适于计算机语言初学者和有一定 BASIC 语言基础从事计算机工作的科技人员学习、参考。

图书在版编目(CIP)数据

从 BASIC 跃到 C/潘正伯,瞿燕编著. —北京:北京大学出版社,1998.1
ISBN 7-301-03621-3

Ⅰ.从… Ⅱ.①潘… Ⅲ.C 语言 Ⅳ.TP312

中国版本图书馆 CIP 数据核字(97)第 26813 号

书　　名:**从 BASIC 跃到 C**
著作责任者:潘正伯　瞿　燕
责 任 编 辑:段晓青
标 准 书 号:ISBN 7-301-03621-3/TP·386
出　版　者:北京大学出版社
地　　址:北京市海淀区中关村北京大学校内　100871
电　　话:出版部 62752015　发行部 62754140　编辑部 62752032
排　版　者:北京市兴盛达激光照排中心
排　印　者:北京大学印刷厂
发　行　者:北京大学出版社
经　销　者:新华书店
787×1029　16 开本　20.125 印张　550 千字
1998 年 5 月第一版　1998 年 5 月第一次印刷
定　　价:28.00 元

序

世界上现今一共有多少种程序设计高级语言已经很难说清。早在1984年《科学的美国人》出的计算机软件专辑中就曾说过："程序设计语言加上它们的'方言'为数至少有几百种，甚至可能有几千种。"[①] 11年后，《BYTE》杂志创刊20周年特刊中发表了20篇综述文章，在"程序设计语言发展简史"一文中，列举出从1946到1995年为止，在社会上产生了一定影响的四十几种程序设计语言[②]。这些语言中的一部分传到我国，几经筛汰，有些随着教学和应用逐渐得到普及，其中最常见、最流行的莫过于微机上的BASIC，Pascal，Fortran和C等几种。

这几种高级语言中，BASIC被公认为是易学易用、发展快、变化大、覆盖面广的初学者语言。自80年代初，在美国出现兼具解释和编译功能的结构化BASIC（即我们称之为**第二代BASIC**的True BASIC，Turbo BASIC和Quick BASIC）以来，BASIC在美国成了最受欢迎的程序设计语言[③]。美国人对它的评价是"Low threshold，high ceiling"，其直译是"门槛低，天花板高"，即入门易而潜力大的程序设计语言。在三田典玄先生提供[④]而被我们选作本书封面的图案中，读者可以看到BASIC语言兼顾了软件和硬件、系统和用户四大领域的应用，几乎成了其他任何一种语言无法代替的语言。

Windows风行后，程序设计的难度陡然提高，几乎酿成广大业余程序员的灭顶之灾。在此期间，一批甩开TEXT状态，采用GUI状态，能在Windows(OS/2)环境下运行的BASIC（即被称为**第三代BASIC**的Visual Basic，CA-REALIZER，GFA BASIC和Power BASIC等）恰如雨后春笋，应运而生，于危厄中解救了一大批非软件专业程序员，使他们很快就能在Windows(OS/2)提供的广阔天地中纵横驰骋，开发他们所熟悉专业的应用软件。只有各专业的应用软件极大的丰富了，才会有计算机应用的真正繁荣。

因此，近七八年来，我们一直呼吁社会各界（尤其是教育界）要重视BASIC的推广与应用，抓住Quick BASIC与第三代BASIC有最好的兼容性这一特点，以Quick BASIC为计算机专业与非计算机专业学生的入门语言，通过两代BASIC的衔接，解决我国大中专学生和广大计算机用户从DOS向Windows(OS/2)的过渡这一棘手问题。经过三年实践检验，证实了这一方案的有效性。

但是，这绝不意味着我们主张"一花独放"——用BASIC去贬低或取代其他几种常用语言，绝非如此！

每种经受住时间检验，得到广泛使用的程序设计语言，都有它特定的目标和存在的价值，都有自己的特长和不足。BASIC的特点决定了它是很理想的初学者语言；至于从事繁复的科学和工程计算，则应首推Fortran；结构严谨的算法描述，则属Pascal之所长；而构筑具有良好可移植性的系统，又是C语言的拿手好戏。为了适应多方面的需要，一个好的程序员掌握好几

---

① 科学（中译本）计算机软件专辑1985 No. 1 21－30页。
② BYTE Vol. 20，Num. 9，121－122 PP. Sept. 1995。
③ BASIC仍是最常用的编程语言，计算机世界，1987年8月23日。
④ 《实习C语言》三田典玄，アスキ-出版局，1987。

种汇编和几种高级语言势在必行。这里就发生了先学的BASIC与后学的其他各种高级语言之间的关系问题。

这问题在国内似乎从未有人提出过，翻翻案头上、书店里、书库中各种各样讲授高级语言的教材就会发现，尽管编者不同、出版社不同，教材有新有旧，有详有略，但有一点是共同的：它们都假定自己的读者是计算机程序设计语言的初学者，每本教材都是孤立地讲授自己这种语言，而且都是从零开始的。

人们获取知识的最好方法莫过于充分利用他已有的知识，通过对比进行学习。一个好的BASIC程序员，他的BASIC知识和编程经验，是非常宝贵的，这是他掌握其他程序设计语言的一块很好的基地和跳板。更何况几种常用高级语言，在漫长的发展过程中互相渗透、互相借鉴，逐渐形成了你中有我、我中有你的局面。其中变化最大的是BASIC语言，它最初是从Fortran脱胎而来，以后陆续吸收了其他语言的诸多特点，在程序设计语言种族中，它是一个不折不扣的"混血儿"。国外某些计算机教育专家早已注意到了这点，他们发现已掌握BASIC的人可以很快地学会任何一种(哪怕是很晦涩的)程序设计语言。

Robert J. Traister在他的《从BASIC一步跳到C++》一书的"前言"和"跋"里分别写下了这样两段话：

> 你的BASIC编程知识，在你学习C++和其他任何一种计算机程序设计语言时，是你的无法估价的好帮手①。
>
> 如果你会用BASIC编写程序，那么你将很快就能用C++写出同样的程序来。带上你所有的BASIC知识踏上C++的征途吧，你一定会很快适应这种高效编程环境。很快你就会非常乐意并且轻松愉快地用C++这种语言去处理你面临的各种任务，就像你今天用BASIC去处理它们一样②。

我们对此深有同感。经过两年多的酝酿筹划，设计并编写了这套系列教材，它们是：

- 从BASIC跃到C
- 从BASIC跃到Pascal
- 从BASIC跃到Fortran

在众多的程序设计语言教材中，这套教材的特点是力求在两种语言(一主，即读者要学的新语言，如Fortran或Pascal或C；一副，即读者已经掌握的BASIC语言)的对比中，更快捷、更深刻地去掌握新语言。两种语言相同之处一笔带过，相似之处加以区别，赢得篇幅和课时去仔细讲授不同语言的独特处，把功夫下在这些地方，以求取得互相对比、相得益彰、事半功倍的效果。

本系列教程的任务：首先是教会读者将他能用BASIC写出的程序，转变成另一种语言(Pascal或Fortran或C)形式。这任务比起教一批对程序设计一无所知的初学者学会一门程序设计语言，显然要容易得多。在此基础上，本系列教材将引导读者进一步考虑：如何发挥这种语言(Pascal或Fortran或C)的特长和优势，用它写出在它专长的领域内功能更强、效率更高的程序来。

---

① Robert J. Traister, Leaping from BASIC to C++, Epilogve 357 PP. AP PROFESSIONAL 1994

② Robert J. Traister, Leaping from BASIC to C++, Preface XVI PP. AP PROFESSIONAL 1994

这套教材彼此间没有横的联系，可根据需要选其中任何一种，也可供已掌握 BASIC 程序设计者自修 Pascal 或 Fortran 或 C 语言之用。这套教材在编写时均采用当前微机上广泛流行的软件，同时考虑了这些软件的升级版本。书中例题均经过上机检验，准确无误。

双语教材甚至半双语(Semi-Bilingual)教材在外语教学中已得到应用，取得了好的效果。本系列教材在计算机程序设计语言的教学中，引入类似的构想，希望也能取得好的效果。

诚挚的盼望广大读者尤其是有志于程序设计语言教学改革的教师，选用本教程作教材，将您在使用中发现的问题、意见和感想写给我们，让我们携手合作，共同来推动并完善这项改进。

编者们怀着崇敬真挚的心情，感谢北京大学出版社的领导及责任编辑段晓青，深深感谢他们在当前科技书出现的困境中，帮助我们将多年的理想变成了现实——这套教材能够摆在读者面前，接受社会实践的检验。

编者们向在美国的杜殿海、曾瑞华、陈涌和在加拿大的陈贺平诸位先生表示感谢，感谢他们不断地提供国外的有关信息、新书和资料。

本系列教程参考了诸多国内外书籍、论文和资料。在此向所有作者表示感谢。

编者们虽已竭尽努力，终因知识和经验所限，教材中难免出现这样那样的错讹，恳请读者指正，不胜感谢！

潘正伯

1998 年 4 月

前　言

这是一本按照“总序”所阐明的构想，为有 BASIC 基础的读者编的讲授 C 语言的教程。

C 语言从诞生至今这 20 多年的历史，已经充分向人们展示了它在计算机软件开发中的重要地位。一个从事计算机系统软件或应用软件的开发研制人员，C 大概是他(她)手中最常用的工具。

C++早已成了开发大型软件的环境，掌握了 C 就在很大程度上进入了 C++，至少会减少学习 C++深层内核的困难。

编写本书时，主要针对当前微机上用得最多的 C 编译系统(Turbo C 和 Quick C)。Quick C 与 Quick BASIC(简称 QB)都是 Microsoft 公司产品，因而具有非常相似的介面(窗口，菜单等)和操作命令，QB 用户进一步学习 QC，在安装、操作等方面几乎接近轻车熟路，无形中得到很多方便。Turbo C(简称 TC)是 Borland 公司的优秀产品，很有特色。用户中有偏爱 QC 的、有偏爱 TC 的，在介面和操作命令方面，二者的差异也不是很大。本书两者都将顾及。

C 语言的品种很多，但因早已存在 C 语言的美国国家标准(ANSI 1983)和国际标准(ISO/IEC 9899:1990)，不同厂家开发的 C 编译器都在努力向这两个标准靠拢，使得不同品种 C 之间的本质差别越来越少。本书编写的出发点是：讲解介绍那些符合 ANSI 和 ISO 标准，因而是 QC 和 TC 所共有的内容，忽略它们之间的微小差别。目的是想使读者尽快将 C 语言用起来，在使用中自会熟悉无关宏旨的微小差别。

由于本书读者已有 BASIC 编程经验，这就给本书编者提供了较大的自由度。当我们在介绍前面的某些语法成分时，有时不得不用到后面介绍的某些语法成分。希望读者通过对全书的翻检并凭借自己在 BASIC 编程中获得的经验，不会构成学习中的障碍。

C 的内容是很丰富的，希望读者在掌握了本书内容之后，根据自己面对的课题，寻找更专门的书籍学习。

读者若发现本书错讹，恳请批评指正。

作　者

1998 年 2 月

目　录

第一章 概 述

1.1 C语言的由来与发展

C语言的出现是与UNIX操作系统紧密联系在一起的。

UNIX最初是用手编语言在GE653机上实现的。

1969年Ken Thompson，Dennis M. Ritchie开始研究UNIX，历时一年获得成功，他俩在PDP-11上用汇编语言实现了第2版。

1970年，Ken Thompson为了提高UNIX的可读性和可移植性，对BCPL(Basic Combined Programming Language)语言——计算机软件人员在开发系统软件时作为记述语言使用的一种结构化程序设计语言，能直接处理与机器本身数据类型相近的数据，具有与内存地址对应的指针处理功能——作了进一步简化，开发出了更接近硬件的B语言，并用B语言写了另一个UNIX操作系统。

1971～1972年，Dennis M. Ritchie用了一年左右时间在B语言的基础上开发了C语言，从1972年起C语言开始投入使用。

1973年，Ken Thompson和Dennis M. Ritchie合作，用C语言重写了UNIX操作系统，这是UNIX的第5个版本(V5)。这个版本给UNIX的发展开辟了新的局面，而UNIX的发展又回过头来给C语言的普及创造了条件。

1978年，Thompson和Ritchie二人合作撰写了C Programming Language一书，确定了C语言中的27个保留字，这就是过去通用的老标准——K&R标准。

1983年，Thompson和Ritchie二人因发明C语言的功绩荣获图灵奖。同年，美国国家标准研究所公布了ANSI C，使得各种C语言编译程序有了一个趋同的标准。

1987～1988年，ANSI C增加5个保留字，使C语言的保留字达到32个，成为新标准——ANSI标准。本书所讲内容是完全按照新标准安排的。

C语言一直不停地发展着。1979～1983年，贝尔实验室的Bjarne Stroustrup在剑桥大学做博士论文时曾经用过Simula语言，Simula中类(Class)的概念使它能直接地将应用程序中的概念映像到语言构造之中。他抓住类的概念对C作了扩充，提出了带类的C。

1983～1984年，Bjarne Stroustrup花9个月时间写出《C++程序设计语言》。1987～1989年，在他的一篇论文中披露了C++ 2.0版。随着时间的推移，C++成为编制大型软件的主要语言，在Windows和OS/2走红后，在第三代BASIC出现前，C++成了在这两个平台上编程的主要工具。C++包含了C的全部内容，要学习C++一般总是从掌握C开始。但是，也不尽然。美国的Robert J. Traister教授一直主张可以从BASIC一步跳到C++。但本书只讲到C，至于C++扩展出去的内容只能留待另一本书去谈。

1.2 C语言的特点

C语言从它诞生之日起就广泛地被专业程序员选用、被业余程序员喜爱，自然是有它内在原因的。

C给人一个崭新的印象，它看起来像高级语言，用起来像汇编语言，它具有下列特色：

1.2.1 高效率

C的效率很高，它的设计充分发挥了当代计算机各方面的效能。它的源代码的编译速度快，编译后所得代码的质量也高，几乎能与汇编比翼，能达到最快的速度，同时又能最有效地使用内存。

1.2.2 良好的可移植性

C是一种可移植语言，它不从属于某种特定的硬件，然而它又具有许多汇编语言的特点。用C写的程序，几乎可以在从微机到超级巨型机的各种级别的机器上运行。其他语言就没有这样方便，例如同属第二代BASIC的TB(True BASIC)和QB，由于它们有不同的保留字、不同的内部函数，移植一个程序，差不多是重写一遍，不像C的移植这样方便。

1.2.3 简洁，紧凑，自由度大

将QB与C做个比较，QB有保留字221个，C只有32个，仅为QB的六十九分之一(1/69)，从这一比较可以看出C是何等的简洁。

C虽然保留字少但运算符特别多(多达34个)，它甚至把括符、赋值号和强制类型转换符等都作为运算符使用，因而C语言可以在一条不太长的语句中执行多种运算、完成多项功能，非常紧凑。而BASIC语言只能靠增加保留字扩展语言的功能，BASIC语言每条语句的功能是确定的，不能像C那样可以任意扩展和附加，因此它就无法写出像C那样紧凑的清单。

由于C的运算符特别丰富，用这些运算符可以构成千变万化的表达式。C语句中有一种语句叫表达式语句。这一弄，C的语句变化就大了。C的规则少、变化大，这样就给程序员留下了更大的自由度。从变化的多样性看，我们可以把BASIC比喻成象棋——它有很多规则，每个棋子的运行都受到特殊规则的制约；而把C比喻成围棋——它只有很少几条规则，每个棋子几乎可以投到棋盘中任何一个目上，它比象棋有更大的自由度，其变化有更丰富的多样性。编程的自由度给富有才华的程序员提供了尽情驰骋的广阔原野和任意翱翔的无垠晴空。难怪整整一代程序员如此酷爱C，正是因为C给他们中的佼佼者提供了大展身手的天地。

1.2.4 C允许低级操作和高级结构

C语言将程序员放在紧贴计算机CPU的位置上，程序员使用C可以作许多只有用汇编语言才能实现的低级操作(如直接访问物理地址、进行位操作等)；同时又可以建立高水平的控制结构(如循环结构和分支结构等)。软件界中流传有“八二律”一说，意思是说计算机时的80%往往是消耗在程序中20%的代码上的。有了C语言，程序员可以将程序中80%的代码用结构非常清晰的高级语言写出，然后根据需要仔细优化那余下的、耗费机时最多的20%代码，必要

时甚至可用汇编语言写出这部分代码。

这里不妨看看坦南鲍姆(A. S. Tanenbaum)的 MINIX 操作系统。在坦南鲍姆著作的中译本中 MINIX 源程序从 325 页到 460 页,即共占 136 页。其中 336 页到 344 页即不足 9 页系用汇编写成,其余 127 页均用 C 写成。由此可见,MINIX 作为一个教学用操作系统源程序其中只有 7%的代码用汇编写成,其余 93%的代码都是用 C 写的。操作系统尚且如此,其他应用程序就更不用说了。

1.2.5 C 的缺点

如果只注意到以上几点,把 C 看成是一种完美无缺的语言,那是不符合实际的。C 有 C 的缺点:

- C 是一种非强类型语言,数据类型及其值不是严格地一一对应的,数据类型转换也比较随便。
- C 的编译器有时允许表达式和参数表重新排列求值顺序,不同的编译器有时会产生不同的结果。
- 不具备数据边界的自动检查功能。例如给数组赋值,超越边界系统不会提出警告,结果把数据搞乱!
- 运算符较多,而且有的运算符又是一符多用(如 * 既是乘法运算符又是取内容运算符,& 即是按位与运算符又是取地址运算符),容易引起混淆。优先级多达 15 级,而且分为左结合与右结合,使初学者感到难记和麻烦。
- 指针的使用,提高了编程的灵活性,但因它能直通系统内核,初学者由于错误地使用指针而造成系统崩溃的事累有发生。

1.2.6 C 语言是面向程序员的语言

C 语言由于追求充分发挥前面所说的那些优势,因而难以避免上面提到的几个缺点。所以,C 语言不是理想的初学者语言,它是面向程序员的语言。

在计算机教育中深刻认识 C 的特点是有重大意义的。美国 Robert J. Traister 和 Greg Perry 等专家都主张以 BASIC 作为初学者的首选语种,在掌握了 BASIC 之后再学习 C 和 C++,将收到事半功倍的效果。

1.3 C 语言程序概貌

1.3.1 C 程序与 QB 程序的比较

在学习 C 语言的各种细节之前,让我们先对 C 程序有个轮廓地了解。

这里先写出一个最简单的已知直径(Diameter)求圆周长(Circle)的 QB 程序:

```
REM FileName: BTOC 1_1.BAS
DEFSNG A_Z                  '将全部变量定义为单精度型
CONST PI=3.141593           '将 PI 定义为常量其值为 3.141593
INPUT "Diameter="; Diam     '由键盘输入直径值存于 Diam 中
Circ=Diam * PI              '算出圆周值,存于 Circ 中
```

```
PRINT "Circle="; Circ        '将圆周值输出
END                          '程序结束
```

运行此程序。当屏幕提出 Diameter=? 时，若直径为 2.1，则键入2.1↙，屏幕上显示 Circle=6.597345；若直径为 3.7，则键入3.7↙，屏幕上显示 Circle=11.62389

下面写出与此程序等价的 C 语言程序：

```
/* FileName: BTOC 1_1.C      */
#include <stdio.h>                  /* 调用 include 文件 */
main ()                             /* 以下为主函数 */
{
   const float PI=3.141593;         /* 定义 PI 并给它赋值 */
   float Diam, Circ;                /* 说明 Diam 和 Cire 的类型 */
   printf ("Diameter=?");           /* 显示 Diameter = ? */
   scanf("%f", &Diam);              /* 将键盘输入值存入 Diam */
   Circ=Diam * PI;                  /* 算出 Circ 值 */
   printf ("Circle =%f", Circ);     /* 输出圆周值 */
}
```

本书读者对 QB 程序一看就很清楚，这里我们只对 C 程序作些必要的说明，以期读者能尽快在头脑中形成一个关于 C 程序的大致轮廓，随着以后章节的学习，把这个轮廓逐渐具体化、逐渐充实起来。

● C 的文件名与 QB 文件名的要求基本相同，C 的源文件以.C 为后缀，目标文件以.OBJ 为后缀，执行文件以.EXE 为后缀。

● C 的源文件中以/*……*/作为注释语句的起止符号。与 QB 相同，注释语句的作用在于改善源文件的可读性，它不参加任何运算，在编译过程中系统会自动将注释语句删除。

● 应该说包含文件是 C 语言源程序中的一个不可缺少的组成部分，通常在程序头部由#include〈……〉引出。QB 中的包含文件的概念，正是出自于 C 语言。不同处有两点：① QB 的包含文件先要由程序员自己建立，然后才能引用；而 C 中除了有这种“自建自用”的包含文件外，还有一批系统中早已建立好拿过来就用的包含文件，如此例中的 stdio.h。② QB 中用元命令 $INCLUDE:'……' 引用包含文件，包含文件均以.INC 为后缀；C 中的包含文件可以有不同的后缀，如上例中是.h 表示这里引用的是一个“头文件”。

● QB 的保留字一律用大写，即使你用小写字母键入，灵巧编辑器经过检查确认无语法错误后，会自动将它转化为大写字母。系统对变量名中字母的大小写不加区别。C 却一律采用小写（当然，变量名和输出字串中可以有大写字母），任何时候编辑器不会自动改变程序员输入的字母。C 对名字比较敏感，它对名字字母的大小写是加以区别的，即 Diam 与 DiAm 与 diam 是不同的。

● 每一个 QB 程序至少都有一个主模块；每一个 C 程序至少都有一个主函数，C 程序一般都从主函数 main()启动。main 后的括号，表示这是一个函数。C 程序中可以有任意多个函数，函数应该有自己的名字。主函数是打开程序后见到的第一个函数，它的名字是 main，不能作任何变动。

● 前后花括号{……}标志着函数体的开始和结束。在构成复合语句时、在往数组中赋初值时也要用到花括号。