

高等学校电子信息类教材

C 语言程序设计

孙淑霞 何建军 肖阳春 编著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书作为 C 语言程序设计课程的教科书, 共由 10 章组成。主要内容包括: C 语言简单程序的编写和调试, C 语言程序设计基础(其中包括: 基本数据类型和基本输入、输出函数、运算符与表达式), 控制结构, 数组, 指针, 函数, 编译预处理与变量的存储类型, 文件, 结构体与共用体, 图形程序设计基础, 每章后面都附有编程练习题。全书内容安排紧凑, 简明扼要, 由浅入深, 实用性强。该书可与《C 语言程序设计实验与习题指导》配套使用。

本书可作为大专院校非计算机专业本科生、研究生的相关课程的教学用书, 也可作为计算机专业学生学习 C 语言程序设计的教材, 同时还可供自学者参考。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有, 侵权必究。

图书在版编目(CIP)数据

C 语言程序设计/孙淑霞等编著. —北京: 电子工业出版社, 2003.1

高等学校电子信息类教材

ISBN 7-5053-8196-2

. C... . 孙... . C 语言—程序设计—高等学校—教材 . TP312

中国版本图书馆 CIP 数据核字(2002)第 096678 号

责任编辑: 王昌铭

印 刷: 北京金特印刷厂

出版发行: 电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 17.5 字数: 448 千字

版 次: 2003 年 1 月第 1 版 2003 年 1 月第 1 次印刷

印 数: 8000 册 定价: 23.00 元

凡购买电子工业出版社的图书, 如有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系。

联系电话: (010) 68279077

出版说明

信息社会对每一个人，尤其是科技人员、教育工作者和广大学生提出了知识的更新和重构的问题，在知识增值的情况下，对如何获取知识和信息产生了强烈的要求。虽然我们可以从网络、从媒体、从书本上获得信息和知识，但在教育领域，教材和技术参考书仍然是重要的知识来源，它们是加工（处理）后的信息。

计算机科学技术的发展，引起了一场革命，形成了一种文化，创新了一种观念，促进了社会的发展。以微电子为依托，计算机与通信的联姻，带来了全球经济文化的大变革，信息全球化已经初见端倪。为了适应知识的更新和信息的获取，经过高等院校、企业事业单位和应用领域用户的认真讨论，我们规划了这套电子信息类教材和技术参考书。

在现代社会中，电子信息类教材和技术参考书包含了计算机专业类和非计算机专业类、网络与通信专业类、电子技术类，读者层次也覆盖了各个应用领域，这些教材和技术书籍将对我们的信息技术的普及起到极大的推动和促进作用。

在现代教育体系中，教材的作用是不言而喻的，但教材的内容则各有千秋。由于学科发展、专业结构、培养对象的不同，全国统编教材已经不完全适应各个学校。多种风格各异、独具特色的教材如雨后春笋般层出不穷，并深受读者欢迎。一本教材的内容不可能包揽一切，也不可能适应各层次的读者对象。但是，教材一定要有自己的特色，这些特色是通过知识点及其之间的关联性体现出来的，这样，教材的选用者就可以了解该教材的适用范围和对象，就能够很好地规划自己的教学。

信息技术类参考书是对专业教材内容的一个重要补充，也是对专业技术实践环节（实验、实训、实作）的一个实施，这种参考书体系将紧跟国际新技术发展，把最实用的技术及时介绍给读者。

本系列教材及技术参考书采用编委会形式，选题内容由需求引导，编委会讨论，专家（行家）撰写，编委会审阅，最后定稿，以保证书的编写质量。

当我们跨进 21 世纪大门的时候，希望广大的读者能与我们携手并进，共同关心我们的教材和技术参考书的出版，将需求、建议和意见及时反馈给我们，与我们一起开辟一片新的天地。

编委会

前 言

C 语言是应用很广泛的一种语言，它的结构简单、数据类型丰富、表达能力强、使用灵活方便。C 语言既有高级语言的优点，又具有低级语言的许多特点。用 C 语言编写的程序，具有速度快、效率高、代码紧凑、可移植性好的优点。利用 C 语言，可编制各种系统软件(例如著名的 UNIX 操作系统就是用 C 语言编写的)和应用软件。

本教材由 10 章组成。第 1 章讲解 C 语言简单程序的编写和调试，第 2 章是 C 语言程序设计基础，第 3 章是控制结构，第 4 章是数组，第 5 章是指针，第 6 章是函数，第 7 章是编译预处理与变量的存储类型，第 8 章是文件，第 9 章是结构体与共用体，第 10 章是图形程序设计基础。

本教材在编写中努力做到概念清楚、实用性强、通俗易懂。在编写中引入了大量的实例来说明相关的内容，力求让读者尽快上手，尽快走进 C 语言的殿堂。本书在编写上有以下的特点：

1. 教材在内容的组织上考虑了 C 语言的特点，在讲解了数组后，紧接着就进行了指针的讲解，使读者很容易将数组与指针联系起来，更好地理解指针。

2. 将文件的使用提前讲解，使读者尽早接触文件，掌握文件的基本操作，给大批量数据的处理带来方便。同时可以较好地解决学生在学习 C 语言时不能熟练地掌握文件的使用，使文件部分成为学生学习过程中的一大难点的问题。

3. 本书通过一个关于学生成绩处理的实例，从简单变量到结构体，始终用该实例贯穿各章。使读者通过一个实例循序渐进地、有比较地进行学习。

4. 本书提供了习题中的全部参考答案。所有程序均在 Turbo C 环境下调试通过。由于篇幅有限，书中的程序只给出了一种参考程序，读者在学习过程中可以举一反三。

与本书一起出版的《C 语言程序设计实验与习题指导》是本书的配套教材，在学习的过程中可通过完成该教材中相应的习题和上机实践加深对所学知识的理解，达到真正掌握 C 语言程序设计的目的。

本书第 1, 4, 5, 6, 8, 9 章由孙淑霞编写，第 2, 3 章由何建军编写，第 7 章由肖阳春编写，附录 A 由魏琴提供，孙淑霞统编了全书。丁照宇、魏琴、鲁红英、袁爱新也参加了本书的部分编写工作。由于作者水平有限，书中难免有错误之处，请读者批评指正。

最后要感谢高等学校电子信息类教材编委会在本书的出版过程中给予的指导和帮助，特别要感谢王昌铭老师在该书的出版过程中所做的大量工作，同时还要感谢电子工业出版社给予的大力支持。

编著者

2002 年 10 月

第 1 章 C 语言简单程序的编写和调试

C 语言是国际上应用最广泛的几种计算机语言之一，它不仅可以用于编写系统软件，如操作系统、编译系统等，还可以用于编写应用软件。大家所熟悉的 UNIX 操作系统就是用 C 语言编写的。

本章简要介绍 C 语言的特点，C 程序的基本结构和 C 程序的调试。

1.1 C 语言的特点

C 语言之所以能够广为流传，是因为它有很多不同于其他程序设计语言的特点。其主要特点有：

数据类型丰富。C 语言除了整型、实型、字符型等基本数据类型外，还具有数组、指针、结构、联合等高级数据类型，能够用于描述各种复杂的数据结构（如链表、栈、队列等）。指针数据类型的使用，使 C 程序结构更为简化、程序编写更为灵活、程序运行更为高效。

运算符种类丰富。C 语言具有数十种运算符，除了具有一般高级语言具有的运算功能外，还可以实现以二进制位为单位的位运算，直接控制计算机的硬件，还具有自增、自减和各种复合赋值运算符等。C 程序编译后生成的目标代码长度短、运行速度快、效率高。

符合结构化程序设计的要求。C 语言提供的控制结构语句（如 if-else 语句、while 语句、do-while 语句、switch 语句、for 语句）使程序结构清晰，其函数结构使程序模块具有相对独立的功能，便于调试和维护，有利于大型软件的协作开发。

可移植性好。用 C 语言编写的程序几乎不作修改就可用于各种计算机和各种操作系统。

C 语言的这些特点使 C 语言很快应用到了各计算机应用领域中的软件编写，如数据库管理、CAD、科学计算、图形图像处理、实时控制、信息处理等软件。

然而，C 语言也不是十全十美的，它也有缺点。它的语法限制不太严格，例如，缺乏数据类型的一致性检测和不进行数组下标越界检查。正因为 C 语言允许编程者有较大的自由度，使 C 程序容易通过编译，但却难以查出运行中的错误。初学者一定不要以为编译通过了，程序就一定是正确的，就应该运行出正确结果。要想尽快找到程序中的错误，一定要掌握调试程序的方法和技术，多上机实践。

1.2 C 程序的基本结构

首先让我们来看两个例子。

【例 1.1】 在屏幕上输出一串字符。

```

main()                               /* main 表示主函数 */
{                                     /* 函数体开始 */
    printf("Let us studing C program together. \n"); /* 屏幕上输出一字符串 */
}                                     /* 函数体结束 */

```

执行该程序将在屏幕上显示如下信息：

Let us studing C program together.

该程序很简单，由一个主函数 main 组成。函数体内（由大括号 { } 括起来的部分）只有一条语句，该语句是调用 C 语言提供的库函数 printf（格式输出函数）输出双引号中的字符串，其中的“\n”是转义字符（参见 2.4.1：格式输出函数），表示换行，即当在屏幕上输出到这里时将光标移到下一行的起始位置，以后再有输出就从该位置开始显示。

程序中每一行后面都是注释。注释可以是由“/*”和“*/”括起来的任何文字，它可以出现在程序的任何地方，用来说明程序段的功能、语句行的作用、变量的作用等内容，主要用于向程序阅读者进行说明和交流，使其能读懂程序，也便于程序的调试。每一个程序编写者都应该养成在必要的位置上加写注释的良好习惯。

C 语言的每一条语句都是以分号结束的。

【例1.2】 求半径为 r 的圆面积。

```

main()                               /* 求半径为 r 的圆面积 */
{                                     /* 函数体开始 */
    float area;                       /* 定义实型变量 */
    int r;                             /* 定义整型变量 */
    printf("请输入圆的半径：");      /* 屏幕上显示“请输入圆的半径：” */
    scanf("%d",&r);                 /* 输入圆的半径 */
    area=3.1416*r*r;                 /* 计算圆面积 */
    printf("\n area=%f\n",area);     /* 输出圆面积 */
}                                     /* 函数体结束 */

```

C 语言的函数体主要由两部分组成：声明部分和可执行部分。声明部分用于定义和说明变量、数组等；可执行部分由可执行语句和函数调用等语句行组成。

该程序中的每一行都加有注释，说明了每一语句行的作用。其中函数体中的第 1 行和第 2 行是对函数体中要使用的变量进行定义，即声明部分；第 3 行是在屏幕上显示“请输入圆的半径：”，这种方法常用于输入前对用户的提示；第 4 行调用输入函数（参见 2.4.2：格式输入函数 scanf()）输入圆的半径；第 5 行是计算半径为 r 的圆面积，并将计算结果赋给变量 area；第 6 行是输出变量 area 的值，即圆的面积。如果输入的半径 r=15，则屏幕上显示的是：

area=706.859985

由此可见，上面两个 C 程序的执行都是从 main() 函数开始，依次执行函数体中各条语句，直到结束。

通过上述例子可以得到如下 C 程序的组成特点。

📖 C 程序的组成特点：

一个 C 源程序由函数构成，其中有且仅有一个主函数 main()。

C 程序总是由 main() 函数开始执行，且结束于主函数。

分号 “ ; ” 是 C 语句的一部分，每一条语句均以分号结束。

C 程序书写格式自由，一行内可写多条语句。

程序的注释部分应括在 /*...*/ 之间，/ 和 * 之间不允许留有空格；注释部分允许出现在程序的任何位置上。

在上述例子中还出现了 C 语言程序设计中的标识符、关键字、运算符、常量和变量、函数调用等基本概念。

1. 标识符

例 1.2 中的 area 和 r，在 C 语言中称为标识符。

C 语言的标识符用于给程序中的常量、变量、函数、文件指针和数据类型等命名。用户可以根据需要进行命名，形成用户标识符。标识符的构成规则如下：

标识符由英文字母、数字、下划线组成；且第一个字符不能是数字，必须是字母或下划线。

标识符中的大、小写英文字母的含意不同。

不同的 C 编译系统对标识符所用的字符个数有不同的规定，ANSI C 可以识别标识符的前 31 个字符，但有的 C 编译系统只识别前 8 个字符。

用户取名时，应当尽量遵循“见名知意”和“简洁明了”的原则。

2. 关键字

例 1.2 中的 float 和 int，在 C 语言中称为关键字。

关键字又称保留字。它是 C 语言中已预先定义、具有特定含义的标识符。C 语言共有 32 个关键字，如表 1.1 所示。所有关键字都用小写英文字母表示，且这些关键字不允许用做用户标识符。

表 1.1 关键字

auto	break	case	char
const	continue	default	do
double	else	enum	extern
float	for	goto	if
int	long	register	return
short	signed	sizeof	static
struct	switch	typedef	union
unsigned	void	volatile	while

3. 运算符

例 1.2 中的=, *, &是 C 语言中的运算符。

运算符是用来表示某种运算的符号，其中有的是由一个字符组成，例如 +, -, *, / 等；有的是由多个字符组成，例如<=, <<, &&, != 等。C 语言的运算符很丰富，主要有以下几类：

算术运算符	(+ - * / %)
关系运算符	(> < == >= <= !=)
逻辑运算符	(! &&)
位运算符	(<< >> ~ ^ &)
赋值运算符	(= 及其扩展赋值运算符)
条件运算符	(? :)
逗号运算符	(,)
指针运算符	(* 和 &)
求字节数运算符	(sizeof)
强制类型转换运算符	((类型))
⑪ 分量运算符	(. ->)
⑫ 下标运算符	[])

有些运算符具有双重含义，例如“%”可以作为求余运算符，但当它出现在输入、输出函数中时，就是“格式控制符”了。

4. 常量和变量

例 1.2 程序中的 3.1416 在程序运行过程中是不会发生变化的，C 语言将其称为常量。程序中的 area 和 r 称为变量，它们的值在程序运行过程中是可以改变的。

C 语言中的常量分为整型常量、实型常量和字符型常量。如 5, 10, -34 是整型常量，2.3, 9.8, -15.0 是实型常量，'a', 'b' 为字符常量。

变量是指在程序运行过程中其值可以改变的量。每一个变量都有一个名字，根据变量的不同类型，系统将为每一个变量分配相应的内存单元。例如，系统为整型 (int 型) 变量分配 2 个字节的内存单元，为实型 (float 型) 变量分配 4 个字节的内存单元，为字符型 (char 型) 的变量分配 1 个字节的内存单元。内存单元中存放的是变量的值。程序执行过程中对数据的读、写是通过变量名找到相应的内存单元来实现的。

📖 C 语言规定：

程序中所用到的所有变量，都必须“先定义（说明变量的名字和数据类型），后使用”。任何一个未经定义就使用的变量都会被 C 语言的编译程序认为是非法变量，引起编译出错：

```
Undefined symbol 'xxxxxx'
```

5. 函数调用

例 1.2 程序中的 printf()和 scanf()分别为输出函数和输入函数的调用。它们是 C 语言的库

函数中为用户提供的输出、输入函数。C 语言中的函数分为：系统提供的库函数和用户自定义函数。对于库函数，用户可以直接调用；用户自定义函数是用户用以解决专门问题所定义的函数（参见第 6 章）。

1.3 C 程序的调试

目前微机上主要用的是 Turbo C，它是一个 C 语言的集成开发环境，用于 C 语言程序的建立、编辑、编译、连接、运行和程序的动态调试。本节主要介绍如何利用 Turbo C 集成环境进行 C 程序的调试。

初学编程，最好是先把 C 程序编好后，再到计算机上进行调试。当编程能力达到了一定的熟练程度后，就可直接在计算机上写程序了。

1.3.1 编辑

无论是新编写一个程序，还是修改一个原有的程序，首先要启动 Turbo C，进入 Turbo C 的集成环境。编辑就是把程序输入到计算机中，并以文本文件的形式保存到磁盘上。用 C 语言编写的程序文件叫源程序文件，其文件的扩展名为“.C”。

1. Turbo C 的启动

在计算机中正确地安装了 Turbo C 编译程序，并设置好了相应的路径，就可启动 Turbo C 了。这时，只需在系统提示符下键入 TC 并回车，即可进入 Turbo C，屏幕上将出现 Turbo C 的集成界面，如图 1.1 所示。

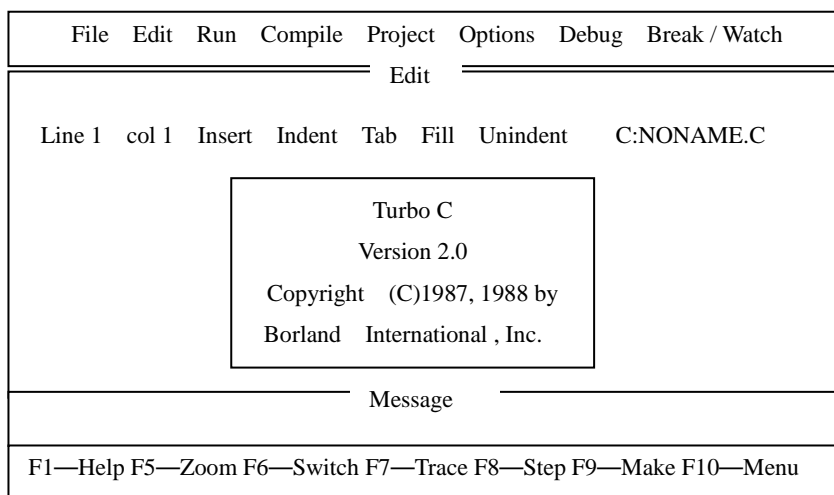


图 1.1 Turbo C 的集成界面

图 1.1 由 3 部分组成：顶部的主菜单区，中间的编辑区，底部的功能键和热键说明区。当键入任意键时，编辑区中间显示 Turbo C 版本号的广告窗口就会消失。编辑区最上面一行是编辑状态提示。其中：

Line n 行显示, n 是行数, 随光标位置变化而变化。

Col n 列显示, n 是列数, 随光标位置变化而变化。

Insert 插入模式开关, 用 Ins 键或 Ctrl-V 键进行模式开关的切换。Insert 表示当前是插入状态, 这时所键入的字符将出现在光标之前; Insert 消失表示当前是改写状态, 这时键入的字符将替换当前光标所在处的字符。

Indent 自动缩进开关, 用 Ctrl-OI 键切换。

Tab 制表开关, 用 Ctrl-OT 键切换。

Fill 它与 Indent 和 Tab 的开关 (ON/OFF) 一起使用。当 Tab 模式为 ON 时, 编辑系统将用制表及空格符优化每一行的开始。

2. 文件的编辑和保存

当启动 Turbo C 时, 主菜单总是指向 File 菜单项, 按 <Esc> 键就可进入编辑窗口。

(1) 新建文件

新建文件时, 如果在系统提示符后只键入 TC, 系统就以默认文件名 NONAME.C 作为新文件的名字, 进入 Turbo C。程序输入完后, 按 <F2> 键保存文件, 屏幕上将出现图 1.2 所示的提示信息。这时用户可以修改该文件名, 也可以修改保存该文件的路径。

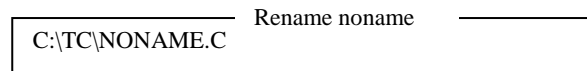


图 1.2 保存文件的提示信息

(2) 编辑一个已存在的文件

当要编辑或调试一个磁盘上已经存在的文件时, 可以在系统提示符后只键入 TC 和要编辑的文件名, 回车后即可打开该文件。例如:

```
C:\> tc 1-1.c
```

这时, 要编辑的文件 1-1.c 就出现在 Turbo C 的编辑窗口中了。

如果只键入 TC 进入 Turbo C 后, 可按 <F3> 键装入文件, 屏幕上出现如图 1.3 所示的画面。用户输入文件名, 就可装入该文件。

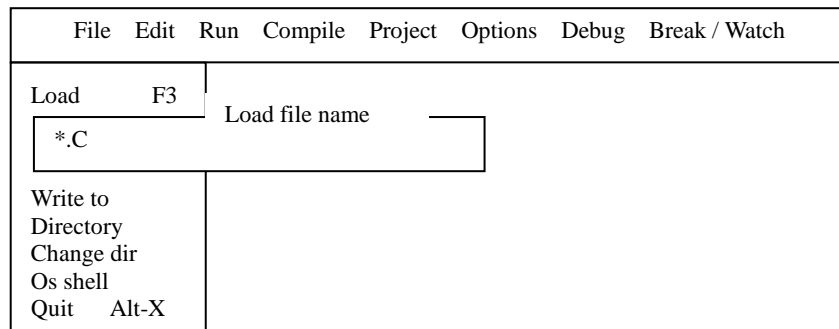


图 1.3 装入文件时的画面

当编辑文件时, 编辑窗口的文件名前将出现一个星号 “*”, 如图 1.4 所示。它表示当前文件已被修改尚未存盘, 提示用户该文件还没有保存。星号后面是文件名, 其中 C: 是当前驱

动器名，1-1.C 是当前编辑的文件名。

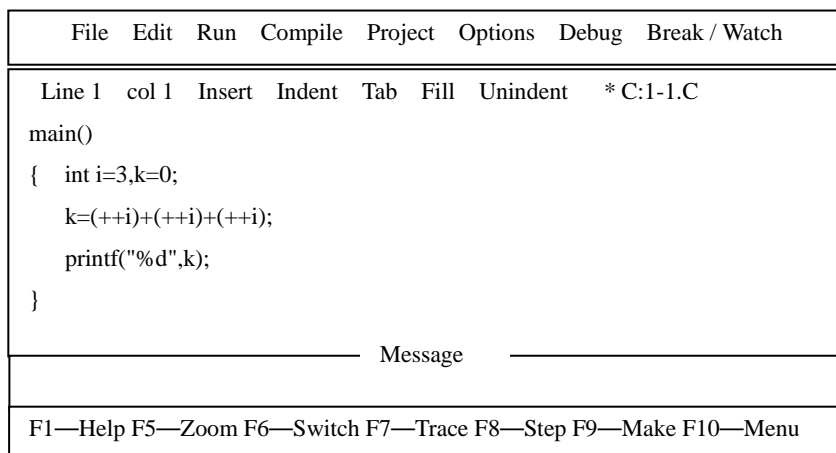


图 1.4 编辑文件时的窗口画面

程序编辑完后，按<F2>键即可保存该文件。如果要以另一个名字保存文件，可用 File 菜单中的 Write to 命令实现。

3. Turbo C 的常用编辑功能键

使用 Turbo C 的编辑功能键可提高编辑的效率。在《C 语言程序设计实验与习题指导》的附录 B 中列出了 Turbo C 的一些常用编辑功能键，其中包括基本移动键、快速移动键、插入与删除键和块操作键。

1.3.2 编译

编译是指用 C 语言提供的“编译程序”将编辑好的源程序翻译成二进制形式的“目标程序”的过程。在 MS-DOS 中，目标程序的扩展名为“.obj”。

在编译过程中，编译器将检查源程序每一条语句的语法错误。当发现错误时，就在编辑窗口下面的信息（Message）窗口中显示错误的性质、位置和错误的类型，如图 1.5 所示。

编译源程序时，选择“Compile”菜单中的“Compile to OBJ”命令。如果编译成功，就可得到一个扩展名为“.obj”的目标程序。如果在编译过程中发现了错误，在屏幕下方的信息窗口中就会显示出错误的类型、错误发生的位置和错误的原因。即：

<错误性质> <错误发生的位置>：<错误内容>

在图 1.5 的信息窗口中指出：程序 1-1.C 的错误性质是 Error，这是一个必须修改的严重错误，否则程序无法运行；该程序错误发生的位置在第 4 行；错误的内容是 main 函数中的符号 k 未定义。

编译过程中还可能出现另外一种性质的错误，即 Warning。这表示编译程序不能百分之百地确定的错误，即源程序在这里可能有错，也可能没有错。如果程序中只有 Warning 错，一般是可以继续连接和运行程序的。

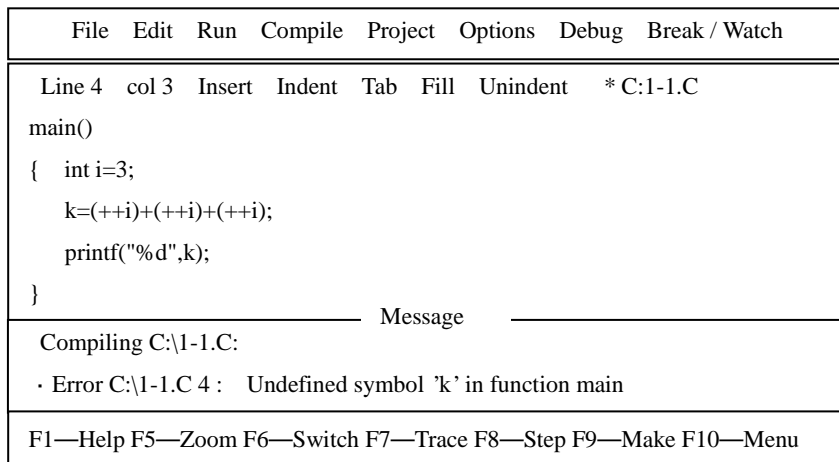


图 1.5 信息窗口中显示的出错信息

在编译过程中可以用<F6>键在程序编辑窗口和信息窗口中进行切换，以便修改程序或者查看错误提示信息。

值得注意的是，有时编译程序会给出很多错误信息，但并不表示程序中真的有那么多的错误，有可能是一个错误所致。例如，程序中有一个用到的变量没有定义，所有使用到该变量的行在编译时都会出错。所以在修改编译错误时，对不明显的错误，最好是修改一个错误就重新编译一次。

《C 语言程序设计实验与习题指导》的附录 C 中列出了常见的编译错误。

1.3.3 连接

编译所产生的“目标程序”是不能运行的，需要进行连接生成可执行文件，即扩展名为“.exe”的文件，才能运行。

连接就是把目标程序与系统的函数库和与该目标程序有关的其他目标程序连接起来，形成一个可执行程序。

连接时，可选择“Compile”菜单中的“Link EXE file”命令。执行该命令将目标程序和函数库连接，连接成功后，就可得到一个扩展名为“.exe”的可执行程序。

选择“Compile”菜单中的“Make EXE file”命令或按<F9>键，可以一次性完成编译和连接。执行该命令首先检查源程序是否已经编译过，如果尚未编译，或编译后又进行了修改，则先编译，再连接。

选择“Compile”菜单中的“Build all”命令，则无论源程序是否编译过，系统都将进行编译和连接。所以，该命令相当于连续执行编译和连接两项功能。

常见的连接错误是外部调用有错，系统将指出外部调用中出错的模块名或找不到的库函数。这时，需要检查程序中是否有错写函数名或缺少文件包含命令的情况。

连接错误是由连接程序检查的。在找到连接错误的原因并修改后，必须进行重新编译才能再次连接。

《C 语言程序设计实验与习题指导》的附录 C 中列出了常见的连接错误。

1.3.4 运行

在操作系统下,可以运行已经生成好的可执行程序。在 Turbo C 的集成环境中,选择“ Run ”菜单中的“ Run ”命令或按<Ctrl-F9>键,就可执行该程序了。如果出现运行错误或运行结果不对,则需要重新修改程序。修改后的程序,需要重新进行编译、连接和运行,或按<Ctrl-F9>键一次完成编译、连接和运行。

运行错误是比较难以查找和判断的。因为运行错误几乎没有提示信息,只能依靠编程人员的经验来进行判断。

运行中的错误通常有两种:一是系统给出错误信息,用户可以根据错误信息进行分析,找出错误;另一种是运行结果不正确或运行异常结束。如果运行结果不正确,就需要仔细阅读程序并分析造成错误的原因;如果运行异常结束(即死机),则可能是程序中的循环结构有错或系统程序被破坏。

1.3.5 程序的跟踪调试

一般来说,编译和连接中的错误比较容易查找和修改,要查找运行中的错误就相对困难一些。为了提高查找错误的效率,一方面要提高阅读程序的能力,另一方面要掌握跟踪调试程序的方法。

跟踪调试是指程序在运行过程中的调试。它的基本原理是通过一句一句地执行程序,分析和观察程序执行过程中数据和程序执行流程的变化,从而查找出错误的原因和位置。跟踪调试有两种方法:一种是传统的方法,即在程序中直接设置断点(使用 `getch()`函数)、输出重要变量的内容等来分析和掌握程序的运行情况;另一种方法是利用 Turbo C 集成环境中的分步执行、断点设置和显示变量内容等功能对程序进行跟踪。

在 Turbo C 集成环境中的跟踪调试主要是通过使用 Run、Debug 和 Break/Watch 3 个菜单中的命令来实现的。菜单 Run 中的命令负责跟踪程序代码。菜单 Debug 中的命令可以计算和显示表达式的值,还可以为变量临时赋一个新值,以便调试进行下去。菜单 Break/Watch 中的命令可以设置程序断点、增加监视表达式,以便随时显示某个变量或表达式的值。

跟踪调试常用的命令有:

(1) Run/Go to cursor (执行到当前光标位置) 命令或按<F4>键

此命令是使程序执行到光标所在处暂停,程序处于跟踪调试状态,且有亮条显示在暂停处。

(2) Run/Trace into (跟踪) 命令或按<F7>键

此命令是单步执行程序,即每执行一次该命令,程序仅执行一步就暂停,程序处于跟踪状态。

(3) Run/Step over (步进) 命令或按<F8>键

该命令的功能与跟踪命令类似,只是在对函数调用的处理方法上有所不同。该命令不转入被调用的函数,直接执行到源程序中的下一行。

(4) Break/watch/Toggle breakpoint (设置断点) 命令或按<Ctrl-F8>键

该命令的作用是使程序可以分段运行。如果在程序中的某个语句处设置了断点,在使用

Run 菜单中的命令时，就会在断点处停下来。这时可以使用其他调试手段进行程序的调试。

(5) Break/watch/Add watch (设置观察对象) 命令或按<Ctrl-F7>键

执行该命令可以设置要观察的变量或表达式，这些变量或表达式的值在程序的调试过程中将显示在屏幕下方的信息窗口中。

(6) Debug/Evaluate (变量查看及修改) 命令或按<Ctrl-F4>键

当程序运行到断点处时，可利用该命令查看变量或其他数据项的内容。执行该命令时，屏幕上弹出一个窗口，窗口分为3栏：最上面是设置(Evaluate)栏，用于输入要查看的变量名或表达式；中间是结果(Result)栏，用于显示要查看的变量或表达式的值；最下方是修改(New value)栏，用于修改变量的值。

📖 C 程序的上机步骤：

启动 Turbo C	输入与编辑源程序	编译源程序	与库函数及其他目标程序连接	运行
	(生成源程序.c)	(生成目标程序.obj)	(生成可执行程序.exe)	

如果编译或连接过程中有错，则需要修改源程序，再重新进行编译、连接和运行，直到得到正确结果为止。

在程序的调试过程中，注意使用动态跟踪调试技术，以便提高调试程序的效率。

1.4 程序举例

【例1.3】 求长为3、宽为5的矩形面积。

```
/* - - - - - 程序 1 - - - - - */
main()
{ float l=3,w=5,area;
  area=l*w;
  printf("\narea=%f\n",area);
}
/* - - - - - 程序 2 - - - - - */
main()
{ float l,w,area;
  printf("Please input rectangle's length and width:");
  scanf("%f%f",&l,&w);
  area=l*w;
  printf("\narea=%f\n",area);
}
```

分析：只要知道了矩形的长和宽，就可以求出该矩形的面积。程序1和程序2都可以按要求计算出矩形的面积，两个程序有什么不同呢？

程序1中直接给出了长和宽的值，可以计算出该矩形的面积。但是，如果要计算长为6、宽为8的矩形面积，就需要修改程序，把程序中的3改为6、5改为8，然后再重新进行编译、

连接和运行。程序 2 是通过键盘输入矩形的长和宽，当输入不同的长和宽，就可以计算出不同矩形的面积，即可以计算出任意矩形的面积。可见，程序 2 比程序 1 灵活。

在程序设计中，要尽量少用或者不用第 1 种方法编写程序，而采用第 2 种方法编写灵活、通用的程序。

习 题

- 1-1 编写计算任意长方体体积的程序，程序名为 1-1.c。
- 1-2 用 1.3 节中所讲的编译、连接、运行和调试程序的方法，将 1-1.c 调试出正确的结果，形成 1-1.exe 文件。
- 1-3 用跟踪调试的方法调试 1-1.c，练习各种跟踪调试技术。
- 1-4 在命令行运行 1-1.exe，观察运行结果。

第 2 章 C 语言程序设计基础

2.1 基本数据类型

2.1.1 C 程序数据分类

在了解数据分类之前，必须清楚 C 程序的组成（见第 1 章），其中函数的组成如下：

```
函数名()          /* 如 main() */
{
    变量定义；
    执行部份；
}                } 函数体
```

在 C 程序的编写中，最重要的是函数体的编写，其内容包括以下两部分：

变量定义：即声明程序中要使用的变量、数组等。

执行部份：根据 C 语言的语法规则，编写的可执行语句。

变量的定义要指出变量的数据类型，C 语言的数据类型可分为如表 2.1 中所列的几种。

表 2.1 C 程序数据分类和常用的关键字

基本类型		构造类型		指针类型		空类型	
种类	关键字	种类	关键字	种类	关键字	种类	关键字
实型	整型	数组	int	指针	int	空类型	void
	单精度		float		float/double		
	双精度		double		char		
字符型	char	结构体	struct				
枚举型	enum	共用体	union				

注意：

C 语言中每种类型又有常量和变量之分。

程序中使用的变量必须用关键字指定其数据类型。

2.1.2 常量、符号常量与变量

1. 常量和符号常量

在程序运行过程中，其值不能被改变的量称为常量。常量也有类型之分，从其字面上就可以判断出类型。如：1, 2, 100 等为整型常量，12.23, -5.67, 1.230e3 等为实型常量，'a', 'b', 'c'为字符常量。

常量往往用在变量初始化及各种语句（如赋值语句和表达式）中。

在 C 程序中，常用一个标识符代表一个常量，如：

```
#define PI 3.14159    /* 定义符号常量 PI,它代表 3.14159,注意它们之间有空格 */
main()              /* 函数名 */
{
    int r=20;        /* 定义了一个整型变量 r,并把常量 20 放在其中 */
    float area;     /* 定义了一个实型变量 area,准备用来装计算结果 */
    area=PI*r*r;    /* 计算面积,放入 area 变量中。注意 PI 为符号常量 */
    printf("AREA=%7.2f\n",area); /* 把计算结果显示到屏幕上 */
}
```

程序中用#define 宏定义（宏定义详见第 7 章）定义 PI 代表 3.14159，因此程序中任何地方出现的 PI 都代表 3.14159。它与常量一样，可以参与任何合法的运算。

这种由一个标识符代表的常量，称为符号常量。注意它不同于变量，在其作用域（本例在主函数中）内不能改变它的值。使用符号常量的目的在于简化程序的书写和便于程序的修改。例如程序中有多个 3.14159 都要换成 6.28318，如果使用符号常量表示，则仅仅改动符号常量代表的值即可，即“一改全改”。

为了与程序中的变量名相区别，符号常量名一般用大写字母表示。

2. 变量

(1) 变量的定义

C 程序中常常使用变量进行各种运算。所谓变量，就是在程序运行过程中其值可以改变的量，它的作用是用来存放常量的。也就是说，他们是用来装数据的仓库，一般在函数体中可执行语句的前面定义。

C 程序中的变量必须遵循“先定义后使用的原则”。变量定义的一般形式为：

类型 变量名 1, 变量名 2, 变量名 3, ……;

说明：

类型为表 2-1 中的关键字，它明确指定了后面一系列变量的类型。

变量名 1, 变量名 2, 变量名 3 等，表明定义了多个变量，每个变量有自己的名字，名字的取法遵循标识符命名规则。各变量之间用逗号分隔开，最后一个变量名后为分号。

同一类型的变量可定义在一行内，也可定义在多行内。

如 `int a,b,c; /* 定义了 a,b,c 3 个整型变量 */`

也可以定义为：