



新编 **C** 语言程序设计

C Programming Newly Revised

宋 丽 蔡春华 蔡丽艳 编著

哈尔滨地图出版社



新编C语言程序设计

XINBIAN C YUYAN CHENGXU SHEJI

宋 丽 蔡春华 蔡丽艳 编著

哈尔滨地图出版社

·哈尔滨·

内 容 简 介

C 语言是在当今软件开发领域有着广泛应用的一种语言。编者基于 Visual C++ 平台, 根据多年的教学经验, 按照难点分解、平滑过渡、循序渐进的思路组织结构框架和编配内容。各章均配有大量的例题、习题, 难点、重点分析深入、透彻, 使读者通过对本书的学习, 不仅能正确了解 C 语言, 还能掌握初步的程序设计方法和技巧。

本书可作为高等院校本专科学生 C 语言程序设计课程的教材和教学参考书, 亦可供各类 C 语言学习对象阅读参考。

图书在版编目 (CIP) 数据

新编 C 语言程序设计/宋丽, 蔡春华, 蔡丽艳编著.
哈尔滨: 哈尔滨地图出版社, 2009.3
ISBN 978-7-5465-0034-8

I. 新… II. ①宋…②蔡…③蔡… III. C 语言-程序设计
IV. TP312
中国版本图书馆 CIP 数据核字 (2009) 第 039338 号

哈尔滨地图出版社出版发行

(地址: 哈尔滨市南岗区测绘路 2 号 邮政编码: 150086)

哈尔滨天兴速达印务有限责任公司印刷

开本: 787 mm×1 092 mm 1/16 印张: 13.25 字数: 331 千字

2009 年 3 月第 1 版 2009 年 3 月第 1 次印刷

ISBN 978-7-5465-0034-8

印数: 1~1 000 定价: 28.00 元

前 言

C 语言是一种应用广泛的程序设计语言。C 语言具有功能丰富、表达力强、使用灵活方便、便于大型程序开发、便于编写、可移植性好等优点,又由于 C 语言引入了反映计算机硬件特性的机制,使 C 语言也能编写直接控制计算机硬件设备的程序,所以 C 语言是一门既适合编写系统软件,又适合设计应用软件的高级语言。本书基于 Visual C++ 平台,编者根据多年的教学经验,按照难点分解、平滑过渡、循序渐进的思路组织结构框架和编配内容。各章均配有大量的例题、习题,难点、重点分析深入、透彻,使读者通过对本书的学习,不仅能正确了解 C 语言,还能掌握初步的程序设计方法和技巧。

全书共分为 10 章。内容包括:C 语言概述,C 语言数据类型、运算符与表达式,简单 C 语言的程序设计,选择结构程序设计,循环结构程序设计,编译预处理,数组,函数和指针,第 10 章内容为 C 语言上机实验。本书是编者在多年来从事 C 程序设计教学工作的基础上,结合教学实践中的经验,例举学生容易出现问题的典型例题,便于学生深入掌握重点内容。本书力求体系统结构安排合理、重点突出、难度适中;在语言叙述上注重概念清晰,通俗易懂;适应计算机教学实际需要。

本书编写分工如下:第 1 章和第 7 章由林利编写,第 2 章和第 3 章由蔡春华编写,第 4 章和第 5 章由高涇萍编写,第 6 章和第 8 章由蔡丽艳编写,第 9 章和第 10 章由宋丽编写。宋丽对全书进行统筹并定稿。

本书可作为高等院校本专科学学生 C 语言程序设计课程的教材和教学参考书,亦可供各类 C 语言学习对象阅读参考。

由于作者水平有限,加之时间仓促,书中错误和不当之处在所难免,敬请广大读者批评指正。

编 者
2009 年 1 月

目 录

第1章 C语言概述	1
1.1 C语言的发展	1
1.2 C语言特点	2
1.3 C语言的基本组成	3
1.3.1 字符集	3
1.3.2 标识符	3
1.4 简单的C语言程序	4
1.5 C程序在 Visual C++下的运行	6
1.5.1 Visual C++ 的安装与启动	6
1.5.2 输入与编辑源程序	7
习题	12
第2章 C语言数据类型、运算符与表达式	13
2.1 C语言的数据类型	13
2.2 常量	14
2.2.1 直接常量	14
2.2.2 符号常量	15
2.3 变量	16
2.3.1 变量	16
2.3.3 类型修饰符	16
2.3.4 变量的定义	17
2.4 数据类型转换	18
2.4.1 显式类型转换	18
2.4.2 隐式转换	18
2.5 运算符和表达式	19
2.5.1 运算符简介	19
2.5.2 算术运算符和表达式	20
2.5.3 赋值运算符和表达式	22
2.5.4 关系与逻辑运算符和表达式	23
2.5.5 逻辑运算符和表达式	24
2.5.6 其他运算符和表达式	25
习题	27
第3章 简单C语言的程序设计	31
3.1 C语言语句	31
3.2 赋值语句	32
3.2.1 赋值语句	32
3.2.2 赋值过程中的类型转换	32
3.3 数据输出、输入函数	33
3.3.1 格式输出函数 printf ()	33

3.3.2 格式输入函数 scanf ()	36
3.3.3 字符输出函数 putchar.....	38
3.3.4 字符输入函数 getchar.....	39
3.4 顺序结构程序设计.....	39
习题.....	41
第 4 章 选择结构程序设计.....	46
4.1 用 if 语句实现选择结构	46
4.1.1 if 语句有 3 种基本形式	46
4.1.2 if 语句的嵌套	49
4.2 switch 语句	50
习题.....	52
第 5 章 循环结构程序设计.....	61
5.1 While 语句	61
5.2 do-While 语句.....	63
5.3 for 语句.....	65
5.4 几种循环的比较.....	68
5.5 循环嵌套.....	69
5.6 用 break 语句和 continue 语句改变循环状态	70
5.6.1 用 break 语句提前退出循环	70
5.6.2 用 continue 提前结束本次循环	72
习题.....	72
第 6 章 编译预处理.....	82
6.1 宏定义.....	82
6.1.1 无参数的宏定义.....	82
6.1.2 带参数的宏定义.....	85
6.2 文件包含	89
6.3 条件编译.....	89
习题.....	92
第 7 章 数组.....	94
7.1 一维数组的定义及应用.....	94
7.1.1 一维数组的定义.....	94
7.1.2 一维数组元素的引用.....	94
7.1.3 一维数组的初始化.....	95
7.1.4 一维数组的应用举例.....	95
7.2 二维数组的定义及应用.....	99
7.2.1 二维数组的定义.....	99
7.2.2 二维数组元素的引用.....	100
7.2.3 二维数组的初始化.....	100
7.2.4 二维数组的应用举例.....	101
7.3 字符串与字符数组.....	102

7.3.1 字符串	102
7.3.2 字符数组	102
7.3.3 用于字符处理的函数	104
7.3.4 字符串处理程序设计举例	106
7.4 二维字符数组	108
7.4.1 二维字符数组的初始化	108
7.4.2 二维字符数组的赋值和引用	108
7.4.3 二维字符数组应用举例	108
习题	110
第8章 函数	116
8.1 函数的定义与调用	116
8.1.1 函数的定义	116
8.1.2 函数调用	118
8.1.3 函数的嵌套调用	119
8.2 递归函数	121
8.2.1 递归的概念	121
8.2.2 递归函数的概念	121
8.3 数组作函数参数	123
8.4 变量的作用域和函数的作用范围	129
8.4.1 变量作用域	129
8.4.2 函数作用范围	129
8.5 运行一个多文件的程序	132
8.6 变量的存储类别	133
8.6.1 局部变量的存储方式	134
8.6.2 全局变量的存储类别	136
习题	139
第9章 指针	145
9.1 指针的概念	145
9.2 指针变量的定义和引用	146
9.2.1 指针变量的定义	146
9.2.2 指针变量的引用	147
9.3 指针变量作为函数参数	149
9.4 指向一维数组的指针变量	151
9.4.1 一维数组指针的概念	151
9.4.2 数组元素地址作实参	152
9.4.3 数组名作函数参数	153
9.4.4 函数的指针形参和函数体中数组的区别	154
9.5 指向二维数组的指针变量	155
9.5.1 二维数组地址的概念	155
9.5.2 通过建立一个行指针来引用二维数组元素	156

9.5.3 二维数组名作为实参时, 实参和形参之间的数据传递	157
9.6 字符串与字符串指针变量	160
9.7 指针与函数	162
9.7.1 指向函数的指针变量	162
9.7.2 返回指针值的函数	163
9.8 指针数组	164
9.8.1 指针数组的定义及引用	164
9.8.2 指针数组作为实参时, 实参和形参之间的数据传递	165
9.9 指向指针的指针	166
9.10 指针数组作 main 函数的形参	167
习题	168
第 10 章 C 语言上机实验	172
实验一	172
实验二	173
实验三	176
实验四	177
实验五	178
实验六	180
实验七	181
实验八	182
附录 1 VC 编译错误信息	183
附录 2 C 语言中的关键字表	185
附录 3 运算符与结合性	186
附录 4 ASCII 码字符表	187
附录 5 模拟试题	188
C 语言程序设计模拟试题 A	188
C 语言程序设计模拟试题 B	193
C 语言程序设计模拟试题 C	198
参考文献	203

第 1 章 C 语言概述

学习目标

- 了解 C 语言的背景。
- 掌握 C 语言程序的结构。
- 掌握 C 语言的运行方法。
- 领会 C 语言程序设计的风格。

C 语言是目前世界上使用最广泛的通用计算机语言。用它既可编写计算机系统软件，也可编写各种应用软件，所以在数百种计算机语言中，C 语言仍然是目前最流行、最受欢迎的计算机语言。

现在，Windows 已成为计算机的主要操作系统，相应的基于 Windows 的程序开发多采用 C++，它虽是一种面向对象的语言，但其核心内容仍是标准 C 语言。

本章从 C 语言的基础出发，让读者对 C 语言有一个初步的认识。

1.1 C 语言的发展

以前操作系统等系统软件主要采用汇编语言编写。汇编语言依赖于计算机硬件，程序的可读性、可移植性都比较差。为了提高程序可读性和可移植性，人们希望采用高级语言编写这些软件，但是一般的高级语言难以实现汇编语言的某些操作，特别是针对硬件的一些操作（如：内存地址的读写、二进制位的操作）。人们设法寻找一种既具有一般高级语言特性，又具有低级语言特性的语言，C 语言就在这种情况下应运而生。其发展过程如下：

1963 年，英国剑桥大学在 ALGOL60 程序设计语言基础上，推出了 CPL 语言（组合式程序设计语言）。

因 CPL 规模大、实现难等原因，1967 年 Martin Richards 对 CPL 作了简化，推出了 BCPL（基本组合式程序设计语言）。

1970 年，美国贝尔实验室的 Ken Thompson 又在 BCPL 的基础上，再次作进一步的简化，设计出简单又接近硬件的 B 语言，并用 B 语言写了 UNIX 操作系统和大量的实用程序。

由于 B 语言只有单一的字类型、过于简单等原因而未能流行。D. M. Ritchie 从 1971 年开始在 B 语言基础上设计了 C 语言，并于 1972 年开始使用。

1973 年 K. Thompson 和 D. M. Ritchie 把 UNIX 系统用 C 语言重写了一遍，增加了多道程序设计功能，使整个系统，包括 C 语言的编译程序都建立在 C 语言的基础上。第五版 UNIX 系统成为 UNIX 系统广泛应用的基础。

到 1975 年，UNIX 第六版问世。随着 UNIX 的巨大成功和被广泛移植到各种机器上，C 语言也被人们所接受，并移植到大、小、微型机上。C 语言已风靡全世界，成为世界上应用最广泛的计算机程序设计语言之一。

1983 年，美国国家标准化协会（ANSI）对 C 语言的各种版本作了扩充和完善，制定

了 C 语言的标准,称为 ANSI C。本书的叙述基本上以 ANSI C 为基础。目前,广泛流行的各种版本的 C 语言编译系统在非主要部分中稍有不同。

IBM PC 机 DOS, Windows 平台上常见的 C 语言版本有:

- Borland 公司
Turbo C, Turbo C++, Borland C++;
C++ Builder (Windows 版本)。
- Microsoft 公司
Microsoft C;
Visual C++ (Windows 版本)。

1.2 C 语言特点

C 语言集中了一般高级语言和汇编语言的优点,用它能方便地编写不依赖于计算机硬件设施的各种应用程序,又能用它编写包括操作系统在内的各种系统程序。C 语言具有多方面的特点,主要的有以下几个方面。

(1) 语言表达能力强

C 语言包含 34 种运算符,有的运算符反映了当前计算机的性能,包含可直接由硬件实现的算术逻辑运算符,足以取代汇编语言编写各种系统程序和应用程序。C 语言的运算符类型极其丰富,有数值运算、字运算、位运算和地址运算等。

(2) 含有丰富的数据类型,具有类型构造能力和结构化的程序控制结构

C 语言能在字符、整数、浮点数等基本类型基础上按结构化的层次构造方法构造数组、结构和联合等各种结构化的数据类型。特别是 C 语言的指针类型灵活多样,非常有助于构造链表、树、图等复杂的数据结构。另一方面,它的结构化程序控制结构符合结构化程序设计的要求,可编写结构良好的程序。此外,它的数据的静态和外部存储类机制有助于信息隐蔽和抽象的模块化程序设计。

(3) 语言简洁、紧凑,使用方便灵活

用 C 语言编写的程序通常比用其他高级语言编写的程序更简练,代码行少。C 语言没有 I/O 设施,也没有并行操作、同步或协同程序等复杂控制。另外, C 语言程序在运行时所需要的支持少,占用的存储空间也小。

(4) 能使编译程序产生执行效率高的代码

一个高级语言能否用来描述系统程序,除语言表达能力之外,还有能否产生高质量的代码这个重要因素。许多高级语言相对汇编语言而言其代码的执行效率要低得多。但 C 语言则不然,许多试验表明,用 C 语言描述较用汇编语言描述,其代码执行效率只低 10%~20%,而用 C 语言编程比用汇编语言编程迅速得多,程序的可读性又高,特别是 C 语言程序比较容易移植。所以 C 语言成了人们描述系统程序和应用程序比较理想的工具。

(5) 用 C 语言编写的程序可移植性较好

程序的可移植性是指在一个环境上运行的程序可以不加或稍加改动后在另一个完全不同的环境上运行。汇编语言依赖于机器硬件,用汇编语言编写的程序不可移植。而有些高级语言,因它们的编译程序不可移植,影响了用它们编写的程序的广泛应用。目前在许多

机器上都有C语言编译系统,且大部分是由C语言编译移植得到的。由于C语言的编译程序便于移植,也就更提高了C语言程序的可移植性。

(6) C语言也有一些不足之处

例如,用C语言编写程序,自由度大(如对变量的类型约束不够严格)、整型和字符型及逻辑型数据可通用、指针和数组可通用等。过多的通用性限制了编译程序对C语言程序作充分的句法和语义检查,可能会无视某些编程人员使用上的失误,依旧能正常编译,从而不能及时发现程序中的错误,给程序的调试和排错造成一些困难。另外,C语言的运算符优先级太多,不便于记忆,有些还与常规约定有所不同;类型检验弱、数据类型转换比较随便等,影响了程序的安全性。

1.3 C语言的基本组成

一门自然语言,如汉语,有汉语的词汇和一套语法规则。C语言也有一套基本的词汇和一套严格的语法规则,按照语法规则将词汇组成语言的各种成分(如表达式、语句、定义和说明等)。C语言的基本词汇可分为字面形式常量(如100,15)、特殊符号和标识符等,其中每个基本词汇由一个或若干个字符组成。

根据C语言的特点,规定了其所需的基本符号和标识符。

1.3.1 字符集

字符是C语言的最基本的元素,C语言字符集由字母、数字、空白、标点和特殊字符组成(在字符串常量和注释中还可以使用汉字等其他图形符号)。由字符集中的字符可以构成C语言进一步的语法成分(如标识符、关键词、运算符等)。

组成C语言基本词汇的基本字符有:

(1) 数字10个(0~9)。

(2) 英文字母大、小写各26个(A~Z, a~z)。

(3) 空白符:是指空格、制表符(跳格)、换行符(空行)的总称。空白符除了在字符、字符串中有意义外,编译系统忽略其他位置的空白。空白符在程序中只是起到间隔作用。在程序的恰当位置使用空白符将使程序更加清晰,增强程序的可读性。

(4) 标点符号、特殊字符: ! # % ^ & + - * / = _ ~ < > \ | . , ; : ? ' " () [] { } 。

1.3.2 标识符

(1) 标识符的概念

标识符是起标识作用的一类符号。用来标识变量名、符号常量名、函数名、数组名、类型名等实体(程序对象)的有效字符序列。

在C语言中,一个合理的标识符只能由字母、数字和下划线三种字符组成,且第一个字符必须为字母或下划线。例如:sum, average, _total, Class, day, stu_name, p4050为合法的标识符, M.D. John, \$123, #33, 3D64, a-b为不合法的标识符。

ANSI C没有限制标识符长度,但各个编译系统都有自己的规定和限制。如Turbo C

变量名最大长度为 32 个字符, MS C 则为 8 个字符。为了程序的可移植性以及阅读程序的方便, 建议变量名的长度最好不要超过 8 个字符。

注意: 编译系统将大写字母和小写字母认为是两个不同的字符, 因此, SUM 和 sum 是两个不同的标识符。但通常标识符命名时应当有一定的意义, 做到“见名知义”。

不同的 C 语言系统对标识符的有效字符个数有不同的规定。一般取 8 个字符; 也就是说, 两个超过 8 个字符的不同标识符, 当前 8 个字符完全一样时, 系统就认为它们是同一个标识符, 而不加以区别。

(2) 标识符的分类

C 语言的标识符可以分为三类。

①关键字

所谓关键字是指它们用来表示 C 语言本身的特定成分, 具有相应的语义, 不能再另做他用。所有关键字都使用小写英文字母表示, 共 32 个:

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					

②预定义标识符

C 语言还使用一类特殊含义的标识符, 被用做库函数名和预编译命令。但使用时前面应加“#”。如 define elif else endif error if ifdef ifndef include
line pragma undef 等。

③自定义标识符

自定义标识符是用户根据自己需要定义的一类标识符, 用来标识符号常量、变量、自定义函数名等。

需要注意的是用户自定义标识符时不允许用关键字, 也建议不把预定义标识符作为用户自定义标识符。

1.4 简单的 C 语言程序

例 1.1 打印一个语句。

```
#include <stdio.h>
void main()
{
    printf("A simple c program. \n");
}
```

经编译后运行结果如下:

A simple c program.

程序说明:

main 表示“主函数”。main 前面的 void 表示此函数是“空类型”, 即执行此函数后不产生一个函数值。每个 C 语言程序都必须有一个 main 函数, 它是每一个 C 语言程序的执

行起始点（入口点）。用{}括起来的是“主函数”main的函数体。main函数中的所有操作（或：语句）都在这一对{}之间。也就是说main函数的所有操作都在main函数体中。本例主函数中只有一条语句，printf是C语言库函数中的输出函数，功能是把该语句中双撇号内的字符原样输出（显示在屏幕上），“\n”是换行符，即在输出“A simple c program.”后回车换行。语句最后必须有一个分号。

include是文件包含命令，扩展名为.h的文件称为头文件。在本例题中，因为用到printf输出函数，因此在程序的开头写上#include <stdio.h>。

例 1.2 求两数中的小者。

```
#include <stdio.h>

void main()                /* 主函数 */
{
    int a, b, c;            /* 定义变量 */
    scanf("%d %d", &a, &b); /* 输入数 */
    c=min(a, b);            /* 调用函数 */
    printf("min=%d\n", c);  /* 输出 */
}

int min(int x, int y)       /* 定义函数 */
{
    int z;                  /* 定义局部变量 */
    if (x>y) z=y;
    else z=x;
    return(z);              /* 返回z值*/
}
```

经编译后运行如下：

14, 4 ✓

min=4

再次运行：

6, 29 ✓

min=6

本程序包括两个函数：主函数main()和子函数min()。min函数的作用是将x, y中较小的数送给变量z, return语句的作用是将z的值返回给主函数main。返回值是通过函数名min带回给main中的调用处。

主函数中的scanf()是系统给出的标准输入函数，通过它输入a和b的值，&a, &b中的“&”是取地址符，在此功能是将输入的值分别送入变量a和b的地址所标志的存储单元中，即输入给变量a和b。“%d, %d”是输入输出格式字符串，用来指定输入输出数据的类型和格式。

main函数中第4行是调用min函数，在调用时将实际参数a, b的值送给min函数中的形式参数x和y。在min函数后得到一个返回值（即min函数中的变量z的值），把这个值送给变量c，然后输出变量c的值。

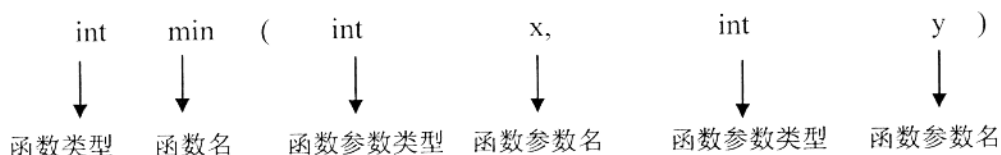
通过上述两个例子，我们可以看到：

(1) C 程序是由函数构成的。一个程序至少含有一个 main 主函数。也可以包含一个主函数和若干个子函数。函数是 C 语言的基本单位。被调用的子函数可以是系统给出的库函数，也可以是用自己定义的函数。

(2) 一个函数由两部分组成：函数的首部 and 函数体。

①函数首部，也应是函数的第一行。包含函数类型、函数名、函数属性、函数参数名（形式参数）、参数类型等。

以例 1.2 中的 min 函数来介绍，



函数参数可以没有，但是函数后必须带一对圆括弧数。如：main ()。

②函数体

函数体是函数首部下面花括弧中的内容，包含声明部分和执行部分。声明部分用于定义函数中所使用的变量，执行部分由若干条语句组成。在某些情况下也可以没有声明部分，甚至也可以没有执行部分。

例如

```
main( )
{
}
```

也是一个合法的 C 语言程序。

③main 函数通常位于程序之首，实际上它位于程序的开头、最后及函数与函数之间均是合法的，但不管在什么位置，一个 C 程序总是从 main 函数开始执行的。

④C 程序书写格式自由，一行内可以写几个语句，一个语句可以分写在多行上。例如：

```
int a, b;
a=5;   b=8;    /*两个语句在一行*/
printf("a=%d, b=%d\n",
      a, b); /*一个语句在两行*/
```

⑤C 语言本身没有输入输出语句，输入输出都由函数实现。

⑥可以用 /* ... */ 对 C 程序中的任何部分作注释，增强程序的可读性。

⑦C 语言程序中每一条语句结束后都必须加一个分号。分号是 C 语言语句的组成部分。

1.5 C 程序在 Visual C++ 下的运行

1.5.1 Visual C++ 的安装与启动

Visual C++ 6.0 是在 Windows 环境中工作的。本节介绍的是 Visual C++ 6.0 中文版。

为了使用 Visual C++ 6.0 集成环境，必须事先在所用的计算机上安装 Visual C++ 6.0 系统。在安装后最好在桌面上设立 Visual C++ 6.0 的快捷方式图标，以方便使用。

双击桌面上 Visual C++ 6.0 图标, 就能进入 Visual C++ 6.0 集成开发环境, 屏幕上出现 Visual C++ 6.0 的主窗口, 如图 1-1。

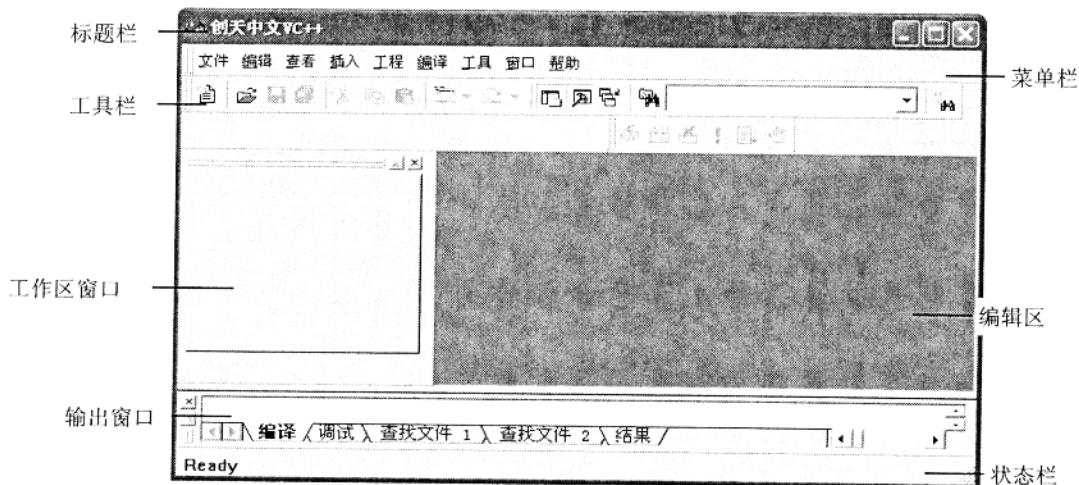


图 1-1 Visual C++ 6.0 主窗口

1.5.2 输入与编辑源程序

(1) 新建一个源程序

在 Visual C++ 6.0 主菜单栏中选择“文件”选项卡, 然后选择“新建”项, 如图 1-2。

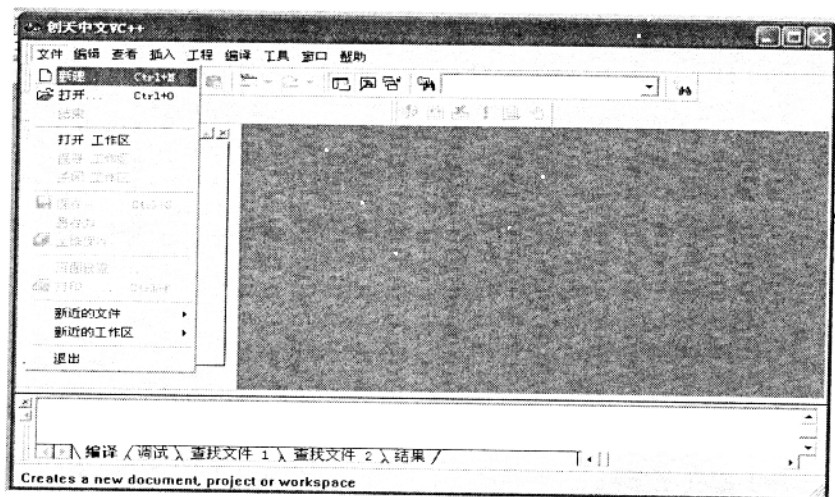


图 1-2 “文件”菜单下的“新建”项

单击“新建”后屏幕上出现一个新建对话框。选择对话框的上方文件选项卡, 在其下拉菜单中选择“C++ Source Files”项, 如图 1-3。然后在对话框右半部的文件中输入编辑的源程序文件名字 (如 b.c)。

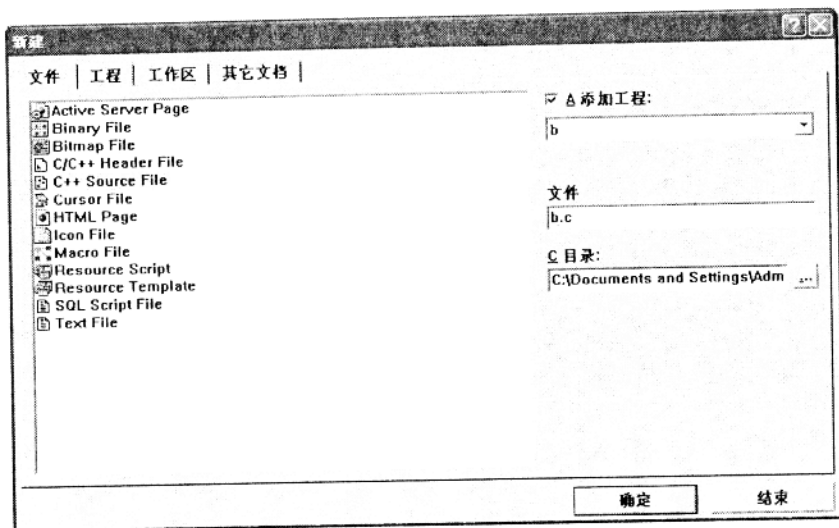


图 1-3 “新建”对话框

则 b.c 文件保存在默认目录下。当然，我们也可以通过目录指定其文件的存储路径。单击“确定”按钮后，在编辑区输入源程序，如图 1-4。

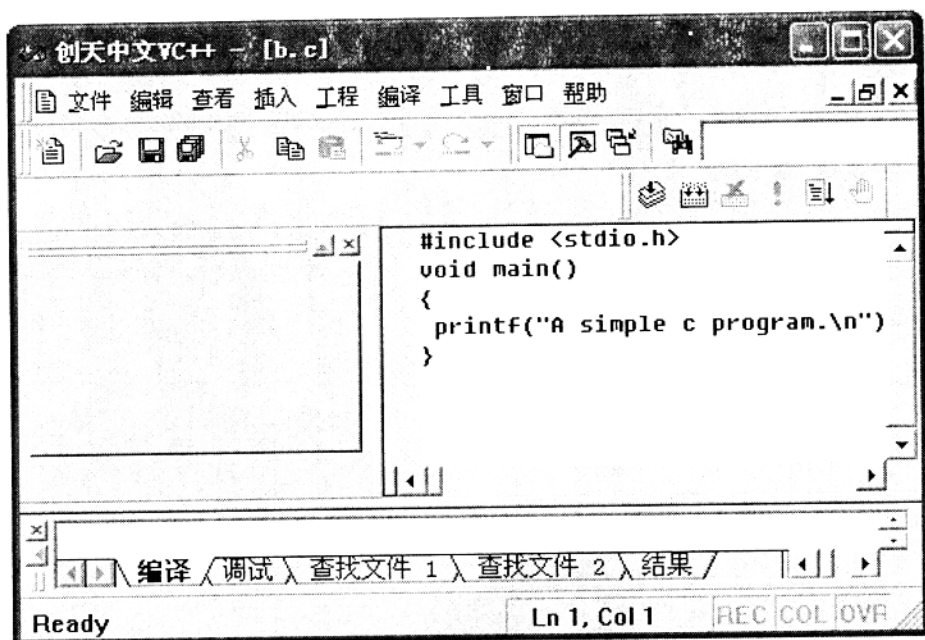


图 1-4 程序在编辑区的输入

注意：源程序一定要在英文状态下输入，即字符标点都要在半角状态下，同时注意大小写，一般都用小写。

输入完毕后，在主菜单栏中选择“文件”下的“保存”项，如图1-5。



图 1-5 程序的保存

(2) 程序的编译

在编辑和保护了源文件以后，选择主菜单栏中的“编译”，在其下拉菜单中选择编译项。在产生的工作区对话框中，选择“是”，如图1-6。

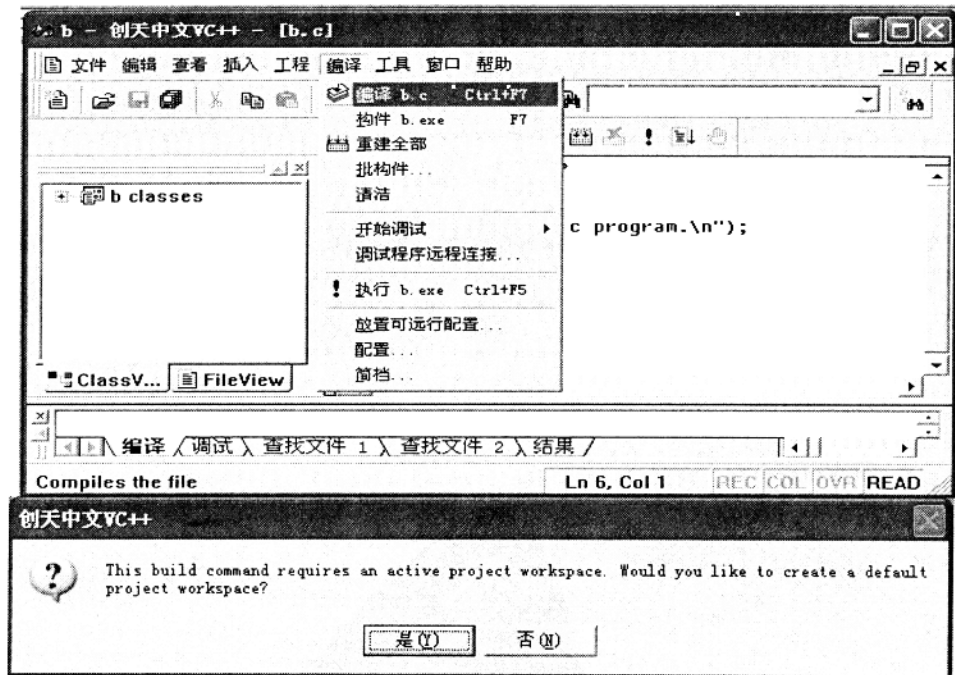


图 1-6 文件的编译