



普通高等教育“十二五”规划教材
21世纪大学计算机基础分级教学丛书

C 语言

(第二版)

程序设计教程

王绪梅 詹春华 陈剑锋 主编



科学出版社

普通高等教育“十二五”规划教材
21世纪大学计算机基础分级教学丛书

C 语言程序设计教程

(第二版)

王绪梅 詹春华 陈剑锋 主编

科学出版社

北京

版权所有,侵权必究

举报电话:010-64030229;010-64034315;13501151303

内 容 简 介

本书是为将C语言作为第一门程序设计课程的学生编写的教材或参考资料,以培养学生的程序设计能力与掌握开发工具为目标,针对初学者的特点,在内容编排、阐述、实验、习题的选择上做了较细致的构思、设计,有利于自学,更便于教学。

本书从程序设计的基本思想和C语言的设计基本原理出发,贯穿“基础—应用”这一主线,紧扣基础,循序渐进,面向应用。为方便授课和学习、加强实践能力的培养,配合本书我们编写了《C语言程序设计实验与习题(第二版)》与本书相呼应。

全书共分10章,系统地介绍了程序设计语言和C语言程序设计的基础理论,重点介绍了C语言程序设计的基本知识和编程思想与方法。其宗旨是让读者学会分析问题、掌握基本的编程思想与程序代码设计。本书加强基础、强化实践、突出重点、注意难点处理,使读者易学易懂。

本书可供高等学校、水平考试、各类成人教育等教学使用,也可供计算机爱好者自学。

图书在版编目(CIP)数据

C语言程序设计教程/王绪梅,詹春华,陈剑锋主编.—2版.—北京:科学出版社,2015.1

(21世纪大学计算机基础分级教学丛书)

普通高等教育“十二五”规划教材

ISBN 978-7-03-042590-4

I. C… II. ①王…②詹…③陈… III. C语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆CIP数据核字(2014)第272471号

责任编辑:闫 陶/责任校对:肖 婷

责任印制:高 嵘/封面设计:陈明亮

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

武汉市新华印刷有限责任公司印刷

科学出版社发行 各地新华书店经销

*

2015年1月第 一 版 开本:787×1092 1/16

2015年1月第一次印刷 印张:19

字数:420 000

定价:40.00元

(如有印装质量问题,我社负责调换)

前 言

C 语言是目前国内外广泛使用的程序设计语言之一,是国内外大学都在开设的计算机程序设计核心基础课之一。C 语言具有功能丰富、表达能力强、使用方便灵活、程序执行效率高并且可移植性好等特点,学习 C 语言已经成为广大计算机应用人员和高校开设计算机语言课程的首选。

为了推进 21 世纪计算机基础教育改革,从素质教育的理念出发,结合信息化社会对高素质、复合型人才的需求,特出版此书。由于 C 语言涉及的概念多而且复杂、规则繁多、使用灵活但易出错,不少初学者感到难以掌握,本书力求较通俗、较为全面地介绍 C 语言的基本内容与程序设计思想,是专为初学程序设计者编写的一本入门教材。

本书编写时力求概念准确、讲述简单明了、内容实用。突出基础,面向应用,循序渐进地引导读者学习程序设计的思想和方法。明确突出每个章节的重点、注意难点处理,使读者易学易懂。

本书的特点是精选内容、分散难点、例题丰富、通俗易懂;通过算法设计的介绍,可以使读者更好地学习程序设计的思想、体系结构和方法,尤其是优化的程序设计方法,读者从实际应用中 C 语言程序设计的基本知识得以融会贯通和进一步提高;书中例题所有代码均在 VC 环境下调试通过,方便读者自学;本书将“编译预处理”内容放到附录中去,由于此部分内容跨越很多知识点,如果单独作为一章内容,那么只能将这部分内容放到较靠后的位置。当基础知识的学习中要涉及编译预处理的内容时,可以根据需要及时查询,读者也可以根据实际情况安排学习编译预处理内容的进度。

全书共分 10 章,第 1 章介绍了程序设计的基本概念和 C 语言的基础知识;第 2 章介绍了 C 语言的数据类型;第 3~5 章讲述了 C 语言的控制结构;第 6 章介绍了数组;第 7 章介绍了函数,第 8 章介绍指针,第 9 章介绍了结构体与共同体,第 10 章介绍了文件。由于各专业的学时数不尽相同,有些章节用“*”进行了标注,供教师选择讲授。

为了加强实践能力的培养,配合本书还编写了《C 语言程序设计实验与习题(第二版)》与本书相呼应,各章安排了习题和上机实验内容,以方便师生学习,从而达到较好的效果。

本书由王绪梅、詹春华、陈剑锋主编,本书在编写过程中,参考了大量的文献资料,在此向这些文献资料的作者表示感谢。由于时间仓促和水平所限,书中难免有欠妥之处,敬请各位专家、读者不吝批评指正。

本书可供高等学校、计算机水平考试培训、各类成人教育学校作为开设程序设计课程的教材,也可供计算机爱好者自学。

编 者

2014 年 4 月

目 录

| | |
|--|----|
| 第 1 章 C 语言概述 | 1 |
| 1.1 C 语言简史 | 1 |
| 1.2 C 语言特点 | 1 |
| 1.3 C 程序基本结构 | 2 |
| 1.4 Microsoft Visual C++ 6.0 集成开发环境的使用 | 5 |
| 1.5 算法 | 10 |
| 1.5.1 算法的基本概念 | 10 |
| 1.5.2 算法中的基本结构及其描述 | 12 |
| 1.6 结构化程序设计方法 | 14 |
| 习题一 | 15 |
| 第 2 章 C 语言的数据表示 | 17 |
| 2.1 C 语言的数据类型 | 17 |
| 2.1.1 C 语言的字符集、标识符与关键字 | 17 |
| 2.1.2 C 语言的数据类型概述 | 18 |
| 2.1.3 C 语言的基本数据类型 | 19 |
| 2.2 常量与变量 | 21 |
| 2.2.1 常量及其数据类型 | 21 |
| 2.2.2 变量及变量定义 | 23 |
| 2.3 运算符及其表达式 | 25 |
| 2.3.1 算术运算符和算术表达式 | 25 |
| 2.3.2 赋值运算符和赋值表达式 | 29 |
| 2.3.3 关系运算符和关系表达式 | 30 |
| 2.3.4 逻辑运算符和逻辑表达式 | 32 |
| 2.3.5 逗号运算符和逗号表达式 | 34 |
| 2.3.6 条件运算符和条件表达式 | 35 |
| 2.3.7 长度运算符——sizeof() | 36 |
| 2.3.8 位运算符 | 36 |
| 2.4 数据类型转换 | 39 |
| 2.4.1 类型自动转换 | 39 |
| 2.4.2 赋值转换 | 39 |
| 2.4.3 强制类型转换 | 40 |
| 习题二 | 41 |

| | |
|--------------------------------|----|
| 第3章 顺序结构程序设计 | 45 |
| 3.1 C语言的基本语句 | 45 |
| 3.1.1 表达式语句 | 45 |
| 3.1.2 函数调用语句 | 46 |
| 3.1.3 控制语句 | 46 |
| 3.1.4 空语句 | 47 |
| 3.1.5 复合语句 | 47 |
| 3.2 格式化输出 | 47 |
| 3.2.1 格式化输出函数 printf() | 47 |
| 3.2.2 格式控制 | 48 |
| 3.3 格式化输入 | 50 |
| 3.3.1 格式输入函数 scanf | 50 |
| 3.3.2 格式控制 | 51 |
| 3.4 字符输入输出函数 | 52 |
| 3.4.1 putchar 函数(字符输出函数) | 52 |
| 3.4.2 getchar 函数(字符输入函数) | 53 |
| 3.5 顺序结构 | 53 |
| 3.6 顺序结构程序设计举例 | 55 |
| 习题三 | 57 |
| | |
| 第4章 选择结构程序设计 | 63 |
| 4.1 if 语句 | 63 |
| 4.1.1 单分支 if 语句 | 64 |
| 4.1.2 双分支 if 语句 | 64 |
| 4.1.3 多分支 if 语句 | 65 |
| 4.1.4 对 if 语句的有关说明 | 67 |
| 4.1.5 if 语句的嵌套 | 67 |
| 4.2 switch 语句 | 69 |
| 4.3 选择结构程序设计举例 | 71 |
| 习题四 | 76 |
| | |
| 第5章 循环结构程序设计 | 83 |
| 5.1 while 循环 | 83 |
| 5.2 do-while 语句 | 84 |
| 5.3 for 语句 | 85 |
| 5.4 循环嵌套 | 87 |
| 5.5 break 和 continue 语句 | 88 |
| 5.5.1 break 语句 | 88 |
| 5.5.2 continue 语句 | 89 |
| 5.6 循环结构程序设计举例 | 90 |
| 习题五 | 92 |

| | |
|------------------------------|-----|
| 第 6 章 数组 | 102 |
| 6.1 数组的概念 | 102 |
| 6.1.1 数组的引入 | 102 |
| 6.1.2 数组的概念 | 103 |
| 6.2 一维数组的定义和引用 | 103 |
| 6.2.1 一维数组的定义 | 103 |
| 6.2.2 一维数组元素的引用 | 104 |
| 6.2.3 一维数组的赋值 | 106 |
| 6.2.4 一维数组的输入和输出 | 106 |
| 6.3 二维数组的定义和引用 | 109 |
| 6.3.1 二维数组的定义 | 109 |
| 6.3.2 二维数组在计算机内存中的存放次序 | 109 |
| 6.3.3 二维数组元素的引用 | 110 |
| 6.3.4 二维数组的初始化 | 111 |
| 6.3.5 二维数组的输入和输出 | 112 |
| 6.4 字符数组 | 114 |
| 6.4.1 字符数组的定义 | 114 |
| 6.4.2 字符数组的初始化 | 114 |
| 6.4.3 字符数组元素的引用 | 114 |
| 6.4.4 字符串和字符串结束标志 | 116 |
| 6.4.5 字符串的输入与输出 | 118 |
| 6.4.6 字符串常用函数 | 121 |
| 6.5 程序举例 | 124 |
| 习题六 | 130 |
| | |
| 第 7 章 函数 | 135 |
| 7.1 概述 | 135 |
| 7.2 函数定义的一般形式 | 137 |
| 7.2.1 无参函数定义的一般形式 | 137 |
| 7.2.2 有参函数定义的一般形式 | 137 |
| 7.3 函数参数和函数的值 | 139 |
| 7.3.1 函数的参数 | 139 |
| 7.3.2 函数的返回值 | 141 |
| 7.4 函数的调用 | 143 |
| 7.4.1 函数调用的一般形式 | 143 |
| 7.4.2 对被调用函数的声明和函数原型 | 144 |
| 7.5 函数的嵌套调用 | 147 |
| 7.6 递归函数和函数的递归调用 | 149 |
| 7.7 数组作为函数参数 | 151 |
| 7.7.1 数组元素作函数实参 | 151 |

| | | |
|--------------|---------------------|------------|
| 7.7.2 | 数组名作为函数参数 | 152 |
| 7.8 | 变量的作用域与存储方式 | 156 |
| * 7.9 | 变量的存储类型 | 160 |
| 7.9.1 | 动态存储方式与静态存储方式 | 160 |
| 7.9.2 | 自动变量 | 161 |
| 7.9.3 | 外部变量 | 163 |
| 7.9.4 | 静态变量 | 164 |
| 7.9.5 | 寄存器变量 | 167 |
| 7.9.6 | 存储类别小结 | 168 |
| * 7.10 | 内部函数和外部函数 | 168 |
| 7.10.1 | 内部函数 | 168 |
| 7.10.2 | 外部函数 | 169 |
| 7.11 | 程序举例 | 172 |
| | 习题七 | 176 |
| 第 8 章 | 指针 | 185 |
| 8.1 | 指针和指针变量 | 185 |
| 8.1.1 | 指针的概念 | 185 |
| 8.1.2 | 指针变量的定义与初始化 | 186 |
| 8.1.3 | 指针变量的运算 | 189 |
| 8.1.4 | 多级指针 | 190 |
| 8.2 | 指针与数组 | 191 |
| 8.2.1 | 指针与一维数组 | 191 |
| 8.2.2 | 指针与二维数组 | 194 |
| 8.3 | 指针与函数 | 199 |
| 8.3.1 | 指针或者数组名作为函数的参数 | 199 |
| 8.3.2 | 返回指针的函数 | 203 |
| 8.3.3 | 指向函数的指针 | 203 |
| 8.4 | 指针与字符串 | 207 |
| 8.4.1 | 字符串指针 | 207 |
| 8.4.2 | 字符指针与字符数组的区别 | 209 |
| 8.5 | 指针数组 | 210 |
| 8.5.1 | 指针数组 | 210 |
| 8.5.2 | 命令行参数 | 212 |
| 8.6 | 程序举例 | 213 |
| | 习题八 | 215 |
| 第 9 章 | 结构体、共用体和枚举类型 | 224 |
| 9.1 | 结构体类型基础 | 224 |
| 9.2 | 结构体类型变量 | 225 |

| | | |
|---------------|----------------------------|------------|
| 9.2.1 | 定义结构体类型变量及变量的初始化 | 225 |
| 9.2.2 | 结构体变量的存储结构 | 226 |
| 9.2.3 | 结构体变量的使用 | 227 |
| 9.2.4 | 结构体变量的输入与输出 | 228 |
| 9.3 | 结构体类型数组 | 228 |
| 9.3.1 | 结构体类型数组的定义 | 229 |
| 9.3.2 | 结构体类型数组的初始化 | 229 |
| 9.3.3 | 结构体元素的操作 | 229 |
| 9.4 | 指向结构体的指针和链表 | 230 |
| 9.4.1 | 指向结构体变量的指针 | 231 |
| 9.4.2 | 指向结构体数组的指针变量 | 232 |
| 9.4.3 | 链表 | 233 |
| 9.5 | 共用体类型 | 239 |
| 9.5.1 | 共用体类型和共用体变量的定义 | 240 |
| 9.5.2 | 共用体变量的引用 | 241 |
| 9.6 | 枚举类型 | 242 |
| | 习题九 | 244 |
| 第 10 章 | 文件 | 255 |
| 10.1 | 文件的基本概念 | 255 |
| 10.1.1 | 文件的类型 | 255 |
| 10.1.2 | 缓冲区 | 256 |
| 10.2 | 文件的打开和关闭 | 257 |
| 10.2.1 | 文件类型指针 | 257 |
| 10.2.2 | 文件的打开 | 257 |
| 10.2.3 | 文件的关闭 | 259 |
| 10.3 | 文件的读写操作 | 259 |
| 10.3.1 | 字符读写的函数 fgetc()、fputc() | 260 |
| 10.3.2 | 字符串读写的函数 fputs()、fgets() | 261 |
| 10.3.3 | 格式化读写函数 fscanf()、fprintf() | 264 |
| 10.3.4 | 数据块读写函数 fwrite()、fread() | 266 |
| 10.3.5 | 文件读写函数选用原则 | 269 |
| 10.4 | 文件的其他常用函数 | 269 |
| 10.5 | 综合应用举例 | 270 |
| | 习题十 | 272 |
| 附录 I | 编译预处理 | 279 |
| 附录 II | ASCII 表 | 283 |
| 附录 III | C 运算符的优先级与结合 | 284 |
| 附录 IV | Turbo C 常用库函数 | 286 |

第 1 章 C 语言概述

核心内容:

1. C 语言简史
2. C 语言的主要特点
3. C 语言的基本结构
4. Microsoft Visual C++ 6.0 安装和启动方法
5. Microsoft Visual C++ 6.0 集成开发环境
6. 利用 Microsoft Visual C++ 6.0 开发 C 程序的步骤

1.1 C 语言简史

C 语言是目前最流行、最有影响力的程序设计语言之一,在众多的程序开发领域中都可以看到它的身影。

20 世纪 60 年代,贝尔实验室的 Ken Thompson 试图开发一个称为 UNIX 的操作系统,这也是后来对计算机业界产生了巨大影响的操作系统。起初使用的是汇编语言,但是由于汇编语言的移植性不高,用汇编语言编写的程序移植起来十分头疼。Ken Thompson 于是就采用了一个叫 BCPL(Basic combined programming language)的语言来进行开发,他对 BCPL 进行了整合,形成了 B 语言。1971 年,Dennis Ritchie 开始协助 Thompson 开发 UNIX,他对 B 语言进行了改良,加入了新的数据类型和语法,就形成了大名鼎鼎的 C 语言(取 BCPL 的第二个字母)。

由于 C 语言是一种高级语言,具有很好的移植性,Ken Thompson 与 Dennis Ritchie 用 C 语言重写的 UNIX 操作系统修改、移植起来就十分方便,为 UNIX 日后的普及打下了坚实的基础。同时随着 UNIX 的逐渐普及,C 语言在全世界推广开来。UNIX 和 C 语言完美的结合成为一个统一体,相辅相成,成就了软件开发史上历时几十年的一个时代。

在 C 语言的推广过程中,出现了许多不同的版本。为了统一各种不同版本 C 语言之间的兼容问题,美国国家标准化协会(American National Standards Institute)1983 年发表制定了一个 C 语言标准,称之为 ANSI C,1990 年国际标准化组织 ISO(International Standards Organization)接收了 ANSI C 为 ISO C 的标准。目前流行的 C 语言编译系统都是以 ANSI C 为基础的。在微型机上使用的 Microsoft C、Turbo C、Quick C 等,它们的不同版本又略有差异。因此,读者在学习时需要了解所用的计算机系统配置的 C 的变异系统的特点和规定,本教材中运行的例子均在 Visual C 环境下实现。

现今有很多程序设计语言也是在 C 语言的基础上产生的,如 C++、Visual C++、JAVA、C# 等。

1.2 C 语言特点

C 语言从出现至今,经过了几十年的时间,在面向对象程序设计语言如此风靡的今天,依

然有大量的程序开发人员和计算机编程爱好者使用它进行程序开发,正是由于 C 语言具有一些不可替代的特点:

(1) 运算符、数据类型丰富。

C 语言的运算符非常丰富,共有 34 种运算符。C 语言把括号、赋值、强制类型转换等都作为运算符处理。从而使 C 语言的运算类型极其丰富,表达式类型多样化。灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

C 语言具有现代化语言的各种数据类型,数据类型有:整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等。能用来实现各种复杂的数据结构,并引入了指针概念,使程序效率更高。

(2) 语言简洁紧凑、使用灵活方便。

C 语言一共只有 32 个关键字、9 种控制语句,程序书写形式自由,主要用小写字母表示,压缩了一切不必要的成分。一行中可以书写多个语句,一条语句可以写在不同行上,可以采用宏定义和文件包含等预处理语句等,这些都使得 C 语言简洁紧凑。

(3) C 语言具有结构化的控制语句。

C 语言提供了一套完整的结构控制语句(例如选择、循环等)和构造数据类型(例如数组、结构体等),使得程序流程具有很好的结构性。

C 语言的主要组成结构是函数,函数是 C 语言的基本模块。项目的各种功能可以利用不同功能的函数实现,从而达到结构化程序设计中模块化的要求。

(4) C 语言允许直接访问物理地址,对硬件进行操作。

C 语言允许直接访问物理地址,可以直接对硬件进行操作,能实现汇编语言的大部分功能,因此它既具有高级语言的功能,又具有低级语言的许多功能,能够像汇编语言一样对位、字节和地址进行操作,可用来写系统软件。

(5) 生成的目标代码质量好,程序执行效率高。

(6) C 语言适用范围大,可移植性好。

C 语言有一个突出的优点就是适合于多种操作系统,也适用于多种机型。C 语言具有强大的绘图能力,可移植性好,并具备很强的数据处理能力,因此适于编写系统软件,三维、二维图形和动画,它也是数值计算的高级语言。

1.3 C 程序基本结构

现在通过几个例子,我们来分析一下 C 程序的基本结构。

【例 1-1】 编写一个 C 语言程序,用其实现在屏幕上显示一句话:Hello World!。

```
/* 在屏幕显示一句话"Hello World!" */
#include "stdio.h"          /* 包含输入输出头文件 */
int main(int argc, char * argv[]) /* 主函数 */
{
    printf("Hello World! \n"); /* 利用 printf()函数输出"Hello World!" */
    return 0;
}
```

分析

(1) 在“/*”和“*/”之间的内容是注释信息,在程序编译、运行中不起作用。注释信息是为了方便阅读程序,它是由程序员加进去的,它必须以“/*”开头,以“*/”结束。

(2) printf();是C程序中的输出函数,用来显示结果:“Hello World!”,若程序中用到输入或输出函数,如 printf(),则需要在程序的首部用头文件进行说明,例如: #include "stdio.h"。

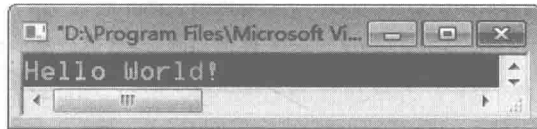
(3) main()为主函数说明,任何程序都有且只有一个 main()函数。

(4) 函数内容用大括号{}括起来。

(5) printf("Hello World!\n");的功能是在屏幕上显示"Hello World!",其中"\n"表示输出后换行。

(6) 每条语句的末尾必须添加“;”表示该语句结束。

程序运行结果:



【例 1-2】 编写一个 C 程序,用来实现求两个数的和。

```
/* 求两个整数相加的和 */
#include "stdio.h"                /* 包含输入输出头文件 */
int main(int argc,char * argv[])  /* 主函数 */
{
    int a,b,sum;                  /* 定义整数变量 a,b,sum */
    a=12;
    b=34;                          /* 给变量赋值 */
    sum=a+b;                        /* 利用表达式求 sum */
    printf("%d+%d=%d\n",a,b,sum); /* 输出 a,b,sum */
    return 0;
}
```

分析

(1) 程序中使用了 printf()函数,包含定义输入输出函数的头文件。

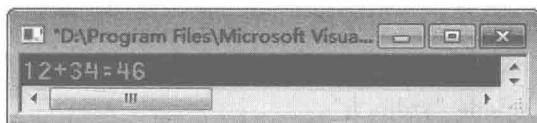
(2) 程序由唯一的 main()函数组成。

(3) main()函数的内容用大括号括起来。在 main()函数中包含:程序所使用变量的定义,变量赋值,利用表达式计算结果,输出程序运行结果。

(4) 语句结束时添加“;”。

(5) 该程序要完成的内容放在{}中表示。

程序运行结果:



【例 1-3】 编写程序,实现用户从键盘输入两个整数,输出其中较小的数。

```
/* 输入两个数,程序输出其中较小的数 */
#include "stdio.h"
```

```

int min(int a,int b)                                /* 辅函数 */
{
    int c;
    if(a>b)
        c=b;
    else
        c=a;
    return c;
}

int main(int argc,char * argv[])                  /* 主函数 */

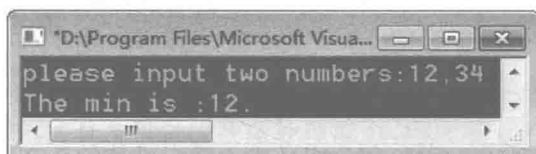
{
    int a,b,c;
    printf("please input two numbers:"); /* 输出提示性信息 */
    scanf("%d,%d",&a,&b);                /* 用户输入两个整数 */
    c=min(a,b);                                /* 调用辅函数,计算较小的数 */
    printf("The min is:%d.\n",c);          /* 输出较小的数 */
    return 0;
}

```

分析

- (1) 程序由 main() 函数和用户自定义函数 min(a,b) 组成。
- (2) 用户自定义函数 min(a,b) 用于求出两个数 a,b 中较小的数。
- (3) 用户自定义函数 min(a,b) 内容同样包含在大括号中,包括:定义存放较小数的变量 c,利用分支选择语句 if-else 求出 a,b 中较小的一个,再使用 return c 语句将 min(a,b) 函数的结果返回到主函数中。
- (4) 每一个语句结束时都添加“;”。

程序运行结果:



通过上述三个例题,我们可以看出 C 语言程序的基本结构是:

- (1) C 语言程序由函数构成,一个完整的 C 语言程序由一个主函数 main() 组成,或者由一个主函数 main() 加一个或多个辅函数组成。

C 语言的函数有三种形式:

① 主函数,main()。每个程序都有且仅有一个主函数。在程序执行时,任何一个程序都是从主函数开始执行起。

② C 语言提供的库函数(又称内部函数),例如 printf() 函数。如果在程序中使用了库函数,C 语言规定:在程序首部应该加入定义库函数的头文件。

例如,例题中的 #include "stdio.h"。

③ 用户自定义函数(又称辅函数),辅函数在程序中可有可无,若有也可以有多个,它由用户按需要实现的特定功能自己定义。

- (2) 程序中的函数的内容用大括号“{}”括起来。

(3) 每条语句的末尾必须添加“;”表示该语句结束。

1.4 Microsoft Visual C++ 6.0 集成开发环境的使用

1. 启动 Microsoft Visual C++ 6.0 开发环境

如果计算机在: Microsoft Windows® XP Professional 或 Microsoft Windows® 2000 Professional 以上版本的操作系统上已经安装好 Microsoft Visual C++ 6.0, 就可以启动 Microsoft Visual C++ 6.0 来实现程序开发。

(1) 启动 Microsoft Visual C++ 6.0

利用命令“开始|所有程序|Microsoft Visual C++ 6.0|Microsoft Visual C++ 6.0”启动 Microsoft Visual C++ 6.0, 或者双击桌面的 Microsoft Visual C++ 6.0 的快捷方式图标 。

(2) 启动 Microsoft Visual C++ 6.0 之后就会显示图 1-1 所示的界面。

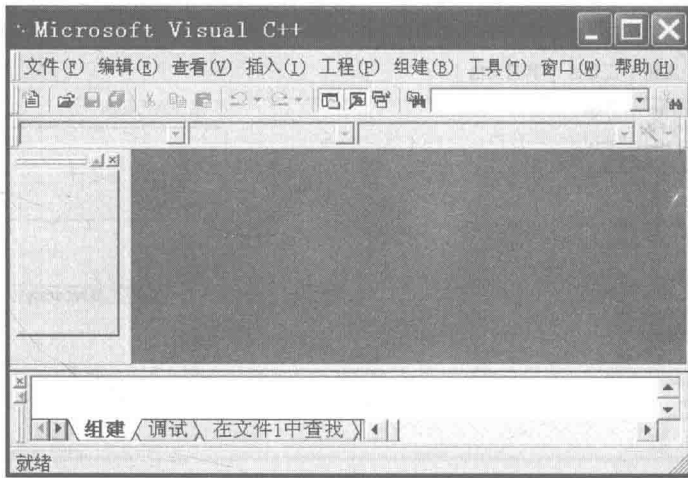


图 1-1 Microsoft Visual C++ 6.0 运行界面

(3) 在 Microsoft Visual C++ 6.0 运行界面利用命令“文件|新建”, 弹出如图 1-2 所示的“新建”对话框。

选择“文件”标签, 出现如图 1-3 所示新建|文件对话框。移动光标选择 C++ Source File 菜单项或 C/C++ Header File 菜单项, 在文件名框中输入文件名, 如输入 HELLO.C, (注意: 在输入文件名时, 扩展名.C 不能省略) 在位置窗口中确认新建文件存放的外存路径位置, 最后点击“确定”按钮, 此时打开了一个名为 HELLO.C 的编辑窗口, 如图 1-4、图 1-5 所示。

在编辑窗口下输入 C 程序。例如:

```
#include<stdio.h>
main()
{
printf("HELLO.C\n");
}
```



图 1-2 “新建”对话框

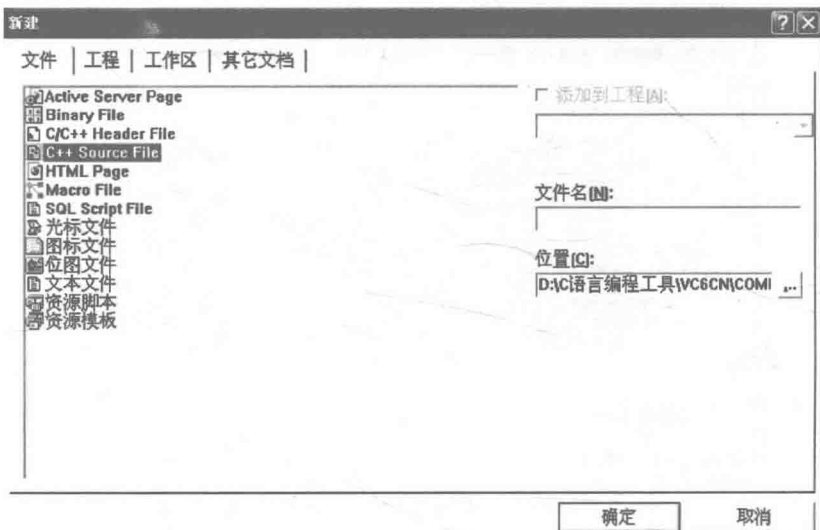


图 1-3 新建|文件对话窗口

(4) 利用命令“文件|保存全部”实现工程的保存。

(5) 利用命令“文件|退出”退出 Microsoft Visual C++ 6.0 开发环境。

2. 打开已经存在的 C 程序

① 启动 Microsoft Visual C++ 6.0 开发环境。

② 利用命令“文件打开”出现如图 1-6 所示的对话框。

在选定的查找范围内,找到需要打开的文件,单击打开按钮。

3. Microsoft Visual C++ 6.0 开发 C 程序的上机步骤

1) C 语言程序开发步骤

对于一个给定的问题,编好一个 C 程序后,上机运行步骤有:这里要经过输入与编辑源程序→源程序编译→与库函数连接→运行目标程序。如图 1-7 所示。

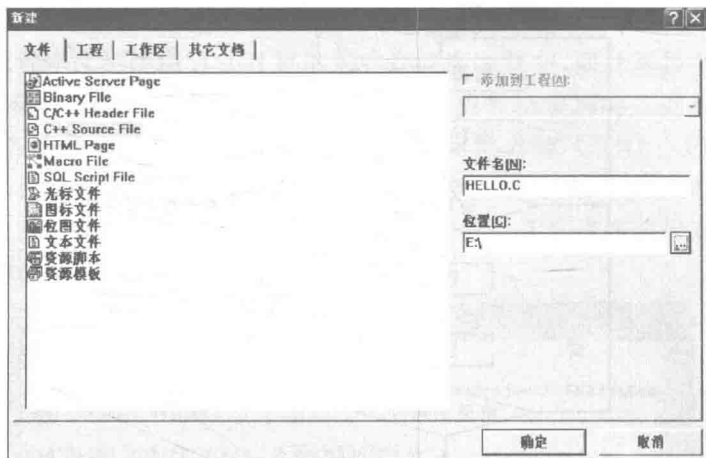


图 1-4 输入文件名和确认文件位置

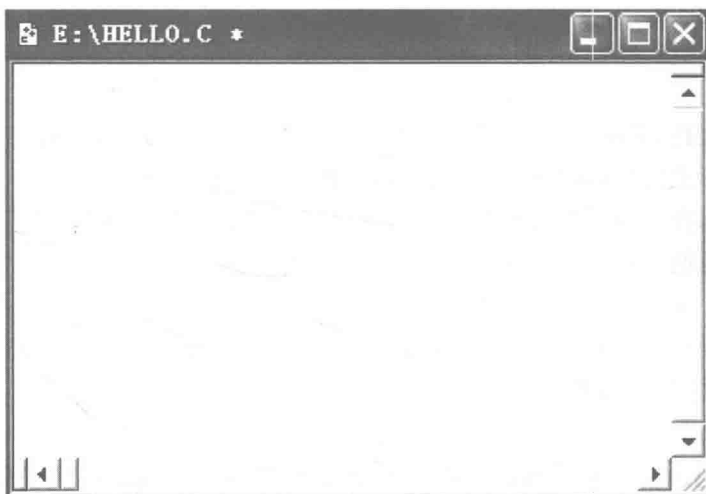


图 1-5 新建文件 HELLO.C 的编辑窗口



图 1-6 打开文件对话框

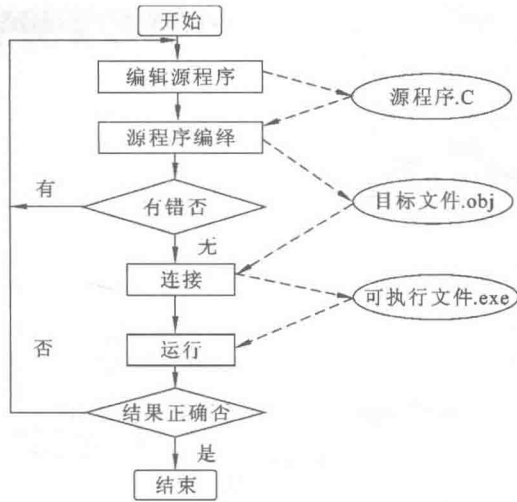


图 1-7 C 语言上机流程图

其中:编辑源程序是将写好的程序输入到计算机并保存到计算机外存中;或源程序有错需要进行修改。

源程序编译是指,采用“C 编译程序”软件,检查源程序的语法错误,若没有错误后,将源程序翻译成二进制形式的“目标程序”,生成的目标程序的扩展名为 .obj。

连接是将目标程序与系统的函数库和其他目标程序连接起来,形成可执行的目标程序,其生成的可执行文件的扩展名为 .exe。

2) Visual C++ 6.0 开发 C 程序的过程

利用 Microsoft Visual C++ 6.0 开发一个 C 程序,一般要经历编辑程序代码、程序编译和组建、程序运行等几个步骤。

(1) 编辑源程序。

用户通过 Microsoft Visual C++ 6.0 开发环境,将自己开发的 C 语言程序输入代码设计窗口,如图 1-8 所示。

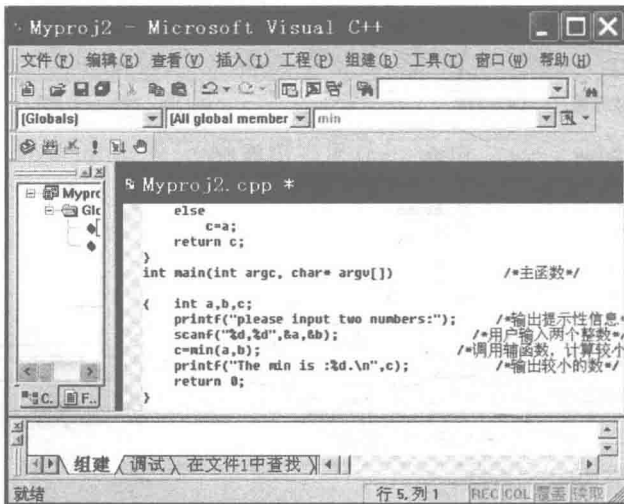


图 1-8 Microsoft Visual C++ 6.0 开发环境