



语言 程序设计

C YUYAN CHENGXU SHEJI

郭 瑞 主 编

学 习 练 习 设 计



C 语言程序设计

郭 瑞 主 编

雷玉明 副主编

云南大学出版社
样 书

云南大学出版社

图书在版编目 (CIP) 数据

C 语言程序设计/郭瑞主编. —昆明: 云南大学出版社,
2007

ISBN 978 - 7 - 81112 - 429 - 3

I. C… II. 郭… III. C 语言—程序设计—成人教育:
高等教育—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 147409 号

C 语言程序设计

郭 瑞 主编

策划编辑 鄢康贵
责任编辑 李兴和 史明舒
封面设计 丁群亚
出版发行 云南大学出版社
印 装 云南国浩印刷有限公司
开 本 787mm × 1092mm 1/16
印 张 16.75
字 数 413 千
版 次 2007 年 10 月第 1 版
印 次 2007 年 10 月第 1 次印刷
书 号 ISBN 978 - 7 - 81112 - 429 - 3
定 价 28.00 元

云南大学出版社地址: 云南大学英华园内

电话: 0871 - 5033244 邮编: 650091

网址: <http://www.ynup.com>

E-mail: market@ynup.com

前 言

自学是成人教育的重要环节之一,是保证教育质量的基础。本教材是大学专科类的配套使用书,适用于成人教育函授三年制理工类非计算机类各专业的学生自学使用。

C 语言是一门通用程序设计语言,它已成为计算机学科和其他应用学科的一门重要基础课。由于 C 语言在开发系统软件和应用软件中的广泛应用,课程的任务是通过本课程的学习,使学生掌握 C 语言程序设计的语法规则、相关的基本概念及常见的计算方法;培养使用 C 语言的编程能力及上机调试能力,为其他后续课程的学习,及从事计算机应用的各种开发打下坚实基础。为进一步培养和提高学生综合编程的能力奠定基础。

本书每一章分为本章知识要点、重点与难点,上机指导,典型例题,测验及作业题等。本教材的编写力求适合目前学校成人教育的教学情况。

本教材的特点:

(1) 针对 C 语言程序设计中的重点、难点做相应的注解、补充和强调,将其归纳、分析,实现学生对知识的更好掌握与理解。

(2) 习题难易程度适中,要适用于成人教育的特点,题中没有偏题和怪题,重点培养学生基础知识的掌握,并使学生举一反三,增强其实际应用能力。

(3) 本教材紧扣教学大纲,书中题型比较灵活,所选习题涵盖了 C 语言程序设计中的大部分知识点,既可以作为随堂练习,也可作为学生的自学作业,具有很高的实用价值,使学生在较多的时间内,尽快熟悉 C 语言的基本知识和编程技巧。

(4) 实践教学—上机指导部分在上机题目的设计上力求由浅入深,渐进增加难度,对学生容易出差错的地方都配有对应的练习题,还增加了一些比较有趣味性的上机程序,增加学生在上机操作时的兴趣,造就其解决问题的成就感。

编 者

2007 年 7 月

目 录

第一章 C 语言入门基础	(1)
1.1 C 程序的结构	(4)
1.1.1 概述	(4)
1.1.2 C 程序的构成	(5)
1.2 C 程序入门	(6)
1.3 C 程序的一些概念	(7)
1.3.1 运算符	(7)
1.3.2 表达式	(7)
1.3.3 常量	(8)
1.3.4 变量	(9)
1.3.5 标识符	(9)
1.3.6 变量地址	(10)
1.3.7 关键字、保留字	(10)
1.3.8 数据的输入和输出	(10)
第二章 C 的数据类型	(17)
2.1 C 语言的基本数据类型	(18)
2.2 整型数据	(19)
2.2.1 整型常量	(19)
2.2.2 整型变量	(20)
2.2.3 整型变量的使用	(20)
2.2.4 整型变量的分类及数据范围	(20)
2.3 实型数据	(22)
2.3.1 实型常量	(22)
2.3.2 实型变量	(23)
2.3.3 实型变量的使用	(23)
2.3.4 实型变量的分类及数据范围	(23)
2.3.5 实型数据的有效位	(23)
2.4 字符型数据	(24)
2.4.1 字符常量	(25)
2.4.2 转义字符	(25)
2.4.3 字符变量	(25)
2.4.4 字符数据在内存中的存储形式	(25)
2.4.5 字符串常量	(26)

2.4.6 不同类型数据的混合运算	(27)
第三章 运算符、表达式	(35)
3.1 C 语言的运算符	(36)
3.2 C 语言的表达式	(37)
3.2.1 表达式	(37)
3.2.2 算术运算符与算术表达式	(37)
3.2.3 关系运算符与关系表达式	(38)
3.2.4 逻辑运算符与逻辑表达式	(39)
3.2.5 赋值运算符与赋值表达式	(39)
3.2.6 自增、自减运算符和自增、自减表达式	(40)
3.2.7 条件运算符与条件表达式	(41)
3.2.8 “&”和“*”运算	(41)
3.2.9 逗号运算符和逗号表达式	(41)
3.2.10 sizeof () 运算符	(42)
3.2.11 强制类型转换运算符与强制类型表达式	(43)
3.2.12 运算符的优先级和结合性	(43)
第四章 C 程序语句、三种基本结构	(50)
4.1 C 语句概述	(52)
4.2 程序的三种基本结构	(53)
第五章 选择结构程序设计	(58)
5.1 if 语句	(59)
5.2 switch 语句	(60)
第六章 循环控制	(71)
6.1 概述	(73)
6.2 goto 语句和 if 语句构成循环	(73)
6.3 while 语句	(73)
6.4 do ~ while 语句	(75)
6.5 for 语句	(76)
6.6 break 语句和 continue 语句	(79)
6.6.1 break 语句	(79)
6.6.2 continue 语句	(79)
6.6.3 continue 语句和 break 语句的区别	(79)
第七章 数 组	(87)
7.1 一维数组	(89)
7.1.1 一维数组的定义	(89)

7.1.2 一维数组元素的引用	(89)
7.1.3 一维数组的初始化	(91)
7.2 二维数组	(92)
7.2.1 二维数组的定义	(92)
7.2.2 二维数组的引用	(92)
7.2.3 二维数组的初始化	(93)
7.3 字符数组	(94)
7.3.1 字符数组的定义	(94)
7.3.2 字符数组的初始化	(94)
7.3.3 字符数组的输入输出	(96)
7.3.4 字符串处理函数	(97)
第八章 函 数	(106)
8.1 概述	(108)
8.2 库函数	(108)
8.2.1 使用库函数	(108)
8.2.2 库函数的调用方法	(109)
8.3 自定义函数	(110)
8.3.1 无参函数的定义形式	(110)
8.3.2 有参函数定义的一般形式	(110)
8.4 函数的调用	(112)
8.5 函数的返回值	(113)
8.6 函数参数的传递方式	(114)
8.6.1 值传递	(114)
8.6.2 地址传递	(115)
8.7 函数的嵌套调用	(116)
8.8 函数的递归调用	(116)
8.9 变量的作用域及其存储类型	(117)
8.9.1 定义变量	(117)
8.9.2 局部变量及其作用域	(118)
8.9.3 全局变量及其存储类型	(119)
第九章 预处理命令	(127)
9.1 宏定义	(128)
9.1.1 无参数的宏定义	(128)
9.1.2 有参数的宏定义	(130)
9.2 “文件包含”处理	(132)
9.3 条件编译	(134)

第十章 指 针	(140)
10.1 地址和指针	(143)
10.1.1 地址	(143)
10.1.2 指针和指针变量	(144)
10.2 指针变量的定义	(145)
10.2.1 指针变量的定义	(145)
10.2.2 指针变量的基类型	(145)
10.2.3 指针与指针变量的区别	(145)
10.2.4 指针变量的赋值	(145)
10.3 指针变量的运算	(146)
10.3.1 指针变量的两个有关的运算符	(146)
10.3.2 指针变量的移动	(147)
10.3.3 指针变量间的其他运算	(147)
10.4 指针与数组	(150)
10.4.1 数组指针	(150)
10.4.2 数组的引用	(150)
10.4.3 用数组名作函数参数	(155)
10.5 字符串和字符指针变量	(157)
10.5.1 字符数组与字符串	(157)
10.5.2 字符指针	(158)
10.6 指针与函数	(160)
10.5.1 指向函数的指针	(160)
10.5.2 返回值为指针的函数	(164)
10.5.3 命令行参数	(166)
10.7 指向多维数组的指针和指针变量	(167)
10.7.1 多维数组的地址	(167)
第十一章 结构体与联合	(178)
11.1 概 述	(184)
11.2 结构体类型变量的定义	(185)
11.3 结构体变量的引用	(188)
11.4 结构体变量的初始化	(190)
11.5 结构体数组	(190)
11.5.1 定义结构体数组	(190)
11.5.2 结构体数组的初始化	(191)
11.6 指向结构体类型数据的指针	(192)
11.7 共用体	(199)
11.7.1 共用体定义、使用 and 初始化	(199)
11.8 枚举类型	(204)

11.8.1 枚举类型定义、使用和初始化	(204)
11.9 类型定义	(206)
11.10 单链表插入、删除、查找等操作	(207)
11.10.1 动态分配系统 malloc、calloc、realloc 和 free 函数	(207)
11.10.2 单链表的基本操作	(208)
第十二章 文 件	(218)
12.1 C 文件概述	(221)
12.1.1 文件的概念	(221)
12.1.2 从用户的角度看文件的分类	(221)
12.1.3 从文件编码的方式来看文件的分类	(222)
12.2 文件类型指针	(222)
12.2.1 文件的结构体类型变量	(222)
12.2.2 文件指针的定义	(223)
12.2.3 文件的打开、关闭	(223)
12.2.4 文件的顺序读写	(226)
12.3 位置指针与文件定位	(235)
12.3.1 位置指针复位函数 rewind ()	(235)
12.3.2 随机读写 fseek () 函数	(236)
12.4 文件检测函数	(237)
12.4.1 文件结束检测函数 feof ()	(237)
12.4.2 读写文件出错检测函数 ferror ()	(237)
12.4.3 文件出错标志和文件结束标志置 0 函数 clearerr ()	(238)
12.5 C 库文件	(238)
附录一 ASCII 码表	(242)
1. 标准码 (0 ~ 127)	(242)
2. 扩展码 (128 ~ 255)	(243)
附录二 运算符及其优先级汇总表	(244)
附录三 C 语言部分常用库函数	(245)
附录四 C 程序的一些常用算法	(251)
1. 两数交换	(251)
2. 大小写字母转换	(251)
3. 递归法转换整数为字符	(251)
4. 判断素数	(251)
5. 求闰年	(251)
6. 求 N 的阶乘	(252)

7. 求 m 和 n 最大公约数和最小公倍数	(252)
8. 求 3 个数由小到大的顺序输出	(252)
9. 求 $ax * x + b * x + c$ 方程的解	(252)
10. 求 Fibonacci 数列 ($f(n) = f(n-1) + f(n-2)$)	(252)
11. 迭代法求 $x = \text{sqrt}(a)$ (迭代公式: $x(n+1) = (x(n) + a/x(n)) / 2$ 绝对误差 < 0.00001)	(253)
12. 求方程 $2 * x * x * x - 4 * x * x + 3 * x - 6$ 的根	(253)
13. 冒泡法排序 (由小到大)	(254)
14. 选择法排序 (由小到大)	(255)
15. 插入法排序	(255)
16. 数列中查找一个关键字	(255)
17. 汉诺塔 (Hanoi) 问题	(256)

子目录中存放所有头文件。

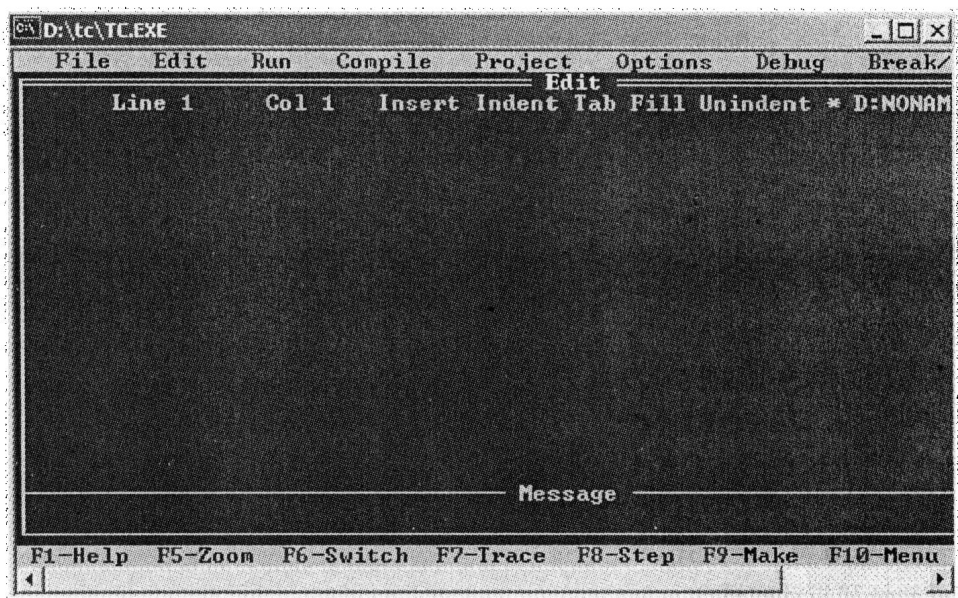


图 1.1 Turbo C2.0 窗口

3. 源程序的输入、编译和运行

用户用 C 语言编写的程序称为源程序，其文件后缀为 c，例如：grabc.c。计算机硬件不能直接执行源程序，必须将源程序编译、连接生成程序文件，其后缀为 exe，例如：grabc.exe。下面我们试着建立一个名为 grabc.c 的源程序。操作步骤如下：

- (1) 启动 Turbo C2.0（击文件 tc.exe），使系统进入 Turbo C 集成开发环境
- (2) 通过键盘输入源程序，例如：

```
main ()
{
printf (" Hello, world \ n");
}
```

- (3) 保存源文件。按 F2 键后输入文件名（后缀为 c，输入时可以省略）回车存盘。
- (4) 编译程序，方法有多种：

① 按 F9 编译、连接生成同名但后缀为 exe 的文件，称为程序文件。通常用于 DOS/Windows 98 操作系统中；

② 按 F10 键返回主菜单，选择 Compile 项，选择 Compile to .OBJ 项，按回车键。进入编译状态后，屏幕会出现一个编译窗口，几秒钟后，屏幕显示一闪烁信息：Success: press any key! 表示编译成功。此时可按任意键，编译窗口消失，光标返回主菜单。如果编译时产生警告 Warning 或出错 Error 信息，这些具体错误信息会显示在屏幕下部的信息窗中，必须纠正这些错误。按 Alt + E 对源程序进行修改，重新进行编译直至没有出错 Error 信息产生。

③ 按 Ctrl + F9 也可以编译、连接并运行程序，通常用于 Windows XP/2000 操作系统下。

(5) 运行程序

① 在 DOS 操作系统下，直接键入文件名后回车即可。

② 在 Windows95/98 下，模拟 DOS 状态下，直接键入程序文件名回车即可。或者，用鼠标双击程序文件名。

③ 在 Windows2000/XP 下，先按 Ctrl + F9，再按 Alt + F5 查看执行结果。

(二) Microsoft Visual C++ 6.0 操作方式

(1) 新建一个工程文件（后缀为 dsw 的文件）：启动 Microsoft Visual C++ 6.0，选择“文件”、“新建”，进入下一页，如图 1.2。

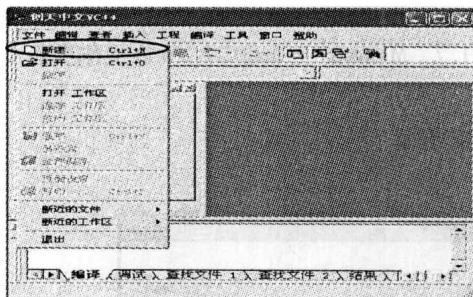


图 1.2 Microsoft Visual C++ 6.0 操作步骤 1

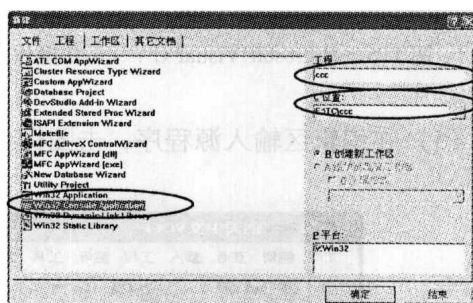


图 1.3 Microsoft Visual C++ 6.0 操作步骤 2

(2) 选择“工程”中的“win32 console Application”；在工程对话框中键入工程名：ccc，在“位置”中指明工程路径，击“确定”按钮进入下一步，如图 1.3。

(3) 选择第 1 项后击“完成”，进入下一步，如图 1.5，击“确定”。

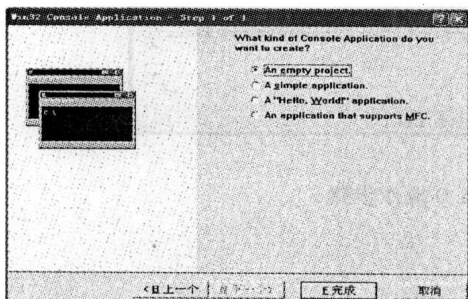


图 1.4 Microsoft Visual C++ 6.0 操作步骤 3

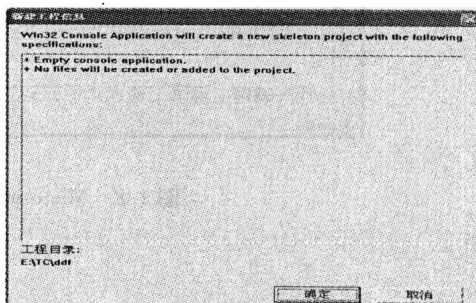


图 1.5 Microsoft Visual C++ 6.0 操作步骤 3

(4) 新建一个程序文件（后缀为 cpp 的文件）：再单击“文件”、“新建”，选择“文件”页、“C++ Source File”、在文件对话框中键入文件名：ccc1，击“确定”。

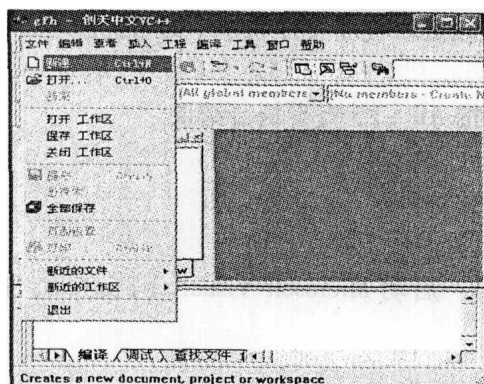


图 1.6 Microsoft Visual C++ 6.0 操作步骤 4

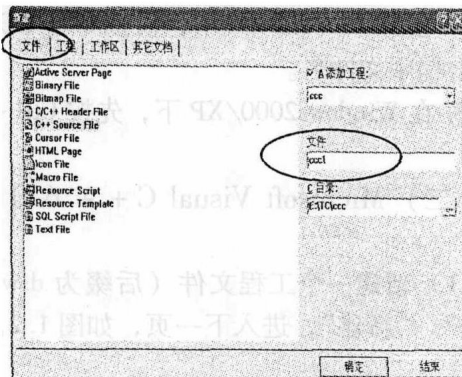


图 1.7 Microsoft Visual C++ 6.0 操作步骤 4

(5) 在编辑区输入源程序，击图标 “!” 或按 F5 键运行。



图 1.8 Microsoft Visual C++ 6.0 操作步骤 5

1.1 C 程序的结构

1.1.1 概述

C 语言是一种结构化语言，它层次清晰。实际上每个“层次”都可以理解为一个函数，函数又分成两类，即标准函数和用户自定义函数，标准函数又称库函数，由系统提供，用户可直接调用，如 `printf()`、`scanf()` 等，调用时要查阅教材函数附录。自定义函数由编程者自己编写，在每个函数中通常由两部分组成：数据的说明部分、算法部分。

(1) 库函数：编程者可以直接调用，调用时使用 `#include <文件名>` 格式进行文件包含。

(2) 自定义函数：用 C 语言编写的函数结构如图 1.9 所示。

文件预处理	
类型 函数名 (类型 参数 1, 类型 参数 2.....)	
函数体	数据声明部分;
	语句部分;

图 1.9 自定义函数结构

1.1.2 C 程序的构成

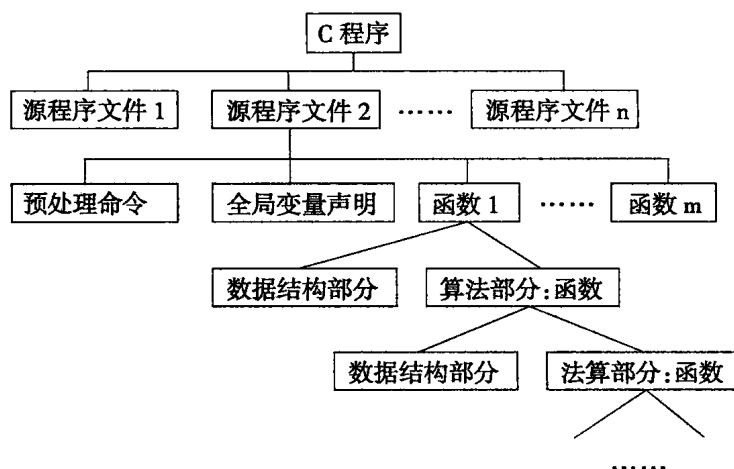


图 1.10 C 程序结构

C 程序是由若干个函数构成，一个完整的 C 程序要包含一个主函数 `main()`，且只能有一个主函数 `main()`，程序运行时从 `main()` 函数开始，最后回到 `main()` 函数，即在主函数中可以调用其他函数，其他函数之间也可以相互调用，但其他函数不能调用主函数。

【例 1.1】 一个完整的 C 程序结构的例子，求半径 $r=12.5$ 的圆的面积。

```

#include "stdio.h" /* 本句也可写成#include <stdio.h> */
float s (float x)
{
    return X*X*3.14
}
main ()
{
    float r, area, s (float x);
    r = 12.5;
    area = s (r);
    printf ("%f\n", area);
}
    
```

定义函数 `s(x)`

主函数 `main()`，调用了函数 `s(x)`，把 `r` 值“代入”求出 `s(r)` 后赋给变量 `area` 通过库函数 `printf()` 输出它。

【例 1.2】 本例只有一个主函数。输出：Turbo C2.0。

```
#include "stdio.h"      /* 文件包含，表示在主函数中使用系统输出或输入函数 */
main ()                /* 主函数，注意其后没有分号 */
{
    printf (" Turbo C2.0"); /* printf ( ) 为系统输出函数 */
}
```



说明

- (1) 在 C 程序中，每句结束必须有一个分号“;”；
- (2) 注意 main ()、#include <stdio.h>、#define PI 3.1415 后没有分号，因为它们不是语句，分别称为主函数、文件包含、预处理命令；
- (3) 在程序中可以使用一对 /* 和 */ 对部分内容进行注释，注释的目的主要是为了让读者读懂程序，它并不运行，可以用它来调试程序；
- (4) 主函数 main () 或自定义函数“内容”必须用一对大花括 { } 括起来；通常情况下，大花括号里包括两部分：数据结构部分（变量或函数说明部分）、语句部分；
- (5) 在 C 程序中，大小写是有区别的，如 printf 和 PRINTF 是两个不同的符号；通常情况 C 程序都是小写字母，只有用户自定义符号常量时可以使用大写字母；
- (6) 程序中必须有且只能有一个 main () 函数。且只从主函数开始运行到主函数结束；
- (7) 在书写 C 程序时，一行可以有多句，建议一句占一行；
- (8) 在 C 程序中，若用到变量一定要先说明其数据类型，后使用。

最常用的三种类型说明符：

int 整型类型说明、float 或 double 为实型数据类型说明符、char 为字符类型说明符是。

1.2 C 程序入门

一个完整的 C 程序由若干个函数或一个主函数 main () 组成。下面看一些简单的 C 程序。

【例 1.3】 最简单的 C 程序，只有一个主函数，没有用户自定义函数。

```
#include <stdio.h>
main ()
{
}
```

分析：程序只由一个主函数 main () 组成，在主函数中什么都没有，什么都不做。#include <stdio.h> 在 Turbo C2.0 中可以省略，在 Microsoft C++ 6.0 中不能省略。

【例 1.4】 在屏幕上输出：“欢迎大家学习 C 语言!”。

```
#include <stdio.h>
```

```

main ()
{
    printf ("欢迎大家学习 C 语言! \n"); /* 函数 printf () 后有分号, "\n" 表示换行
*/
}

```

分析: printf () 为库函数, 在文件 stdio. h 中定义了它的算法, 故在主函数 main () 中调用时应主函数 main () 前使用 #include <stdio. h> 或 #include "stdio. h", 称为“文件包含”。此句在 Turbo C2.0 中可以省略, 但在 Microsoft visual C++ 6.0 中不能省略!

【例 1.5】 求任意两数之和。

```

#include "stdio. h"
main ()
{ float a, b, sum;
  float add (float a, float b);
  scanf ("%f,%f", &a, &b);
  sum = add (a, b);
  printf ("%f\n", sum);
}

float add (float x, float y)
{
  return x + y;
}

```

分析: 此程序是由两个函数构成, 即主函数 main () 和自定义求和函数 add (), 程序只运行主函数 main (), 在主函数中调用了 add () 函数, 调用时应说明其数据类型 float add (float a, float b);。

1.3 C 程序的一些概念

1.3.1 运算符

运算符: 是指 C 程序能够进行运算的符号。如: +、* 等。C 语言的运算符范围很宽, C 程序中共有 34 种运算符。主要有以下几类:

1. 算术运算符 2. 关系运算符 3. 逻辑运算符 4. 位运算符 5. 赋值运算符 6. 条件运算符 7. 逗号运算符 8. 指针运算符 9. 求字节数运算符 10. 强制类型转换运算符 11. 分量运算符 12. 下标运算符 13. 其他运算符 (如函数调用运算符 ())

1.3.2 表达式

表达式: 用相应的运算符连接起来的式子就称相应的表达式。如用算术运算符连接起来的式子 $5+6$ 就称算术表达式, 用关系运算符连接起来的式子 $8>6$ 就称为关系表达式。

在 C 程序中, 任何一个表达式都“规定”有一个值, 如表达式 $5+6$ 的值为 11, 而 $5<6$ 的值为 1。下面介绍一些常用的运算符及其表达式。

1. 算术表达式:

- ① $5 * 6$ 的值为: 30
 ② $5/2$ 的值为: 2 (两个整数进行/运算时其值为整型数据)
 ③ $5.0/2$ 的值为: 2.500000 (当实型和整型数据运算时, 计算机自动把整型转换成实型)

④ $6\%2$ 的值为: 0 (% 运算只能在两个整数之间运算, 表达式 $6.6\%3$ 是语法错误)

2. 关系表达式:

在 C 程序中, 若关系表达式成立, 则其值为 1, 否则为 0。如:

- ① $5 > 6$ 的值 0
 ② $5 != 6$ 的值为 1 (! = 为双符号用来表示“不等于”)
 ③ $5 < = 5$ 的值为 1
 ④ $5 = = 6$ 的值为 0 (= = 为双符号, 表示“等于”, 注意与 $5 = 6$ 区别, $5 = 6$ 为语法错误)

3. 逻辑表达式

在 C 程序中, && 称为“与”运算, || 称为“或”运算, ! 称为“非”(取反)。逻辑表达式有下列三种形式: $a \&\&b$ 、 $a || b$ 、 $! a$ 。逻辑表达式的值根据表 1.1 真值表来判别。

表 1.1 逻辑运算真值判别

a	b	$a \&\&b$	$a b$	$! a$
非 0	非 0	1	1	0
0	非 0	0	1	1
非 0	0	0	1	0
0	0	0	0	1

如:

- ① $5 \&\&0$ 的值为: 0
 ② $5 || 0$ 的值为: 1
 ③ $! 5$ 的值为: 0 (不能写成 $5!$)

1.3.3 常量

常量: 在程序运行中, 其值不能被改变的量称为常量。在 C 程序中常量有不同的类型, 常用的有: 整型、实型和字符型。下列为几种不同数据类型常量的表达方式:

1. 整型常量:

- ① 1200、-45、0 等为十进制整型常量;
 ② 0236、-0112 等为八进制整型常量 (以 0 开头的数);
 ③ 0x688、0xff、0X62 等均是十六进制整型常量 (以 0x 或 0X 开头的数)。

2. 实型常量:

- ① 3.1415926、-0.0023 等小数形式的实型常量表达方式;
 ② $1256e-2$ 、 $1.5E3$ 等为指数形式的实型常量表达方式, 分别相当于: 1256×10^{-2} 、 1.5×10^3 。