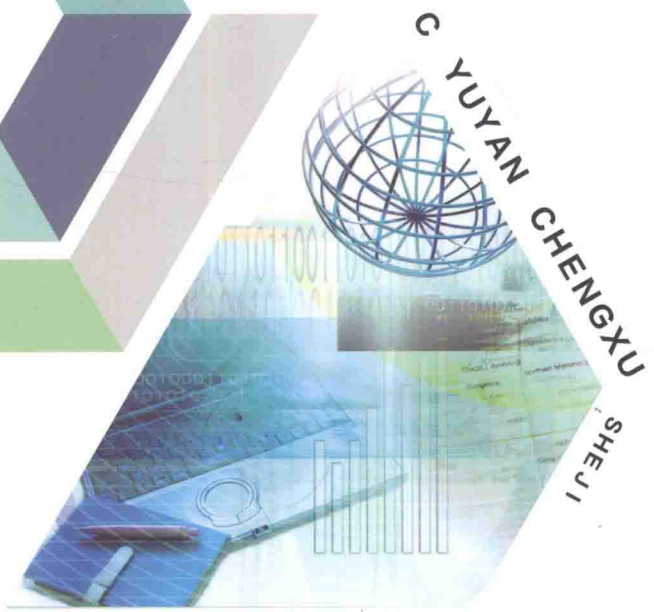
普通高等院校计算机基础教育规划教材 · 精品系列

C语言程序设计

（第四版）

罗 坚 徐文胜 主 编
傅清平 李雪斌 副主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

普通高等院校计算机基础教育规划教材·精品系列

C语言程序设计

（第四版）

罗　坚　徐文胜　主　编
傅清平　李雪斌　副主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书以程序设计为主线，以编程应用为驱动，理论联系实际，通过丰富的实例分析详细地介绍了C程序设计的思想及方法。全书叙述严谨、实例丰富、由浅及深、重点突出。

全书共8章，内容包括C程序设计入门，数据类型、运算符和表达式，算法与程序设计基础，函数，数组类型与指针类型，结构类型与联合类型，文件，面向对象技术与C++，为读者在学完C语言之后向C++过渡提供了帮助。为避免在学习过程中枯燥乏味，书中精选了一些实用性强、趣味性足的实例，增强了全书的可读性和学生的参与性，便于学生在轻松愉快的气氛中学习。

本书适合作为高等院校各专业C语言程序设计课程的教材，也可作为广大编程爱好者的自学读物，还可作为各类计算机等级考试的辅导书。

图书在版编目（CIP）数据

C语言程序设计/罗坚，徐文胜主编. —4版.
—北京：中国铁道出版社，2016.2
普通高等院校计算机基础教育规划教材. 精品系列
ISBN 978-7-113-21365-7

Ⅰ. ①C… Ⅱ. ①罗… ②徐… Ⅲ. ①C语言—程序设计—高等学校—教材 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2016)第027722号

书　　名：C语言程序设计（第四版）
作　　者：罗　坚　徐文胜　主编

策　　划：刘丽丽　　**读者热线：**（010）63550836
责任编辑：周　欣　曹莉群　冯彩茹
封面设计：一克米工作室
责任校对：汤淑梅
责任印制：郭向伟

出版发行：中国铁道出版社（100054，北京市西城区右安门西街8号）
网　　址：http://www.51eds.com
印　　刷：北京尚品荣华印刷有限公司
版　　次：2003年1月第1版　2005年12月第2版　2009年2月第3版
2016年2月第4版　2016年2月第1次印刷
开　　本：787 mm×1 092 mm　1/16　**印张：**20.25　**字数：**490千
印　　数：1～3 000册
书　　号：ISBN 978-7-113-21365-7
定　　价：45.00元

前言（第四版）

C 语言历史悠久，是一种被广泛使用的计算机高级程序设计语言。

它以精练、灵活、可移植性好、应用领域广泛而著称，已经历经 40 多个春秋，但至今依旧活跃在计算机应用以及计算机专业教学领域，彰显出无穷的编程魅力和蓬勃的生命力，国内外许多高校都将 C 语言列为学习程序设计课程的首选语言。

C 语言程序设计课程特色鲜明，以编程语言为平台，详细介绍程序设计的思想及方法。本书以培养学生掌握程序设计的方法与技能为重点，通过对本课程的学习，学生不仅能掌握程序设计语言的知识，为后续专业课程的学习打下基础，更重要的是能在实践中逐步培养求解问题和应用语言的能力。

本书作者长期从事高校 C 语言课程的一线教学，教龄都在 20 年以上，他们爱岗敬业，乐意为祖国建设培养青年才俊，力争在平凡的教学岗位做出不平凡的业绩。在漫长的 C 语言课程的教学过程中，作者亲身感受到了学生在初学编程时所遇到的种种不易，既为学生能够始终保持着强烈的学习兴趣和饱满的学习热情而感到高兴，也会因为学生遇到学习困难而主动迎难而上，为他们答疑解惑、排忧解难。从 2002 年开始撰写第一版的《C 语言程序设计》（含配套实验教材），一直到第四版教材的修订出版，我们始终都在努力着。为了更好地帮助学生领悟到程序设计的奥妙，适应新形势下创新人才培养模式的需要，我们有责任把教材写得更好、更精彩，这正是编者殚精竭虑之所在。

在前后 14 年的写书及教学过程中，我们编写的《C 程序设计教程》（含实验教程）获得了江西省普通高等学校优秀教材一等奖，修订的《C 语言程序设计（第三版）》（含实验教材）成功入选普通高等教育“十一五”国家级规划教材。

以本套书作为主讲教材、由王昌晶博士和罗坚共同主持，本书其他作者作为主要成员参与的“C 语言程序设计”课程研究，被评为“2015 年度江西省高等学校（本科）省级精品资源共享课”和江西师范大学的优质精品共享课，相关的课程网站（http://ntp.jxnu.edu.cn/G2S/site/preview#/home/v?currentoc=926）正在积极地完善中，内含教学通知、师资队伍、课程介绍、教学资源、教学视频、科研课题、教学成果和文献资源等版块。写书的成功和教学研究上的长进，永远是鞭策我们前进的动力。

全书以程序设计为主线，以编程应用为驱动，以丰富的实例详细介绍了 C 语言程序设计的思想及方法。全书共 8 章，第 1 章介绍了 C 程序的基本构成与 Visual C++ 6.0 的使用；第 2 章介绍了基本数据类型、运算符和表达式、基本输入与输出操作；第 3 章介绍了算法的概念、结构化程序设计中的三种控制结构（即顺序、选择和循环）；第 4 章介绍了自定义函数、变量的存储类型；第 5 章介绍了数组类型、指针类型的定义与使用，以及字符串

函数、动态内存分配与动态数组的应用；第6章介绍了结构类型和联合类型的使用，涉及常用链表的定义及操作；第7章介绍了数据文件的类型与操作方法；第8章介绍了面向对象程序设计的思想和C++的基本语法知识，供有较高要求的读者学习。

本书第四版延续了前三版教材叙述严谨、循序渐进、突出实践、方便自主学习的特点，并在以下几方面进行了修订：

（1）删除了有关TurboC 2.0上机环境的介绍，重写并完善了Visual C++ 6.0集成开发环境IDE的使用，由浅入深地介绍了上机调试程序的完整过程，并列举了在调试中可能出现的问题及其应对方案。

（2）受主教材篇幅的限制，把介绍另一种上机环境Code::Blocks（简称CB）的内容移至配套的实验教材上进行介绍。

（3）考虑到全国高校不同专业的教学特点及实际需要，对原书的部分章节进行了重写或者调整，使得新版书内容更加紧凑，实例更具有针对性，结构更加合理。

（4）修正了上一版教材中的错误，保证书中的所有源程序均能在Visual C++ 6.0环境下运行通过。

（5）为培养学生对编程的兴趣和爱好，书中继续补充了一些实用性强、趣味性足的实例，并给出了详细的分析过程。

（6）调整了各章课后的习题，删减了部分实用性弱的题目，补充了一些技巧性强的练习。

（7）继续为读者参加全国计算机等级考试二级（C语言）及其他同等级别的考试提供帮助。

本书由罗坚、徐文胜任主编，傅清平、李雪斌任副主编，其中第1章、第3章和附录A由傅清平编写，第2章、第4章和附录B由李雪斌编写，第5章和第6章由徐文胜编写，第7章、第8章和附录C由罗坚编写。全书由罗坚审核、修改及定稿。

为配合教学，我们还编写了配套的实验指导用书《C语言程序设计实验教程（第二版）》（中国铁道出版社出版）。书中除了为主教材《C语言程序设计（第四版）》提供全部的习题解答之外，还根据教学进度设计了同步的上机实验，并提供了五套模拟测试题，方便读者进行自测。

在本套书的编写过程中，江西师范大学计算机信息工程学院的老师们给予了很大的支持，对本书提出了宝贵的意见，在此表示感谢！中国铁道出版社的领导及编辑为本书的校审出版提供了无私的帮助，一并表示感谢！此外，在编写过程中还参考了大量的文献资料，在此谨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，书中难免存在疏漏和不足之处，恳请各位专家、读者批评指正。

编　者

2015年12月于江西师范大学

目　录

第1章 C语言程序设计入门

对于把C语言作为第一门编程语言的读者来说，最关心的问题是如何才能尽快地学会用C语言编程。要做到这一点，首先必须熟悉C语言的基本语法规则，对照已学过的例题进行模仿，开始简单的编程；然后再经历反复编写程序、调试程序和运行程序这些必需的环节，才能逐步领会和掌握相关的算法设计与编程技巧，进而达到用计算机编程解决实际问题的目的。

其实，学习编程的过程与学习英语的过程很相似。在英语学习中，通常都是从字母、音标、单词、短语这些基本的语法单元开始，进而掌握句型、翻译与写作。在学习C语言时，也需要从数据类型、常量与变量、运算符与表达式这些最基本的C语言语法知识开始，才能逐步了解算法与程序的基本结构，进而掌握函数、数组、指针、结构（结构体）、联合（共用体）、数据文件等复杂的语法知识。这一系列的过程，实际上就是一个不断学习、不断积累，编程实践的过程。

本章从几个简单的C语言实例开始，由浅入深地介绍C语言基于函数的程序结构，涉及常量与变量、算术运算、选择结构、循环结构、基本输入/输出函数等基本语法，帮助读者了解C语言程序的基本框架和书写格式，让读者体会到初次编程的快乐。至于本章例题中所涉及的C语法规则，读者初次接触时可不必深究，在后续章节的学习过程中都将学到。

一个写在纸上的C语言源程序是否正确，只能通过上机环节来验证。本章也将介绍Visual C++ 6.0 IDE集成开发环境的使用（注：集成开发环境，Integrated Development Environment），详细介绍C程序上机的完整过程，即C语言源程序经过编辑、编译、连接、调试等环节，直至最终程序能够正确运行。

1.1 引　例

【例1.1】要求在命令提示符窗口中显示“Hello, world!”这一行文字。

```
/*第一个 C 语言程序举例*/
/*包含有关标准库的信息*/
#include<stdio.h>
/*定义名为 main 的函数，它不接收实参值*/
void  main()
{
    /*main()的语句括在花括号中*/
    /*main()调用库函数 printf()，将在命令提示符窗口中显示 Hello,world! */
    printf("Hello,world! \n");
}
```

程序分析：

（1）注释。程序代码中位于“/*”与“*/”之间的字符序列称为注释，用于解释程序（或

者语句）。被注释的内容可以是一行文字或者连续的多行文字，使用注释能增强程序的可阅读性，使程序易于理解，便于程序员之间进行交流。注释可以在程序中自由地使用，但在程序编译时被自动忽略。读者应重视使用注释，养成良好的编程习惯。

（2）main()函数。函数是 C 语言的基本组成单位，任何一个 C 程序，不论大小，都是由一个或多个函数组成的。函数是一个单独的程序模块，完成指定的功能。本例中用到了两个函数，其中一个是名字为 main 的函数，另一个是 printf()函数。一般可以给函数任意命名，但 main 是一个特殊的函数名，一个 C 程序不论由多少个文件组成，有且只能有一个 main()函数，通常称为主函数，任何一个 C 程序都从它开始运行。main()函数常常还要调用其他的函数来协助其完成某些工作，被调用的函数有些是由程序员自己编写的，有些则由系统函数库提供。本例中的另一个函数就是由系统函数库提供的名为 printf()的函数。函数中的语句用一对花括号“{}”括起来。本例中的 main()函数只包含一条语句：printf("Hello,world!\n ");，语句最后有一个分号，表示语句的结束。

（3）函数的调用。当要调用一个函数时，先要给出这个函数的名称，然后考虑与被调函数的数据交换。在函数之间进行数据交换的一种方法是让调用函数向被调用函数提供一串称为实参（又称变元）的值。函数名后面的一对圆括号用于把这一串实参（实参表）括起来。在本例中，main()函数不要求任何实参，故用空实参表示。main()函数用“Hello,world!\n”作为实参来调用 printf()函数。

（4）printf()函数是一个格式化输出库函数，在本例中，它用于在命令提示符窗口照原样显示双引号内的字符序列“Hello,world!\n”（不包括双引号）。用双引号括住的字符序列称为字符串。本例中仅使用字符串作为 printf()函数的实参。

（5）显示内容中的字符序列'\n'是一个换行符，用于控制从下一行的最左边位置开始显示其后续的字符。'\n'只表示一个字符，注意不能把它看成“\”和“n”这两个符号。

具有这种特征的字符称为转义字符，除'\n'之外，还有表示制表符的'\t'、表示退格符的'\b'、表示双引号的'\"'、表示反斜杠符本身的'\\'，在第 2 章中会详细介绍。

（6）文件包含命令#include<stdio.h>。这里的“#include”称为文件包含命令，其意义是把尖括号（< >）内指定的文件包含到本程序中，成为本程序的一部分。被包含的文件通常是由系统提供的，其扩展名为“.h”，常称为头文件或首部文件。如果使用了系统提供的库函数，一般应在文件的开始用 #include 命令，将被调用的库函数信息包含到本文件中。本例中使用#include<stdio.h>是因为调用了标准输入/输出库中的 printf()函数。

需要说明的是，C 语言规定：如果一个程序只使用了 scanf()或 printf()这样的库函数，此时对头文件的包含命令可以省略不写，因此本例中也可以删去#include<stdio.h>这一行。有关函数使用的详细内容将在第 4 章中介绍。

【例 1.2】从键盘上输入一个圆的半径，要求计算圆的面积。

```
#include<stdio.h>
void main()
{
   double r;                    /*定义双精度实型变量 r，代表半径 */
   double s;                    /*定义双精度实型变量 s，代表面积 */
   printf("radius=? ");         /*屏幕提示输入半径的值*/
   scanf("%lf", &r);            /*从键盘上输入半径，将值存入变量 r 中*/
   s=3.1415926*r*r;             /*将面积的值计算出来，保存在变量 s 中*/
```

```
    printf("Area=%f\n",s);      /*按双精度实数格式显示计算结果*/
}
```

程序分析：

（1）变量说明。本例的主函数体中包含两部分：一是说明部分，二是执行部分。在说明部分定义了在函数中所用到的变量名称与类型，譬如本例中的变量 r 与 s，通常写在函数的开始处，且要求放在第一条可执行语句之前。例 1.1 没有使用任何变量，因此程序中就没有说明部分。C 语言规定，程序中所有用到的变量都必须先定义（有时也称说明）后使用，否则将会出错。说明语句由一个类型名与若干所要说明的变量名组成，double r;语句中的 double 是类型名，r 是变量名。double 类型表示所列变量为双精度实型变量，除它之外，C 语言还提供了其他一些基本数据类型，包括 char（字符型）、short（短整型）、int（整型）、long（长整型）和 float（单精度实数）等。另外，还有由这些基本类型构成的数组类型、结构类型、联合类型，以及指向这些类型的指针类型和返回这些类型的函数，这些内容将在后续章节中介绍。

（2）键盘输入函数 scanf()。本程序中的 scanf()函数的作用是从键盘上输入一个双精度实数给对应的双精度实型变量 r，此处的格式控制符%lf 表示按双精度实数输入数据。&r 的含义是将要接收数据的那个存储单元的地址，也就是变量 r 的存储地址，书写时注意不能漏写“&”，其含义将在第 2 章中详细介绍。

（3）赋值表达式。与 BASIC、Pascal、Fortran 等计算机高级程序设计语言一样，C 语言中数学值的计算要通过赋值表达式来实现。在 C 语言中，“=”称为赋值号，含义不同于数学中的等号，而是将其右边的表达式的值赋给左边的变量。

（4）算术表达式。编程时需把传统的数学式子转换为 C 语言中等价的表达式。C 语言中的算术运算符包括加法（+）、减法（-）、乘法（*）、除法（/）、求模（或求余）运算符（%），本例中赋值号右边是根据圆的半径求面积的公式，就是把数学表达式转换成与之等价的 C 语言表达式，更详细的内容将在第 2 章中介绍。

（5）格式输出：本例使用了 printf()函数的更多功能。printf()函数是一个通用格式化输出函数，将在第 2 章详细介绍。本例中 printf()函数具有两个变元，第一个变元是要打印的格式控制串%f，它与双引号后输出列表中的第二个变元 s 进行替换。

本例程序运行时，首先在命令提示符窗口中根据屏幕的提示信息，输入半径值 2，最终的圆的面积就显示在屏幕上。

```
radius=? 2↙
Area=12.566370
```

【例 1.3】从键盘上输入一个正整数 m，计算 1+2+3+…+m 的值。

本例中的正整数 m 是一个未知数，由例 1.2 可知，m 的数值可通过输入函数 scanf()得到，但从 1 加到 m 这个式子，显然不可能逐一把 m 个整数全部列举出来。如果 m 的值比较大，如 100，逐一列出的写法肯定是行不通的。这里可以巧妙地通过循环结构来解决这个求和问题。

```
#include<stdio.h>
void main()
{
  int i,sum,m;          /*定义三个整型变量，其中 i 为循环控制变量，m 用于接收正整数*/
  sum=0;                /*给和变量 sum 赋初值 0*/
  printf("Enter a positive integer: ");
```

```
    scanf("%d",&m);          /*根据屏幕提示输入 m 的值，运行程序时要求输入正整数*/
    for(i=1;i<=m;i++)
      sum=sum+i;             /*通过 for 循环进行累加求和*/
    printf("The sum is %d\n",sum); /*输出最后的结果*/
}
```

本例程序运行时，首先在命令提示符窗口中根据屏幕的提示信息，输入正整数 100，最终的运行结果就显示在屏幕上。

```
Enter a positive integer: 100↙
The sum is 5050
```

程序分析：

（1）键盘输入函数 scanf()：本程序中调用 scanf()函数的目的是从键盘上输入一个整型数给整型变量 m。

（2）for 循环语句：本例的特点是累加计算，所以可以采用循环语句（即重复计算）来实现。for 是实现循环结构的语句之一，它后面的圆括号内共包含三部分，它们之间用分号隔开。第一部分 i=1 是初始化部分，仅在进入循环前执行一次。第二部分是用于控制循环的条件测试部分，对 i<=m 这个循环判断条件要进行求值，如果所求得的值为真，就执行循环体（本例循环体中只包含一条语句 sum=sum+i;）。然后，再执行第三部分 i++（即 i=i+1），步长加 1，并再次判断此时 i<=m 的值；一旦求得的条件值为假，就终止循环的执行。for 循环语句的循环体可以是单条语句，也可以是包含在花括号内的一组语句。有关循环结构的用法将在第 3 章中介绍。

（3）条件测试：本例中用于控制循环执行次数的条件测试（即 i<=m），在 C 语言中称为关系表达式。如果 i 小于等于 m，其结果是“真”（true），在 C 语言中真值用 1 表示，否则是“假”（false），假值用 0 表示。除小于等于（<=）外，还有大于（>）、小于（<）、大于等于（>=）、等于（==）和不等于（!=）等关系运算符。数学里的 a=3 表示 a 与 3 的值相等，在 C 语言中应写为 a==3；数学里的 $b\neq10$ 则应写为 b!=10。

【例 1.4】采用 C 语言中自定义函数的结构，按照例 1.3 的要求重新编程，实际上是例 1.3 的第二种写法。

```
#include<stdio.h>
int func(int x);          /*对程序中用到的自定义函数 func()进行函数声明*/
void main()               /*主函数*/
{
   int m,sum;             /*说明部分，定义变量 m 和 sum*/
   printf("Enter a positive integer: ");
   scanf("%d",&m);        /*提示用户输入一个整数，它表示求和的项数值*/
   sum=func(m);           /*调用函数 func()求累加和，并将调用之后的结果赋给 sum*/
   printf("The sum is %d\n",sum);      /*显示结果*/
}
/*定义函数 func()，函数返回值为整型，形式参数 x 为整型*/
int func(int x)
{
    /*func() 函数中的说明部分，定义本函数中用到的变量 i 和 s*/
    int i;                /*变量 i 为循环控制变量*/
    int s=0;
    for(i=1;i<=x; i++)s=s+i;          /*利用 for 循环求 1 加到 x 的累加和*/
```

```
    return s;          /*返回s的值，通过func()带回调用处*/
}
```

程序运行结果如下：

```
Enter a positive integer: 100↙
The sum is 5050
```

程序分析：

（1）由功能划分确定函数划分：本程序进一步表达了C语言基于函数的基本结构。本例中程序的执行过程是，首先在屏幕上显示提示字符串，请用户输入一个整数，按【Enter】键后计算出累加和并在屏幕上显示。程序需要处理三件事情：从键盘接收输入的m；调用函数func()计算累加和（其功能是接收main()传递的m，计算出1加2，加3，一直加到m的和，并把它返回给main()）；输出计算结果。所以程序除一个主函数main()外，还包含计算累加和的函数func()、处理键盘输入的函数scanf()和处理输出的函数printf()，输入函数和输出函数都是系统库函数，由C编译系统库提供，用户只需按规定使用而不必自己编写，但累加和计算函数func()是由用户自己编写的自定义函数。

（2）自定义函数func()。用户自定义函数的一般定义形式为：

```
函数返回值类型  函数名(形参类型 形参1,形参类型 形参2,…)   /*函数首部*/
{                                                    /*函数体*/
    函数的说明部分
    函数的执行部分
}
```

对照上述的定义格式，自定义函数func()的形式为：

```
int func(int x)                                      /*函数首部*/
{                                                    /*函数体*/
    …
}
```

自定义函数必须先定义后使用。在定义一个自定义函数时，函数名的后面必须跟上一对圆括号，这是函数结构的特有标志。圆括号中的形参可以是一个或多个，也可以一个都没有，当没有形参时，此时的圆括号也必须加上，不能省略不写。

函数体一般包括说明部分和执行部分。

说明部分：在此定义函数内部所要使用到的变量。例如：

```
int i;
int sum=0;
```

另外，对被调用的函数也需要进行说明。

执行部分：由若干条语句组成。例如：

```
int func(int x);
for(i=1;i<=m;i++)sum=sum+i;
return sum;
```

func()函数计算得到的值由return语句返回给main()函数。关键字return可以后跟任何表达式，函数不一定都返回一个值。不含表达式的return语句用于使控制返回调用者（但不返回有用的值），如同在执行到函数的终结右花括号时脱离函数一样。调用函数也可以不用一个函数所返回的值。如果一个函数没有返回值，则可用关键字void来说明函数的类型，例如本例在函数名main()前面有一个void，说明main()函数没有返回值，没有返回值的函数可以省去return语句。在某些情况下，函数体也可以既没有声明部分，也没有执行部分。

（3）如何调用函数 func()。main()函数前的说明语句 int func(int x);表明 func()有一个 int 类型形参并返回一个 int 类型函数值的函数。这个说明称为函数原型，要与函数的定义和使用相一致。main()函数通过语句 sum=func(m);调用 func()函数，调用时将实际参数 m 的值传送给 func()函数的形式参数 x，通过执行 func()函数得到一个返回值，把这个值赋给变量 sum。

（4）一个 C 程序的主函数 main()可以放在程序的任何位置，例如例 1.4 程序的 main()函数也可以放在 func()函数的后面，写成例 1.5 的形式。

【例 1.5】要求同例 1.3，交换例 1.4 中 main()与 func()的位置，实际上是例 1.3 的第三种写法。

```
#include<stdio.h>
/*将自定义函数 func()的位置摆放在 main()之前*/
int func(int x)
{
    int i;
    int s=0;
    for(i=1;i<=x;i++) s=s+i;
    return s;                    /*返回 s 的值，通过 func()带回调用处*/
}
void main()                      /*主函数*/
{
   int m,sum;
   printf("Enter a positive integer: ");
   scanf("%d",&m);
   sum=func(m);                  /*去调用函数 func()*/
   printf("The sum is %d\n",sum);
}
```

其实，main()函数的书写位置不论在前还是在后，在运行程序时，首先都会自动找到 main()函数，从 main()函数中的语句开始执行。

通过以上几个例子的分析，说明解决简单问题的 C 程序的编写并不是十分困难，从书写清晰，便于程序的阅读、理解和维护的角度出发，读者在一开始就应养成良好的程序书写习惯。下面是书写 C 程序时应注意的几点习惯。

（1）虽然 C 程序中一行可以写多条语句，一条语句也可以分写在多行，为清晰起见，一般一条语句单独占一行。

（2）用{}括起来的部分，通常表示了程序的某一层次结构。{}一般与该结构语句的第一个字母对齐，并单独占一行。

（3）低一层次的语句或说明可比高一层次的语句或说明缩进若干格后书写，以便看起来层次更加清晰，增加程序的可读性。

（4）标识符、关键字之间必须至少加一个空格以示间隔。若已有明显的间隔符，虽然也可不再加空格来间隔，但也可以加一个空格来增加清晰度。

1.2 C 语言概述

为解决实际问题，人们通常会用计算机能够识别的代码来编排一系列的加工步骤，这就构成了计算机程序（Program）。计算机能严格按照这些步骤去做，包括计算机对数据的处理。程序的执行过程实际上就是对程序所表示的数据进行处理的过程。一方面，程序设计语言提