C程序设计实用教程

刘振安　苏仕华　编著

西安电子科技大学出版社

C程序设计实用教程

刘振安　苏仕华　编著

西安电子科技大学出版社
1996

（陕）新登字 010 号

C 程序设计实用教程
刘振安　苏仁华　编著
责任编辑　梁家新

西安电子科技大学出版社出版发行
西安盈洋印刷厂印刷
新华书店经销
开本 787×1092　1/16　印张 16 4/16　字数 381 千字
1994 年 9 月第 1 版　1996 年 6 月第 3 次印刷　印数 10 001—15 000

ISBN 7－5606－0346－7/TP・0131　　定价：14.50 元

内 容 简 介

本书以实例为蓝线，以通俗的语言和简要的内容阐述了C语言的程序设计方法，介绍了使用集成环境编辑完整的实用程序（包括头文件的编制、多个C语言文件及工程文件的编制等），查错及调试程序的方法，并对C语言编程错误进行了较系统的分析，介绍了预防与排除的方法。本书还配有C程序结构化设计实例及习题以加深对C语言的理解。

本书取材新颖、内容丰富、阐述系统，不同领域的读者均可有所受益，可作为大专院校及培训班教材，自学教材及工程技术人员的参考书。

前 言

C语言是一种通用的程序设计语言，它既不是"非常高级"的语言，也不是"低级"语言，而通常被称为"中级"语言。但这并不意味着C语言功能差，难以使用。C语言的通用性和无限制性使得它对许多程序设计来说都比想象中的功能的语言更加通俗，更加有效。目前C语言已用于各个方面的程序设计，无论是设计系统软件（操作系统，编译系统等），还是应用软件（图形处理），数据处理（如企业管理）以及数值计算等都可以很方便地使用C语言。目前各大专院校及成人教育都开设了C语言课程，C语言的学习班就更普遍了，已经有愈来愈多的用户加入学、用C语言的队伍之中。

C语言的特点是多方面的,简单说来，它有如下几个特点:

(1) C语言吸取了汇编语言的精华，使C语言对高级语言来讲是"低级"语言。由于它具有描述准确和目标程序质量高的优点，所以有很强的生命力。

(2) C语言继承和发扬了高级语言的长处，使C语言相对汇编来讲又是"高级"语言。

(3) C语言的规模适中，语言简洁，其编译程序简单而紧凑。它在运行时所需要的支持少，占用的存储空间也小。

(4) C语言的可移植性好，这是指程序从一个环境不加或稍加改动就可搬到另一个完全不同的环境上运行。汇编程序因依赖机器硬件，所以根本不可移植，而一些高级语言，如FORTRAN等编译的程序也是不可移植的。

由于上述几个突出优点，使越来越多的人加入到学习和研究C语言的行列之中。而C语言以它的格式自由，限制较少，语句简洁等特点更使学习者爱不释手。

本书有如下特点:

(1) 以实例为蓝线，以通俗的语言和简要的内容阐述C语言的程序设计方法。

(2) 理论联系实际。书中所列例题简易，针对性强。实例由浅入深，有一定工程背景，能起到学以致用的效果。

(3) 取材新颖，内容丰富，阐述系统。还精选了一些综合应用实例以加深对C语言的理解。

(4) 本书以Turbo C为集成开发环境实验手段，不仅给学习者提供了极大的方便，还使学习者对编辑、编译、查错及连接运行C程序有了完整的认识，从而编制自己的实用程序。

(5) 介绍了大程序的设计方法、C语言头部文件、工程文件、库管理及多个文件的编制和调试技术，使读者能很快掌握C语言的实用技术，这在C语言程序设计的教材中还是一种尝试。

(6) 通过作者自己的经验，较系统地介绍了C语言编程错误、错误分析方法及程序测试方法，以使读者尽快入门并掌握编制完整的C语言应用程序的基本方法，这不仅对初学者大有好处，对具有一定经验的C语言程序设计者也大有益处。

(7) 结合实例，介绍了使用集成环境及调试程序的方法。

(8) 全书突出了结构化、模块化设计的基本思想。

(9) 配备一定数量的基础习题。

本书共分七章。第一章介绍 C 程序设计基础；第二章介绍结构化程序设计基础；第三章介绍函数与变量类型；第四章介绍构造类型（数组和指针）；第五章介绍结构类型；第六章介绍文件；第七章介绍 C 程序结构化设计实例。

本书第一、二、四、六章由刘振安执笔；第三、五章由苏仕华执笔；最后由刘振安修改定稿。

本书是我们写作的 C 语言教材三部曲的第一部。第二部《C 语言实践》解答了本书的全部习题并给出上机指导与实验，可以与本教材配合使用。

第二部本身又具有自己的特点，它从实例入手，介绍语法规则，由浅入深，循序渐进，容易理解。它从另一个侧面入手介绍 C 程序设计方法，而且单独开辟三章，从而在广度与深度上突破纯粹内容配合的限制，所以它除了与前后两部书遥相呼应之外，还可以单独作为学习教材和上机指导。

第三部《高级 C 程序设计基础》，则完全从提高和实用程序设计的角度取材。

在本书的写作过程中，得到许多领导的大力支持。尤其是在中国科学院院士、原我校谷超豪校长为我们提词“让更多的人了解计算机，让更多的人使用计算机”的鼓励下，为普及我国的计算机教育事业而向读者提供了这部 C 语言系列教材。

安徽大学副校长程慧霞教授，北京工业大学副校长沈兰荪教授，我校计算机系主任陈国良教授审阅了书稿；中央候补委员、科技大学校长汤洪高教授也给予了大力支持。对此我们表示感谢，并对引用资料的作者再次表示感谢。

我们曾收到很多读者及同行的来信，给予我们很多支持和鼓励，使我们受益匪浅。在此特向诸位表示感谢，欢迎广大读者一如既往地关心我们，提出宝贵的意见。

因我们才疏学浅，错误之处在所难免，敬请读者和同行批评指正。

作　者

1994 年 2 月于合肥

目　录

第一章　C 程序设计基础

1.1　C 语言的特点

C 语言是 70 年代初期美国贝尔（Bell）实验室 Dennis M.Ritchie 设计的一种程序设计语言，正式发表于 1978 年。

1970 年，Ken Thompson 在早期编程语言 BCPL 的基础上开发了一种新的语言，取名叫“B”。Dennis M.Ritchie 在“B”的基础上，于 1971 年开发了第一个 C 编译程序，1972 年开始使用（主要是在贝尔实验室内部使用）。以后，C 语言又经过多次改进，直到 1975 年用 C 语言编写的 UNIX 操作系统第 6 版公诸于世，C 语言才举世瞩目。

1978 年，Brian Kernighan 和 Dennis M.Ritchie 在 C 程序语言（The C Programming Language）一书中对 C 语言作了详尽的描述。随着微型计算机的日益普及，大量的 C 语言工具相继问世，然而这些工具没有统一的标准，还有不一致的现象。为了改变这种情况，ANSI 于 1983 年成立了一个专门委员会，为 C 语言制定了 ANSI 标准。Turbo C 不仅满足 ANSI 标准，还提供了一个集成开发环境，同时也按传统方式提供了命令行编译程序版本以满足不同用户的需要。

C 语言是一种通用的程序设计语言。C 语言的通用性和无限制性，使得它对许多程序设计来说都比想象中的功能的语言更加通俗，更加有效。目前 C 语言已用于各个方面的程序设计，无论是设计系统软件（操作系统，编译系统等），还是应用软件（如图形处理），数据处理（如企业管理）以及数值计算等都可以很方便地使用 C 语言。

C 语言有如下几个特点：

(1) C 语言吸取了汇编语言的精华，使 C 语言对高级语言来讲是“低级”语言。汇编语言是一种面向机器的程序设计语言，尽管它的编程相对高级语言来要麻烦得多，但由于它具有描述准确和目标程序质量高的优点，所以汇编语言仍然有很强的生命力。

①C 语言提供了对位、字节以及地址的操作，使程序可以直接对内存及指定寄存器进行操作。

②C 语言吸取了宏汇编技术中的某些灵活的处理方法，提供宏代换 #define 和文件包含#include 的预处理命令。

③C 语言能很方便地与汇编语言连接。C 程序中引用汇编程序与引用 C 语言函数一样，这为某些特殊功能程序的设计提供了方便。

(2) C 语言继承和发扬了高级语言的长处，使 C 语言相对汇编来讲又是“高级”语言。

①吸取了 ALGOL 的分程序结构思想。C 程序中，可用一对花括号“{ }”把一串语句括起来而成为复合句（分程序），在括号内可定义变量。它还继承了 PASCAL 的数据类型，提供了相当完备的数据结构。

②吸取了 FORTRAN 语言的模块结构思想。C 程序中，它的每一个函数都是独立的，可以单独编译。对设计一个大的程序来说，有利于分工编程和调试。

③C 程序中的任何函数都允许递归，这样对某些算法实现起来就十分方便、清晰。

(3) C 语言的规模适中、语言简洁，其编译程序简单而紧凑。C 语言在表示上尽可能地简洁（比如用一对花括号{}代替 Begin__End，运算符尽量缩写等），语言的许多成分都是通过显示函数调用来完成的，比如 C 语言中没有 I/O 设施，也没有并行操作等；另一方面，它在运行时所需要的支持少，占用的存储空间也小。

(4) C 语言的可移植性好，这是指程序从一个环境不加或稍加改动就可搬到另一个完全不同的环境下去运行。汇编程序因依赖机器硬件，所以根本不可移植，而一些高级语言，如 FORTRAN 等编译的程序也是不可移植的。

(5) 生成的代码质量高，在代码效率方面可以和汇编语言相媲美。

C 语言的优点很多，但也有些不足之处。例如运算符优先级太多，不便记忆；有些还与常规约定有所不同；类型检验较弱，转换比较随便，所以不太安全。尽管如此，由于上述几个突出优点，C 语言仍不失为一个实用的通用程序设计语言，学习和使用它的人越来越多。

1.2　简单的 C 程序结构及函数

1.2.1　简单的 C 程序结构

用 C 语言编写的程序称为 C 语言源程序，简称 C 程序。C 程序一般由一个或若干个函数组成，而这些函数可以保存在一个或几个源程序文件中，这些文件都以 .C 作为文件扩展名。在组成一个程序的若干函数中必须有一个且只能有一个名为 main 的函数（称为主函数），在运行 C 程序时总是从 main 函数开始执行。一个函数在其名字之后一定要有一对圆括号，圆括号中的参数可有可无。我们首先看一个简单的 C 程序。

【例 1.1】　打印字符串。

```
/*    功能: 打印字符串    */
main()
{
 printf("Hello! How are you ?");      /*    打印字符串    */
}
```

这是一个完整的 C 程序，以"/*"开头到"*/"结尾之间的内容是注释，编译时不产生目标代码。

在例 1.1 的程序中有一对花括号，类似于 PASCAL 语言中的"Begin__End"，可以看作为程序体括号，也可以用来括起任何一组语句，从而构成一个复合句（或叫分程序），要注意在一个函数中至少应有一对花括号。C 程序的一般函数或"main()"之后应有一个"{"，函数的最后应是一个"}"，在一个 C 程序或一个函数中，"{"和"}"必须是成对出现。

printf("Hello！ How are you ?");语句是一个函数调用，调用名叫 printf 的库函数，括号内双引号括起来的是所带的参数，即要打印的内容。printf 是标准输出函数，对应于输出设备终端显示器，上述语句表示要在终端输出字符串

Hello！ How are you ?

在右括号")"之后的分号";"是语句结束标志。也就是说，C 语言与 BASIC 语言不一样，

C语言必须用";"号做为语句的结束标志.

1.2.2 C函数简述

C函数分为两类。一类是系统本身提供的库函数（标准函数），在编程时只要在需要的地方写上函数名，带上参数即可调用库函数，一般情况下要在主函数之前加上相应的蕴含函数库名。比如要调用数学库函数，就要在main()之前加上#include <math.h>，C语言有非常丰富的库函数；另一类叫做自定义函数，程序设计人员可根据编程的需要自行设计一段程序完成一个特定的功能，它相当于FORTRAN中的子例程子程序或函数子程序，等价于PASCAL中的函数或过程.

由此可见，C语言中函数的概念类似于其它语言中的子例程或子程序的概念，有所不同的是C语言中的主函数也被称为函数并且将其定义为main()。因此必须记住：一个C语言程序由一个或多个函数组成，其中必须有一个主函数main()，而且一个可执行的C语言程序总是从main()函数开始执行的。

C语言函数使用简单、方便，执行效率高。在C语言程序设计中，要形成良好的设计风格，一般是用多个小函数或多个小程序构成一个大程序，每一个小函数或小程序完成一个独立的功能，单独编译和调试。请看下面的例子:

【例1.2】 求bin(n, k) = n!/(k!*(n-k)!).

```
main()
{
    int  n, k;
    n=6;
    k=5;
    printf("bin(n, k) = %d\n", fac(n) / (fac(k) * fac(n-k)));
}
fac(m)      /*   求m的阶乘，即m!   */
int  m;
{   int  i, h;
    h=1;
    for(i=1; i<=m; ++i)
    h=h*i;
    return(h);
}
```

以上是个求阶乘的函数fac(m)及调用它的主程序（主函数）。主函数中要求函数bin(n, k)=n!/(k!*(n-k)!)调用了3次即fac(n), fac(k), fac(n-k)。从上例可以看出，每一个函数都有基本相同的形式。

函数类型说明 函数名（参数表，如有的话)

参数说明，如有的话

{

变量说明

语句部分

}

函数可按任何顺序出现，且可出现在一个源程序文件或多个源程序文件中。函数定义中不可缺少的部分是:

函数名 ( )

{

}

其它部分根据需要来确定有无。分析函数的定义形式，可以把定义分为两部分，即函数说明部分和函数体。

1. 函数说明部分

函数类型说明定义函数返回值的数据类型。C 程序所使用的基本数据类型有 int（整型）、char（字符型）、float（浮点型），没有类型说明的函数隐含为整型。在 C 程序里，子程序和函数是一个意思，都称作函数。

函数名是识别函数的名字。有效的名字是以英文字母(a～z, A～Z)或下划线（_）开始的英文字母、数字及下划线组成的字符序列，一般是以前 8 个字符为准。此外，在 C 程序里，函数名用大写字母和小写字母表示被认为是两个不同的函数。

参数表在函数名后的圆括号( ) 内，这里的参数是形式参数，简称形参（亦称哑元）。当该函数被调用时，形参将被实在参数（亦称实元）所替换，这种替换常叫做哑实结合。函数也可以没有参数，但函数名后的圆括号( ) 不能省略。

2. 函数体

函数体是需要完成功能的处理部分，它从大括号{ 开始，直到与此对应的大括号 }为止。

变量说明通常接在{ 后面，基本类型有 int、float 和 char。接在变量说明后的是语句部分。函数体的最后是 }，表示该函数到此结束。

1.2.3　最小 C 函数

在 BASIC 语言程序里，最小程序是仅有一个 STOP 语句的程序。最小 C 程序可以写成如下形式:

tmpc ( ){ }

tmpc ( ) 是个函数名。这里没有用 main ( ) 这个名字，为的是使它更具有一般性。

在上面的程序里，因为 { } 内没有任何可供执行的语句，所以该函数一旦被调用，什么也不做而立即返回到调用它的函数里去。虽然如此，这个函数还是很有价值的。在程序开发时，为给将来要设置的函数事先安排一个位置，往往暂时先用这个形式，等函数设计好以后，为它起一个新的名字，取而代之即可。

1.2.4　基本的输入与输出

输入输出设施以标准函数形式提供，程序在引用库函数的每一个源程序文件的开头处含有一行: #include <stdio.h> （scanf、printf 函数除外，但也有的 C 编译要求用 #include <stdio.h>语句)。

文件 stdio.h 定义了 I/O 库所用的某些宏和变量，使用#include 把 stdio.h 文件包含进来，一起编译。下面简要介绍格式化输入输出函数 scanf( )和 printf( )。

C 语言的标准库中提供了两个很有用的格式化输入输出标准函数，它类似于 FORTRAN 中的 READ 和 WRITE 语句，为程序员实现各种格式输入输出提供了方便。

格式化输出函数 printf 的形式如下:

printf (控制字符串，参数 1，参数 2，...);

printf 的功能是按照控制字符串将参数进行转换，按格式在标准输出设备上输出。在控制字符串中包含两种字符: 一种是普通字符，将原样输出; 另一种是格式符，以"%"开头，其后紧跟格式字符，说明对应参数的输出格式，下面先介绍最常用的 4 种格式符:

d——将参数按十进制形式输出。

c——将参数看作单个字符输出。

s——将参数所指出的字符串输出，字符串以空字符为终止符。

f——将参数按浮点数形式输出。

例如有下面语句

```
x1 = 1234;
x2 = 2345;
printf ("x1 = %d x2 = %d\n", x1, x2 );
```

其中"x1 = %d x2 = %d\n"是控制字符串，x1、x2 是参数，控制字符串中的 x1 = 和 x2 = 都是普通字符，按原样输出，两个%d 是格式说明符，依次说明 x1、x2 均按十进制(整型) 形式输出。上述语句执行后的输出结果为

```
x1 = 1234        x2 = 2345
```

格式化输入函数 scanf 提供的格式与 printf 类似，为

scanf (控制字符串，参数 1，参数 2，...);

scanf 函数实现从标准输入设备 (终端) 上按控制字符所规定的格式输入数值或字符，并将输入内容存在参数所指定的单元中。对于参数的书写，程序设计人员要特别注意，printf 中的参数是给出要输出值的变量名，而 scanf 中的参数是给出要接收数据的变量地址。例如语句

```
scanf ("%d  %d", &x, &y );
```

该语句实现从终端输入二个十进制数分别赋给 x 和 y。这里&x 和&y 表示 x 及 y 的地址。在输入的二个十进制数之间用空格分隔。

1.3　典型 C 语言程序结构

【例 1.3】　已知三角形的边长分别为 A、B、C，求三角形的面积及由它组成的一个平行四边形的面积。

已知三角形的边长分别为 A、B、C，虽然由它可以组成几个平行四边形，但任意一个平行四边形的面积均是三角形面积的 2 倍。

为了便于理解，给语句编上号，空行是为了说明程序风格而设的。

```
#include <stdio.h>
#include <math.h>
#define RATIO 2

main( )
{
      float   a, b, c, d, area( );      /* 定义为浮点数 */

      printf ("请输入三角形的三边边长:\n");
      scanf ("%f %f %f", &a, &b, &c);

      if ( a * b * c == 0 )
            {
            printf ("输入为: %f %f %f\n", a, b, c );
            printf ("输入有错, 请重新运行。\n");
             }
      else
             {
             d = area ( a, b, c)
             printf ("三角形的三个边为: %f %f %f\n", a, b, c );
             printf ("三角形面积为: %f\n",  d );
             printf ("平形四边形面积为: %f\n", RATIO * d );
             }
}

 /* 本函数计算三角形的面积 */
float   area (a,  b,  c);
float a, b, c;
{
      float   s;

      s = (a+b+c);      /* 计算三角形的周长 */
      s = s / 2.0;
      return ( sqrt (s *  (s-a) * (s-b) * s-c) );
}
```

上述程序中出现的 scanf 函数是实现从终端输入 3 个浮点数分别赋给变量 a、b、c；第一个 printf 显示一个汉字串，第二个 printf 除了显示汉字串外还把输入的边长输出出来。程序中用到了汉字，这可以在汉字操作系统支持下运行。但更方便的是使用汉卡，如我们生产的"星河"汉卡。星河汉卡不需要汉字操作系统的支持，它在西文系统里运行，所

以西文软件可以直接使用汉字。建议在此系统下，给程序写详细的中文注释。

下面就对例 1.3 的程序分步说明。

1.3.1　函数与主函数

C函数分为库函数（标准函数）和自定义函数两类，本例中的求三角形面积的函数area()，就是自定义函数。

我们必须记住：一个可执行的C语言程序总是从main()函数开始执行的。

1.3.2　C语言预处理器

C语言预处理器是C编译程序的一部分，它负责分析处理几种特殊的语句，这些语句被称为预处理语句。

顾名思义，预处理程序对这几种特殊语句的分析处理是在编译程序的其它部分之前进行的。为了与一般C程序语句相区别，所有预处理语句都以位于行首的符号"#"开始。

预处理语句有三种，它们分别是宏定义、文件包含和条件编译。

C预处理器和有关语句能够帮助程序员编写易读、易改、易移植并便于调试的程序，对于模块化程序设计也提供了很大的帮助。

例 1.3 中有前两种形式。语句

```
#define RATIO 2
```

是用名字RATIO来代替数字2，是最简单的形式。又例如:

```
#define   PI    3.14159
#define   YES   1
#define   NO    0
```

则定义PI、YES、NO分别是3.14159、1和0。

预处理器把所有出现的、被定义的名字全部替换成对应的定义串。#define中的名字与C中标识符有相同的形式，为了区别，往往用大写字母来表示（源程序用小写字母）。

语句1和2是文件包含语句，它指的是一个程序把另一个指定文件的内容包含进来。书写时，可以使用引号也可以用尖括号。例如

```
#include "filename"
```

或

```
#include <filename>
```

都是在程序中把文件filename的内容（引号或尖括号是一定要的）包含进来。

另外还要注意，文件名是用双引号还是尖括号，其含义并不一样。使用尖括号时，C编译系统将首先在C语言系统指定的目录中寻找包含文件，如果没有找到，就到当前工作目录中去寻找，这是引用系统提供的包含文件所采用的方法。自己定义的包含文件一般都放在自己指定的目录中，所以在引用它们时，就采用双引号以通知C编译在用户指定的目录下和当前目录下寻找包含文件。例如自己定义的包含文件myfile.h，引用形式为

```
#include "myfile.h"
```

在程序设计中，文件包含语句是非常有用的。一般C系统中带有大量的.h文件，用户可根据不同的需要将相应的.h文件包含起来。

在本程序中，因为要用到C语言提供的数学运算库，所以要用

```
#include <math.h>
```

语句。而标准输入输出是定义在库函数 stdio.h 中的，所以要用

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

语句。可以说，一般的C程序都离不开这条语句。初学C语言的读者也最容易遗漏这条语句。

1.3.3 程序注释

程序中以"/*"开头到"*/"结尾之间的内容是注释，用以帮助阅读、理解及维护程序。在编译时，注释部分被忽略，不产生目标代码。一个好的程序设计者应该在程序中正确使用注释来说明整个程序的功能、注意事项及有关算法等。有人认为，注释愈多，程序的可读性和可维护性愈好，其实不然，仅需在必要的地方进行注释即可。也就是说，应该加的是程序的注释，不是对程序的说明。例如上面程序中第7句的注释，就是不必要的说明。

注意：不要漏掉 /* 或 */，一定要配对使用它们。

1.3.4 程序语句

和其它高级语言一样，C语言的语句也是用来向计算机系统发出操作指令的。一条语句经过编译后生成若干条机器指令。在C语言中只有可执行语句，没有非执行语句(BASIC和FORTRAN语言都有执行和非执行语句之分)。一个为实现特定目的的程序应包含若干条语句，即一个C程序可以由若干个源程序文件（分别编译的文件模块）组成，一个源文件可以由若干个函数和预编译命令组成，一个函数又由数据定义和执行语句两部分组成。一条完整的语句必须以分号";"结束。可以把程序语句分成如下几类：

1. 说明语句

用来说明变量的类型和初值。例1.3中的第7条和第28条语句是把变量说明为浮点数，下面的语句是把变量sum说明为整型变量，并赋初值为零。

```
int  sum=0;
```

2. 表达式语句

由一个表达式构成一个语句，用以描述算术运算、逻辑运算或产生某种特定动作。最典型的用法是由赋值表达式构成一个赋值语句。例如：a=3是一个赋值表达式，而a=3;就是一个赋值语句。从中可以看到，在一个表达式的最后加一个分号就构成了一个语句。一个语句的最后必须出现分号，分号是语句中不可缺少的一部分。又例如

```
i=i+1                是表达式，不是语句
i=i+1;               是语句
```

任何表达式都可以加上分号而成为语句。例如

```
i++;
```

是一个语句，作用是使i的值加1，而

```
x+y;
```

也是一个语句，作用是完成x+y的操作，它是合法的，但没有实际意义。例1.3中的语句

```
s=a+b+c;
```