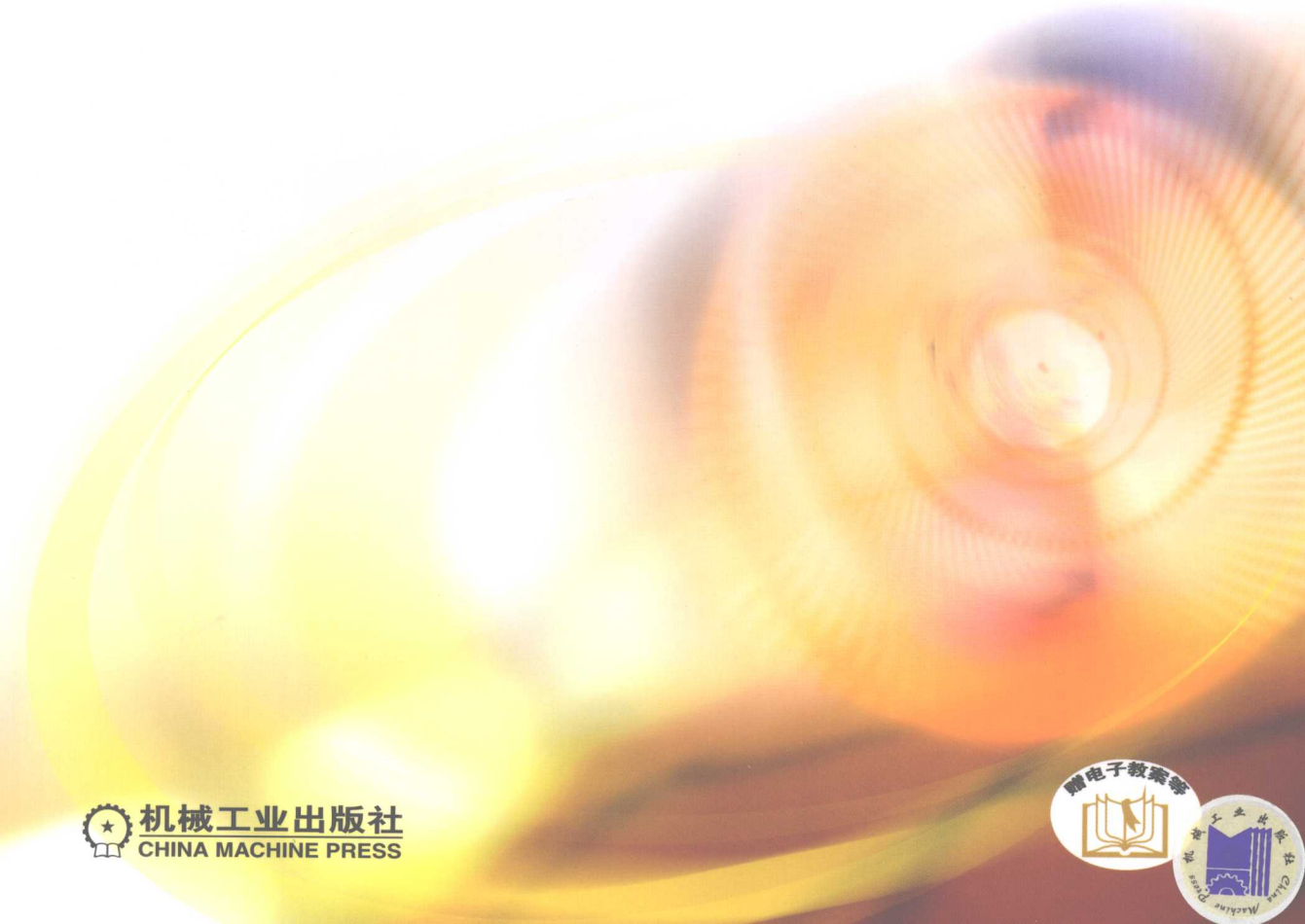
全国高等工科教育自动化类专业规划教材

C语言程序设计教程

齐从谦　主编



机械工业出版社
CHINA MACHINE PRESS

全国高等工科教育自动化类专业规划教材

C语言程序设计教程

主　编　齐从谦
副主编　甘　屹
参　编　丁鸿杰　王士兰

机械工业出版社

C 语言是一种模块化、结构化的程序设计语言，它功能丰富，表达能力强，应用面广，还具有灵活方便，目标程序效率高，程序代码可移植性强，且能对硬件直接进行操作等多方面的优点，深受国内外工程技术人员的欢迎，被广泛用作科学计算和事务处理的计算机语言，学会和掌握 C 语言是工程技术人员的基本功之一。即使是在信息技术快速发展、突飞猛进的今天，C 语言仍然是编程人员的重要工具，而且是进一步学习 C ++/VC ++、JAVA、C#及 .net 等高级程序设计的基础。

本书主编早在 20 世纪 80 年代在国外学习和工作期间，就用 C 语言编程解决了数控加工、机器人控制、计算机外设的驱动和通信等技术问题；回国后即把自己学习和应用 C 语言的体会编写成一本专著，作为大学工科学生的程序设计教材。根据发展的需要和应用型本科、高职高专学生的实际情况，作者对原书全部内容做了较大的更新，增加了大量的应用实例，特别在解决问题的方式方法上注重对学生能力的培养。每章都安排了大量的习题，供学生练习、自学使用。全书内容丰富、新颖、实用。

本书可作为高等工科院校机械制造及自动化、机械电子工程、工业工程、管理工程等专业本专科学生的程序设计教材，也可供广大企业、科研单位的工程技术人员学习和参考。

为方便教学，本书配有免费电子教案，凡选用本书作为教材的学校，均可来电索取，咨询电话：010-88379564。

图书在版编目(CIP)数据

C 语言程序设计教程/齐从谦主编 . —北京：机械工业出版社，2007.3

全国高等工科教育自动化类专业规划教材

ISBN 978-7-111-20826-6

Ⅰ.C... Ⅱ.齐... 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(2007)第 013154 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策划编辑：于 宁 责任编辑：曲世海 版式设计：霍永明

责任校对：陈延翔 封面设计：鞠 杨 责任印制：李 妍

北京铭成印刷有限公司印刷

2007 年 2 月第 1 版第 1 次印刷

184mm × 260mm · 17.25 印张 · 423 千字

0 001—4 000 册

标准书号：ISBN 978-7-111-20826-6

定价：26.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

销售服务热线电话：(010)68326294

购书热线电话：(010)88379639、88379641、88379643

编辑热线电话：(010)88379564

封面无防伪标均为盗版

前　言

C语言是一种模块化、结构化的程序设计语言，它功能丰富，表达能力强，应用面广，还具有灵活方便，目标程序效率高，程序代码可移植性强，且能对硬件直接进行操作等多方面的优点，深受国内外广大工程技术人员的欢迎，被广泛用作科学计算和事务处理的计算机语言，学会和掌握C语言是工程技术人员的基本功之一。即使是在信息技术快速发展、突飞猛进的今天，C语言仍然是编程人员的重要工具，而且是进一步学习C++/VC++、JAVA、C#及.net等高级程序设计语言的基础。

本书由上海师范大学天华学院齐从谦教授任主编(负责编写第5章~第7章)，上海理工大学甘屹博士任副主编(负责编写第1章和第4章)，上海工程化学设计院丁鸿杰高级工程师(负责编写第8章和第9章)和同济大学王士兰副教授(负责编写第2章、第3章)参加编写，由齐从谦教授统稿、定稿。本书主编早在20世纪80年代在国外学习和工作期间，就用C语言编程解决了数控加工、机器人控制、计算机外设的驱动和通信等技术问题；回国后即把自己学习和应用C语言的体会编写成一本专著，作为大学工科学生的程序设计教材。根据发展的需要和应用型本科、高职高专学生的实际情况，参考上海市计算机等级考试“二级C语言程序设计考试大纲”的要求，作者对原书全部内容做了较大的更新，增加了大量的应用实例，特别在解决问题的方式方法上注重对学生能力的培养。每章都安排了大量的习题，供学生练习、自学使用。全书内容丰富、新颖、实用。

作者积20余年来C语言程序设计教学和工程应用实践的经验，从中深刻体会到：学习C语言的目的不仅仅是为了掌握一门编程技术，更多的是用C语言的严谨、规范、细腻和灵活来磨炼自己，学会用“自顶向下、逐步细分、模块化编程”等手段解决一般问题的方法；在C语言的教与学中，不仅要求学生掌握C语言的编程规律和编程技巧，还要学活用活，举一反三，从而逐步培养和训练学生掌握解决问题的方法、能力和技巧。这恰恰是素质教育的重要内容。

本书可作为高等工科院校机械制造及自动化、机械电子工程、工业工程、管理工程等专业本专科学生的程序设计教材，也可供广大企业、科研单位的工程技术人员学习、参考。

在本书编写过程中，参阅了许多同行专家编著的教材和资料，得到了不少启发和收益，在此向编著者表示诚挚的谢意。上海师范大学天华学院杨贤英同志为本书的文字录入、编排打印付出了辛勤的劳动。编者一并在此表示感谢。

为方便教学，本书配有免费电子教案，凡选用本书作为教材的学校，均可来电索取，咨询电话：010-88379564。

C语言本身就是一门发展迅速的新兴学科和技术，新的知识和技术资料不断涌现，由于编者水平有限，书中难免有错误和疏漏之处，敬请各位师生及广大读者给予批评指正。

编　者

目　录

第1章　C语言概述

1.1　C语言的发展历史

C语言是美国贝尔(Bell)实验室于1972年开发出的一种简捷而又高效的程序设计语言，最初由Dennis Ritchie设计编写，后来他与Ken Thompson一起用它重写了Unix操作系统，从此建立起的C语言便逐步成为程序设计人员的一种有力工具。

此后，C语言发展很快，十分流行，被认为是当今结构和语言都最简单且通用性最强的程序设计语言之一。C语言属于简捷、易用、规则性强而又可靠的结构化程序设计语言系列，是一种高效率、可移植的语言，在某一个系统上用C语言编写好的程序，只需对程序稍作修改甚至不加修改，即可在另一个系统上运行。若要用于系统软件的设计，如编写编译程序、操作系统和文本处理程序等，则C语言是最好的一种开发工具。由于C语言具有描述系统程序的属性，从而引起许多软件工作者在开发商务应用、数据库应用、空白表格软件和通信包软件等方面发挥了C语言的潜在优势；C语言提供汇编语言通常所具有的存取操作，因此，程序员能使他们的程序取得最大的效率。

C语言的编译系统还包括可供C程序员充分有效使用的50～200个内部函数。此外，C语言还允许用户对其功能进行扩充，即根据自己的需要设计并编译通过而生成的自定义函数，然后按系统规范添加到系统标准函数库中去。以短小、简单而又直接的方法来解决普通而复杂的程序设计问题，C语言是行之有效的程序设计语言之一。

可以说C语言反映了当前计算机的能力，是一种高效实用、灵活的软件开发工具。尽管它的首次开发使用是在Unix操作系统上，但今天它已相当普遍地应用于个人计算机(PC)领域。由于越来越多的开发人员和用户使用C语言，所以必须有一套保证C程序员使用相同语言工作的工具。通常人们将B. W. Kernighan和Dennis Ritchie于1978年所著的《The C Programming Language》一书中提出的C语言版本作为基础，并称该版本为标准C。1983年美国国家标准局(ANSI)语言标准化委员会对C语言问世以来的若干发展、补充进行了较系统的归纳和总结，并进一步作了修改订正后，公布了一个C语言标准草案83 ANSI C，1987年又推出了87 ANSI标准C。此后，一些在世界上颇有影响的软件公司如Microsoft、Borland International等相继在87 ANSI标准C的基础上又陆续发展了更为实用、灵活、功能丰富的C编译系统，如Microsoft C 1.0～6.0版本，Turbo C 2.0、3.0，Turbo C++，Borland C++等。

1.2　一个简单的C程序

这一节，首先通过一个非常简单的C语言程序，使读者初步了解C程序的概貌和最基本的特征。

例1.1　求三个数的平均值。

```
1：#include "stdio.h"
2：/* 一个简单的程序例 */
3：main()
4：{
5：    float a,b,c,average;
6：    a=5.6；b=4.7；c=8.3;
7：    average=(a+b+c)/3.0;
8：    printf("average=%f",average);
9：}
```

这个小程序共有 9 行，每行的标号仅仅是供讲述用的，在实际的编程中并不出现。各行的意义如下：

第 1 行：告知计算机本程序需要的有关信息包含(include)在头文件“stdio.h”内。

第 2 行：这是一个注释行，常用来说明程序的主要功能、变量的含义或其他说明。当我们了解如何在 C 语言中使用汉字后，也可以用汉语予以注释。对 C 语言来说，使用“/*”和“*/”把注释的内容夹在其间；对于 C++来说，可以用更简捷的方式，即把每行注释的内容放在符号“//”之后，编译系统在遇到“/*”和“*/”之间的内容(对于 C++则是“//”之后的内容)时，将分别跳过而不予编译，也就是说注释的内容是仅供程序员阅读、分析程序作参考的。

第 3 行：main()表示一个名称为 main 的函数。在 C 程序中，凡是在一个标识符之后带有一对圆括号“(”、“)”，就表示是一个函数，括号内可以含有参数或默认参数。函数是 C 程序中的基本模块，这里的 main()是一个特殊的函数，称之为主函数，在每一个 C 程序的源文件中只能有一个主函数 main()，而程序的执行过程总是从主函数开始，又在主函数中结束。

第 4 行：这个左花括号表示组成函数的语句开始，函数的具体内容就是在左花括号和最后的右花括号之间的部分。

第 5 行：这一行说明 a、b、c、average 四个变量为实型变量。注意到这里采用了“缩进”格式，这种格式能够使程序显得层次分明、清晰和有条理。建议初学者都应该采用合理的“缩进”格式书写 C 程序的源文件。

第 6 行：给三个实型变量分别赋以具体的数值。

第 7 行：求三个数的平均值，并把结果赋给 average。

第 8 行：在屏幕上显示 average 的数值，这里 printf()是一个标准函数，它是由编译系统提供的，它的功能包含在第一行中的“stdio.h”文件(称之为头文件)中。

第 9 行：以右花括号结束 main()和整个程序。

例 1.1 无疑是一个正确的程序，经编译、运行后程序会在屏幕上显示结果：

```
average=6.2
```

但它显然是一个不太理想的程序，因为 a、b、c 三个实型数在程序中是固定的，如果想求另外三个数的平均值，必须对程序进行修改，还要对源程序重新进行编译。为此，可以对例 1.1 稍作改进，以实现求任意三个数的平均值。

例 1.2 输入三个实型数，并求它们的平均值。

```
#include "stdio.h"
main()
{
  float a,b,c,average;
  printf("请输入三个实型数:\n");
  sacnf("a=%f b=%f c=%f",&a,&b,&c);
  average=(a+b+c)/3.0;
  printf("average=%f",average);
}
```

这个程序中增加的“printf("请输入三个实型数:\n");”一行语句，是利用标准输出函数在屏幕上原封不动地把双引号中的“请输入三个实型数:”显示出来(当然,必须在中文操作系统环境下才能显示中文)，用以提醒操作人员，并把光标返回到下一行的第一列位置上。所增加的另一行语句中的 scanf()也是一个包含在“stdio.h”文件中的标准函数，它的功能是通过键盘接受操作者输入的数据。

经过改进的程序可以使操作人员输入任意三个合法的实型数并求解它们的平均值。

这个例子毕竟十分简单。通常用 C 或 C++所设计的用来解决实际问题的程序往往要复杂得多，这时，我们希望写在 main()中的内容愈简洁愈好，所以总是把那些实现某些特定功能的语句另外写成一个相对独立的函数。而在 main()内往往只进行数据输入和最后的输出工作，而具体的运算操作则是通过传递参数去调用那些相关的函数就可以了。按照这一思路，可以对例 1.2 作进一步的改动。

例 1.3 通过函数调用，求三个数的平均值。

```
#include "stdio.h"
float ave(float x,float y,float z);          /*对求平均值的函数进行声明*/
main()
{
  float a,b,c,average;
  printf("Enter 3 real numbers:\n");
  scanf("%f %f %f",&a,&b,&c);
  average=ave(a,b,c);                        /*传递参数,调用求平均值的函数,并接受函数
                                               的返回值*/
  printf("average=%f",average);              /*输出计算结果*/
}                                            /*程序结束*/

float ave(float x,float y,float z)           /*对求平均值的函数进行定义*/
{
  float aver;
  aver=(x+y+z)/3.0;
```

```
    return(aver);                          /* 把所求得的平均值返回主调函数 */
}
```

我们看到在主函数 main( )中并没有求三个数平均值的具体运算，这个运算是在名为 ave 的函数内实现，然后通过 return( )函数，把所求得的平均值即 aver 的值返回到主函数中，再把它赋值给变量 average。由于 main( )函数要调用 ave 函数，而 ave( )函数定义在 main 函数之后，所以需要在 main 函数之前对 ave( )函数进行说明(或声明)。一是说明 ave( )是一个函数，二是说明 ave( )函数的返回值是实型，三是对 ave( )函数中的参数进行说明。

最后通过一个稍微复杂的例子来描述 C 程序的基本结构。

例 1.4 输入三个实型数，分别求它们的平均值、平方和及均方根值。

```
#include "stdio.h"
#include "math.h"
float ave(float x,float y,float z);          /* 分别对三个函数进行声明 */
float sum_square(float x,float y,float z);
float square_root(float x,float y,float z);
main( )
{
  float a,b,c,average,s,sn;                  /* 在主函数中定义变量并输入数据 */
  printf("Enter 3 real numbers:\n");
  scanf("%f,%f,%f",&a,&b,&c);
  average = ave(a,b,c);                      /* 对三个函数依次调用 */
  s = sum_square(a,b,c);
  sn = square_root(a,b,c);
  printf("average = %f \n",average);         /* 分别输出计算结果 */
  printf("sum_square = %f \n",s);
  printf("square_root = %f \n",sn);
}
foal ave(float x,float y,float z)            /* 对 ave 函数进行定义 */
{
  float aver;
  aver = (x + y + z)/3.0;
  return(aver);                              /* 把计算结果返回主调函数 */
}

float sum_square(float x,float y,float z)    /* 对 sum_square 函数进行定义 */
{
  float s_abc;
  s_abc = x * x + y * y + z * z ;
```

```
    return(s_abc);                          /* 把计算结果返回主调函数 */
}

float square_root(float x,float y,float z)  /* 对 square_root 函数进行定义 */
{
float sn_abc;
sn_abc = sqrt(x * x + y * y + z * z);
return(sn_abc);                             /* 把计算结果返回主调函数 */
}
```

在最后一个函数 sum_square( ) 中，sqrt( ) 函数也是一个标准库函数，它被包含在一个名为“math.h”的头文件中。该程序第二行的“包含”即完成该使命。

1.3 C语言的特点

通过以上介绍的几个简单程序，读者应该对C语言有了初步的印象。C++语言比C语言在功能上更完善，它基本上包括了C语言的功能，所以这里先简单介绍C语言的共同特点，至于C++语言的特点将在后面介绍。

1）C语言是一种模块化的程序设计语言。C语言程序是按函数形式进行装配的，可以说C语言程序是函数的集合，其中的每个函数都能实现特定的功能。函数有两大类，一类是由标准函数库所提供的，像上节中已出现的 printf( )、scanf( ) 及 sqrt( ) 函数等。C语言编译系统能提供极其丰富的标准函数，又称库函数，程序员可以把它们作为标准部件来装配自己的程序。另一类函数需要自行设计，可以称其为“自定义函数”。设计自己的函数时，同样可以用前面一类标准库函数来装配，自行设计的函数一旦经过编译调试通过并运行证明它是正确无误之后，也可以把它按照一定的规则转化为标准函数，这就是对标准函数库的扩充。以后就可以像使用标准部件那样使用它。这些函数应组织成层次结构。最顶层的一个函数为主函数：main( )，一个C程序文件中必须有一个主函数且只能有一个主函数。除了主函数之外，其他所有的函数在结构上都是平行的，在功能上是独立的；不仅主函数可以调用其他函数，其他函数之间也可以互相调用，**但唯独不能调用主函数。**

2）C语言有预处理功能，预处理命令如#include(文件包含)、#define(宏定义)等是在编译时预先处理的命令，它告诉编译系统在哪些系统文件中包含了程序中所用到的标准库函数，说明一些特殊变量的含义。采用预处理命令可以提高程序的可读性和可移植性，并给程序调试提供了方便。

3）C语言要求对程序中的数据和函数都要先作说明，然后才能使用。即在使用数据和函数之前必须先声明和定义它们的类型，稍后我们将会看到，C语言提供有丰富的数据类型，能满足现代程序设计的要求。变量的定义语句和说明语句可以指定其作用域。变量一经定义，同时也就进行了说明。

4）运算符种类丰富。汇编语言中使用的运算符大多数都能作为C语言的运算符来定义。除了在通常的高级语言中使用的算术运算符和逻辑运算符等以外，还定义了按位逻辑运

算符、移位运算符、赋值运算符和逗号运算符等。指明函数用的括号“(”和“)”以及数组中所用的括号“[”和“]”等也被定义成运算符。因此 C 语言的运算符多样化且具有较强的数据处理能力，不仅能完成一般高级语言的运算功能，也能实现低级语言的许多功能；既适用于编写系统软件，也适用于编写面向硬件的应用软件。

5）C 语言主要使用英文字母且大小写敏感。所使用的保留字全部由英文小写字母构成，变量等标识符也由英文小写字母构成。因此，大写和小写字母是当作不同符号来处理的，故称之为大小写敏感。在字符串中可以使用汉字等特殊符号，但只有在汉字能够受到正确处理的情况下方能使用汉字(如在中文 DOS 和 Windows 环境下)。

6）C 语言源程序书写灵活，可以每行一句，也可以一行数句，且可任意换行；但语句间须用“;”隔开，每一条正确的 C 语句都是以“;”结尾的。这样能给程序设计者以较大自由，使程序更为简练。C 语言语法检查不严格，如对数组下标溢出不作检查。因此使用 C 语言编程，对程序设计人员的要求比较高，所以 C 语言往往为熟练的程序人员使用。

7）表达式必定有值。一个常数、一个变量和一个函数都可称为表达式。而由一个或一个以上的表达式进行各种运算的式子也称为表达式。表达式后面若跟着分号“;”则成了一个执行语句，执行一个表达式就是求出该表达式的值。表达式执行后所得到的值叫做评价值。

8）C 语言有高级的控制结构。C 语言源程序原则上是按顺序执行的，其控制流程通过条件分支、多路开关和循环等实现，C 语言的条件分支和循环等控制功能丰富，可以编制出易于理解且便于修改和扩充的结构化程序。

9）C 语言中，指针(地址)可作为数据使用，指示数据存放场所的数据称作地址。其值为地址的变量称作指针。地址和指针这些概念在汇编语言中并不少见，由于 C 语言可以处理与汇编语言级别相当的数据，因而可以实现与汇编语言同等的功能。使用在函数间传递地址的方法可以编出一些通用的函数，同时可以高速地进行某些复杂的处理，而用通常的高级语言是不能进行这些处理的。不过，地址和指针的概念难以理解，使用起来易出错，且这些错误在多数情况下很难查出。

10）C 语言既可以处理由基本数据组合而成的数据，也可以定义并使用由基本数据自由组合而成的集合数据。基本数据有字符型和整型等几种，集合数据中有数组和结构，特别是结构对复杂的数据处理尤为重要，在 C ++ 语言中它被扩充后起着重要的作用。

11）C 语言没有专门的字符串处理功能。字符串和字符有着明确的区别，字符串是作为字符的序列集合而处理的。有对作为基本数据字符进行运算的运算符，却不存在对作为集合数据的字符串(一般说来是对数组)进行运算的运算符，像 BASIC 语言那样用运算符进行字符串的赋值、比较等运算在 C 语言中是做不到的。所有这些运算均需利用函数进行，但在 C ++ 语言中可以自由定义并使用对字符串进行处理的运算符。

12）C 语言中没有其他高级语言所具备的输入输出功能，所有的输入输出操作均由函数来实现。通常由编译程序来实现的输入输出及文件处理功能往往相当复杂，而在 C 语言中却都是通过函数来实现的，就使编译程序变得非常紧凑。而且在编译程序中不含那些受硬件影响较大的输入输出功能，使得程序在不同机种间的可移植性很强。这也是 C 语言得以广泛普及的原因之一。

13）C 语言允许把一个程序分割成若干部分进行处理，即把一个大型程序分割成多个小

的模块进行编译，对它们分别进行编译和调试，最后再把它们连接成一个可执行的程序文件。分割后的文件可以仅含有说明语句和定义语句，也可以仅含有函数。利用这种分割功能可以保证变量的有效范围特征，可以由多人同时对一个程序进行开发，这样有利于开发大型程序。不过，虽然一个文件所含的函数个数没有限制，但对于一个独立的函数来说，它必须在一个文件中结束(即把一个函数分散在多个文件里是不容许的)。

14) 采用C编译系统的命令行编译功能可以方便地构成命令，在启动用户编制的程序时，可以把命令行的参数传递给执行程序。被传递的参数在程序内进行解释，并据此进行适当的处理，因此可以用用户自编的程序对操作系统(OS)的功能进行扩充，这些追加的程序实际上成了与OS所提供的系统命令用法相同的可执行命令，即外部命令。

从以上介绍不难看出，仅仅利用C语言的功能已能显示出迄今为止的其他语言所不具有的优良特色，使用扩充后的C++语言还将能编制出更为精彩漂亮的程序来。

C++语言基本上包含了C语言的功能。因此，在下面的介绍中，凡说到“C语言”的时候，所涉及的功能和规则对C++语言均成立。至于那些只对其中一种语言成立的情形，均特别加以声明。

1.4 C程序的结构和风格

这一节主要从C语言源程序书写的角度来介绍C语言程序的结构和风格。前面已经提到，C语言程序书写格式自由，一行内可写几个语句，一个语句也可以写在多行上，无须换行符，而且C语言程序没有行号，也不像FORTRAN和COBOL语言那样严格规定了语句必须从某一列开始；除少数情况例外，C语言各语法成分间可以空格，总之C语言的灵活性为我们的程序设计带来很多方便。但不能因此而随心所欲，在程序设计中应养成良好的风格，尽量采用清晰、明朗和规范的书写方法，既便于阅读理解，又便于分析调试。

1.4.1 程序书写格式

C程序的书写格式要求遵守结构化方式，对初学者来说，不赞成一行写多个C语句，尽量使用“缩进”格式，使程序层次分明，条理清晰。从一开始就应该养成良好的程序风格。

例如下面的程序，尽管可以通过编译能够运行，但它的源代码让人读起来感到非常杂乱无序，除了节省版面，别无任何好处。

```
main( ){unsigned int albatross,marlin,whale;
char sen _ strins[81];double stars _ insky,fish _ in _ sea;
while(stars _ in _ sky > fish _ in _ sea){save _ whales(&albatross,&marlin,&whale);
which _ greater(& stars _ in _ sky,& fish _ in _ sea,
sea _ string);} printf("%s",sea _ string);
```

把它改写成例1.5的书写格式，给人一种整齐规范，井然有序的感觉，符合结构化程序设计的要求。

例1.5 C程序的正确书写格式。

```
#include "stdio. h"
```

```
main( )
{
  unsined int albatross,marlin,whale;
  char sea _ string[81];
  double stars _ insky, fish _ in _ sea;
  while (stars _ in _ sky > fish _ in _ sea)
      {
       save _ whales(&albatross,&martin,&whale);
       which _ greater(&stars _ in _ sky,&fish _ in _ sea,
       sea _ string);
      }
  printf("%s",sea _ string);
}
```

这样的程序阅读起来一目了然：主函数包含了三种类型数据变量的定义，然后根据是否满足一个条件表达式来循环调用另外两个函数，或者仅显示一个字符串数组的值。

1.4.2 采用缩进格式

所谓缩进格式主要是用来指定每一行的语句从哪一列开始写起。在 C 语言编程中，程序书写的格式是完全自由的，只要符合 C 语言的语法规则，从什么地方开始写都无关紧要。甚至在变量名、常数和表达式等操作符和操作数之间，都可以出现“空白”，而且原则上都可以自由地进行换行。但是由于随意换行和加空白，可能会使写出来的程序很难读。但是，如果像例 1.6 那样，在每一行的开头加入适当的空白字符，可以使程序显得更便于理解。这种在每一行的开头加适当空白的方法就叫做“缩进”。

通常把键盘上的“Tab”（严格地讲应为水平 tab）键和“Enter”（回车）键视同于“Space”（空白）。以下把 Tab、Enter 和 Space 统称为空白字符。使用这些空白键，可以根据需要快速地实现各种缩进。

此外，为提高程序的可读性，在功能模块之间也可插一些空行。在字符串数据内部直接插进 Enter 是不行的，加进 Tab 和 Space 则被视为不同的字符串数据。例 1.6 就是用缩进格式来对例 1.4 的重新书写。为了更具有一般性，可对例 1.4 的功能略作修改，在输入三个任意实型数后，再由键盘输入一个功能选择值。即当输入值分别为 1、2、3 时，程序将自动地在求三个数的平均值、平方和或均方根值函数中进行选择。

例 1.6 输入三个实型数，根据开关值执行相应的运算。

```
switch = 1     求三个数的平均值
switch = 2     求三个数的平方和
switch = 3     求三个数的均方根值

#include < stdio. h >          /* 程序开始 */
#include < math. h >
float ave(float x,float y,float z);
```

```
float sum _ square(float x,float y,float z);
float square _ root(float x,float y,float z);
                                    /* 这是一个空行,把预处理命令与程序文本隔开 */
main()
{
  float a,b,c,average,s,sn;
  int switch;
  printf("请输入3个实型数:\n");
  scanf("%f,%f,%f",&a,&b,&c);
  printf("请输入选择符:");
  scanf("%d",&switch);
  if(switch==1)
    {
      average=ave(a,b,c);
      printf("average=%f \n",average);
    }
  else if(switch==2)
        {
          s=sum _ square(a,b,c)
          printf("sum _ square=%f \n",s);
        }
  else if(switch==3)
        {
          sn=square _ root(a,b,c);
          printf("square _ root=%f \n",sn);
        }
}
                                    /* 这是一个空行,把主函数与其他函数隔开 */
float ave(float x,float y,float z)
{
  float aver;
  aver=(x+y+z)/3.0;
  return(aver);
}
                                    /* 空行,把函数与函数之间也隔开 */
float sum _ square(float x,float y,float z)
{
  float s _ abc;
```

```
    s_abc = x * x + y * y + z * z;
    return(s_abc);
}

                        /* 空行 */
float square_root(float x,float y,float z);
{
    float sn_abc;
    sn_abc = sqrt(x * x + y * y + z * z);
    return(sn_abc);
}
```

采用缩进格式书写，程序中的每个功能模块显得相对独立，流程清晰，作为每个模块开始和结束的花括号“{”、“}”相互对应，即使出现错误或丢失，也容易查找。

1.4.3 用现代风格来说明和定义函数

函数的说明是对程序中所用到的函数的特征——返回值的类型、函数参数的类型和个数进行必要的说明。编译系统以函数声明中给出的信息为依据，对调用表达式进行检测，以保证调用表达式与函数之间能正确传递参数。函数的定义是对函数的功能予以具体的表达，是由变量定义部分和可执行语句组成的独立实体。函数的说明与定义之间要有一定的对应关系。这里所说的对应关系主要是指函数的参数在说明和定义中的一致性，由此而产生两种不同的风格，请看下面的例子。

例 1.7 函数的说明和定义

```
float new_style(int a,float x);          /* 函数的说明或声明,一般放在程序开头处 */
main()
{
    …
    …
}


float new_style(int a,float x)           /* 函数的定义,一般放在主函数之后 */
{
    函数体
}
```

我们注意到函数的说明与函数的返回值的类型、数名、函数的参数个数及类型之间严格的一一对应关系。函数的参数是放在左右圆括号“(”、“)”内的，所不同的是函数说明不包含函数体，它是 C 语言程序中的一条语句，因此必须以分号“;”结尾，它起到预先通知编译系统在该程序文件中存在这样的一个函数的作用，并不要求分配内存空间。函数定义则要求给它分配内存单元，用来存放经编译后的函数指令。这里的函数说明和定义所采用的风格称为“现代风格”，对于例 1.7，传统的说明和定义方式如下：