C语言教程

杜开珍　黄迪明　编著

科学出版社

计算机专业计算机系列教材

非计算机专业计算机系列丛书

C 语言教程

杜开珍　黄迪明　编著

科学出版社

1995

(京)新登字 092 号

内 容 简 介

本书是理工科大学的C语言程序设计教材，是为有一定计算机程序设计知识，旨在学会C语言并掌握其编程技巧的读者而编写的。

本书以讲授C语言为其基本内容，在介绍C语言主要内容的同时，以丰富的示例，深入浅出地为读者展示C语言灵活、精致的编程方法和在工程、科研生产等方面的应用。本书强调程序设计的基本概念，注意培养学生在程序结构和程序可读性方面的编程能力，增强上机能力。书中还为读者介绍一种常用C语言编译系统软件，以做到C语言知识和编程能力的融会贯通。

本书深入浅出，系统全面，既可作为大学、大专的C语言教材，也可为非计算机专业和其他学习C语言的人员阅读参考。

非计算机专业计算机系列丛书

C语言教程

杜开珍 黄迪明 编著

责任编辑 徐 帆

科学出版社出版

北京东黄城根北街16号

邮政编码：100717

[illegible]印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1995年3月第 一 版 开本：787×1092 1/16

1995年3月第一次印刷 印张：21

印数：1—5700 字数：483 000

ISBN 7-03-004375-8 TP·1

定价：23.00元

前　言

C 语言是一种通用的程序设计语言。它的结构简单，数据类型丰富，运算灵活方便。用它编写的程序，具有速度快、效率高、代码紧凑、可移植性好等优点，能够有效地用来编制各种系统软件和应用软件，是当今最为流行的一种计算机语言。

在很多科研、工程、生产过程中大量应用计算机来完成诸如实时控制、数据采集与处理、通信、图形和图象处理等方面的工作。在这些应用领域里，一般的高级语言难以满足要求，而 C 语言却表现出它的灵活、精致、高效等独特的优点。因而越来越受到程序设计人员的青睐。C 语言水平的高低，往往用来衡量一个编程人员水平的高低。因此，迫切需要对理工科学生开设《C 语言程序设计》课程。

目前已出版了多种关于 C 语言的书籍，但在作为教材使用时学生普遍感到学习起来比其他语言困难。由于 C 语言本身的内容比较繁杂，对学生的计算机知识要求也较高，而多数教材的读者对象往往是计算机专业的学生和职业程序员，对一般理工科学生来说确实存在一定的困难。

本书立足于面向非计算机专业的理工科学生，根据学生的特点，在教材编写中注意到章节编排上的循序渐进，并联系教学进度，由浅入深、逐层展开地为学生提供大量典型的 C 语言示例。在例题和习题选择上尽量考虑 C 语言在各种应用领域里的实例，使学生能把语言学习与今后的工作紧密联系起来，并能在课程结束后灵活自如地把 C 语言应用到科研、实践、毕业设计等工作中去，真正做到学以致用。在介绍 C 语言的同时，逐步深入地介绍了一种常用的 C 语言编译系统软件及使用方法，使学生在课程学习的过程中能迅速地掌握 C 程序的编制、编译、调试和运行方法，以加强语言学习和编程能力的有机联系。这样，不仅减少了学生阅读参考文献的盲目性，也减轻了教师备课、补充讲义的负担。

本书共 12 章，每章附有问题与习题。附录中内容，可根据教学实际需要选取，也可留给学生自学。全部例题都在 PC 机上用 Turbo C 调试通过。

第一章至第六章由杜开珍编写，其余部分由黄迪明编写。

本教程由电子科技大学杨国纬教授审阅，许家怡副教授、蒋锡民讲师、周涛讲师对本书的编写提出了不少宝贵意见。四川省非计算机专业计算机等级考试中心及电子科技大学教材科对本书的出版给予了热情地支持。在此一并表示衷心感谢。

由于编著者水平有限，书中的缺点和错误在所难免，恳请读者批评指正。

编著者

1994 年 4 月于电子科技大学

目　录

第一章　C语言基础

§1.1　C语言简史及特点

在现代科技的发展过程中，随着计算机的日益普及和广泛使用，面向各种应用领域的计算机语言也逐渐发展和丰富起来。在各种计算机语言中，C语言可以说是最为成功者之一。C语言与UNIX操作系统有着十分密切的关系，它是在UNIX的研制和开发过程中诞生的。UNIX的开发源于60年代末，它是美国贝尔实验室为PDP-7微型机开发的操作系统。它的第一版完全是用汇编语言编写的。在UNIX开始投入使用以后，贝尔实验室的计算机专家Ken Thompson开发出了一种专门用于编写系统软件的语言：B语言，并用B语言将早期的UNIX重写。一年以后，PDP-11微型机研制出来了，因而必须为它开发UNIX系统。此时贝尔实验室的Dennis. M. Ritchie开始加入到Ken Thompson的工作中。Dennis. M. Ritchie发现，用B语言来编写PDP-11的操作系统存在某些困难，问题之一是B语言只能对字进行操作而PDP-11的内存组织是按字节构造的。为了克服这些困难，Dennis. M. Ritchie在用B语言来描述UNIX的同时开展对B语言本身进行改造和更新的工作，提出了一种严格按结构化程序设计思想进行程序设计的新的语言——C语言。到1973年，新的UNIX版本就完全是用C语言编写的了。起初，C语言只是在贝尔实验室内部使用，但后来随着UNIX操作系统的成功，C变逐渐在各大学和研究机构中加以推广。由于C语言具有精致、灵活、高效等独特优点，深受广大编程人员的喜爱，因此现在人们广泛地采用C语言来完成各种编程任务。

C语言是一种通用的结构程序设计语言，它的结构与Alogl、Pascal的结构非常类似。用C语言编写的程序由多个模块(函数)组成，每个模块完成一项功能，这些模块之间除了必要的参数交流以外彼此互不影响，因而任一模块运行时不会对程序的其它部分产生副作用，这种方式使程序之间很容易实现程序段的共享，利于大型软件的开发和调试，并且使得程序本身层次分明，结构清晰，便于阅读和管理。

和其他语言相比，C语言可以说是一种兼有高级语言和汇编语言两者优点的语言。它把高级语言的基本结构和低级语言的实用性有机地结合起来，一方面它具有高级语言面向用户、语句简单、编程容易的优点，另一方面它又能完成一般高级语言难以完成、只能在汇编级进行的运算或处理，例如地址操作、字位操作、系统功能调用等等。或者说它可以完成大部分汇编功能而又避免了汇编语言编程困难的问题，为用户提供了很大的能力和灵活性。

C语言程序代码通常小于其他语言的代码长度，具有代码紧凑、执行速度快等优点。C的执行速度几乎可以达到汇编的速度。C所具有的丰富的数据类型及简洁的表达方式也使它具备了在实际应用中使用方便的特色。

近几年来，每当一种新的工作站计算机或处理器被研制出来时，首先为这台机器所开发的软件工具一定是两种语言的编译器：汇编语言的汇编程序和C语言的编译程序，其他高级语言的编译器和各种应用软件的开发通常要在此之后逐渐进行。这一现象一方面说明了C语言应用的广泛性，另一方面也进一步推动了C语言的广泛应用。

作为一种系统程序设计语言，C可以用来为各种不同的计算机系统编写有关的系统软件，如操

作系统，编译系统，汇编器及编辑器等。除此以外，C 还广泛地用于编写面向各种应用领域的应用软件，例如数据库管理软件、CAD/CAM 软件、文字处理软件、图形软件、办公自动化软件、科学计算及工程应用软件等等。人们喜爱 C 的原因除上述的优点以外，还由于目前各种 C 的编译器都为用户提供了大量丰富的 C 函数库。这些库函数能帮助人们解决相当多的程序设计问题，简化了程序设计的过程和难度，使编程的速率和效率都有了极大的提高。

§1.2 C 语言的基本结构

在系统地学习 C 语言的各种语句、语法规则和数据类型以前，我们首先应该对 C 语言程序的结构有一个基本的了解。为此我们先看下面的例子：

例 1.1 编写一个程序在屏幕上输出：This is C programming language。

完成这个任务的最简单的 C 程序如下：

```
main()
    {
      printf("This is C programming language\n");
    }
```

如果我们已经学过 Pascal 或 FORTRAN 语言，那么我们可以很容易地写出完成上述任务的 Pascal 和 FORTRAN 程序：

Pascal 程序：

```
PROGRAM example1(output);
  BEGIN
      writeln("This is C programming language")
  END.
```

FORTRAN 程序：

```
      PROGRAM EXAMPLE1
      WRITE( * ,10)
10    FORMAT('This is C programming language')
      END
```

通过对比以上几个程序，你可以猜想得到 C 程序各部分的含义：

程序的第一个词 main 表示"主函数"。实际上所有的 C 程序都必须有一个 main 函数，main 告诉系统程序从这里开始执行。注意，当你用 C 编写程序时，既不要忘了程序第一行的 main，也不要把 main 写成 MAIN，因为 C 语言中对大小写字母的管理与其它语言不同。它不象 FORTRAN 那样只采用大写字母或象 Pascal 那样对大小写字母不加区别。C 语言中可以使用大、小写字母，但把它们当做不同的字符加以处理。因此，切记不要在程序头上写上 MAIN。

一对花括号{ }表示程序段的开始和结束，它与 Pascal 中的 BEGIN、END 相对应。实际上，这两者的使用是十分相似的，它们既分别用于表示程序的开始和结束，也分别用于表示复合语句。

printf 是 C 的一个输出函数，它把一个字符串打印在屏幕上。被打印的字符串用双引号括起来。注意到双引号中的字符\n 并没有在屏幕上显示出来，实际上它只是一个打印控制符，告诉系统在打印完字符串后回车换行。在 printf 中使用\n 或不使用\n，其效果等效于在 Pascal 程序中选择

writeln 语句或 write 语句。

在 C 语言中，所有程序模块（包括主程序）都被当做函数来对待，每个函数的定义都是从函数名及紧跟在函数名后面的圆括号开始。圆括号中存放要通过函数传递的参数，例如 printf("This is C programming language")，其中"This is C programming language"是传递给 printf 的参数。main 模块也不例外。这里 main 后面的一对圆括号表明 main 函数没有参数传递。

所有 C 程序的语句都必须用分号结尾，因此分号紧跟在 printf 语句后面出现。

例 1.2 计算 50+20 的值。

```
#include <stdio.h>
main()
 {
    int x;
    x=50+20;
    printf("The sum of 50 and 20 is%d\n",x);
 }
```

程序的输出结果是：

The sum of 50 and 20 is 70

程序用了一个变量 x 来表示两个整数 50 加 20 的和，和 Pascal 等高级语言一样，C 程序中所有的变量也都必须先定义后使用，定义的内容包括变量名和变量的数据类型。变量可以有不同的数据类型，int x 说明 x 是一个整型变量。

x=50+20 是一个赋值语句，它把 50+20 的值赋给变量 x。

从程序的输出结果可知，符号%d 不是直接显示在屏幕上的两个字符，它们是 printf 函数的打印格式控制参数，它告诉系统按整数格式显示变量 x 的值。

程序第二行的 #include <stdio.h>是 C 语言的预处理控制行，所有的预处理控制行都由字符 # 打头。这些控制行传送某些信息给编译器，告诉编译器在对程序进行编译以前所需要进行的预处理工作。这里，#include <stdio.h>要求编译器把一个文件 stdio.h 加到程序中去。

stdio.h 是一个标准输入输出头部文件，它的内容是编译器在处理你的程序中的输入输出函数时所需要的信息。这个头部文件并不是一定要包含在你的程序中，在一般情况下，如果你没有给出 #include <stdio.h>而程序中又调用了输入输出函数，则编译器会自动把 stdio.h 文件包含在你的程序中。

例 1.3 计算任意两个整数的乘积。

```
main()
  {
    int i,j,k;
    printf("Please input the two integers:i and j\n");
    scanf("%d %d",&i,&j);      /* This is an input statement */
    k=i*j;
    printf("The result of i*j is%d\n",k);
  }
```

程序的执行结果是：

(1)屏幕显示：

Please input the two integers:i and j

(2)键入两个整数：

4　5

(3)计算结果：

The result of i * j is 20

程序的第三行定义了三个整型变量i,j和k，第四行的printf的作用是在执行输入功能以前，在屏幕上显示出：Please input the two integers: i and j,以提示使用者程序对他的要求，当使用者读到这条屏幕信息时，他就可以开始键入数据。

程序中的scanf语句是一个输入函数，它从键盘上读入两个整数i和j，scanf中双引号的内容是输入格式说明，它表明要读入的是两个整型变量的值。&i和&j分别是变量i和j的地址。这种输入方式是C和其它语言的不同处之一，它的所有变量的输入都必须指明变量的地址而不是变量本身。

程序第五行的后面一部分/*.....*/称作注释。C语言中把从/*开头到*/结尾的部分作为注释部分，注释可以加在程序的每一个语句后面或程序开头。C的编译程序在编译过程中自动跳过注释部分，不对它作任何处理。给程序加上注释可以用来对程序的功能加以注解，便于提醒编程者、加深理解和记忆。

通过阅读以上几个程序，我们可以了解到C语言程序的几个基本部分为：

```
main()
  {
      变量定义
      程序语句
  }
```

变量定义部分定义所有需要的变量，程序语句包括C的各种基本语句、控制语句及函数调用。由于C语言中的所有程序模块都是按函数来处理的，故任一函数的结构也应是这种方式。

§1.3　保留字和标识符

和其它计算机语言一样，C语言也采用了若干保留字和标识符来描述各种语句和运算的功能。保留字是语言本身最基本的词汇，主要用来描述C的语句和定义各种数据类型；标识符是用户自己定义的单词，用来为各种变量名、常量名和函数名命名。

C的保留字有以下32个：

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

标识符的命名规则是：每个标识符可以由1～32个字符组成，这些字符可以是字母，数字或下划线，但第一个字符必须是字母或下划线。在为标识符命名时，必须记住C语言对大小写字母要加以区别，因此对于C的编译器来说，COUNT和count是两个不同的标识符。另外，自定义的标识符不能与C语言的保留字重名，但保留字可以作为标识符名的一部分出现在标识符中。

以下是一些合法的标识符的例子：

number_one	WriteData	_Max
sum1_3	All_Consts_from_1_to_100	k23
first_int	if_error_goto	

以下是一些不合法的标识符的例子：

%sd	$_dollar	1add
FIRST&SECOND	test.	'a+d'
continue	do	

一个良好的编程习惯是：当你在为你的变量命名时，应尽量选择与该变量的功能相接近的词汇来表示该变量，如用result来表示最后的计算结果，用first_value来表示第一个输入值等等。这种方式在变量名较多时便于记忆各变量，避免出错，也使程序阅读起来更加清楚。

附录1.1　C程序示例

参看图1.1所示的电路，假定已知各电阻R1，R2，R3，R4和电压V的值，求流过电阻R1的电流值I。

解：

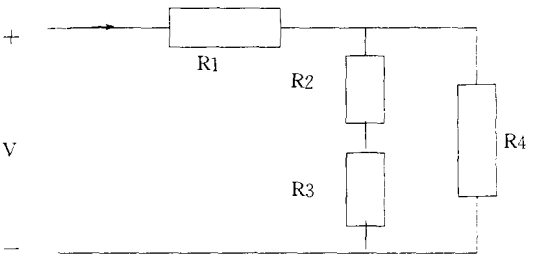

图　1.1

根据电路学知识可知，I的计算公式为：

$$I=\frac{V}{R1+\dfrac{(R2+R3)*R4}{R2+R3+R4}}$$

我们编写一个C程序来完成上述计算。该程序首先读入R1，R2，R3，R4和V的值，然后按公式计算电流I并打印输出结果。

```
/* This program caculates the current value */
main()
  {
    float r1,r2,r3,r4,i,v;              /* variable definition */

    printf("Please input R1,...R4 in order\n");
    scanf("%f %f %f %f",&r1,&r2,&r3,&r4);       /* input data */
    printf("Now,please input V\n");
    scanf("%f",&v);
    i=v/(r1+(r2+r3)*r4/(r2+r3+r4));             /* caculate I */
    printf("The current value is I=%f\n",i);
```

}

程序的第三行定义了六个浮点型变量，浮点型是C的另一种数据类型，它用来实现对实数的存储和处理。保留字float用来说明浮点型变量。

第五行和第七行的scanf中的格式控制说明符不再是%d，而变成了%f，它们说明要输入的数据是浮点型的数据。第九行的printf中的格式控制说明符也为%f，它说明要把变量i按浮点格式输出。

附录1.2 C的编程风格

良好的编程风格有助于对程序的阅读、理解和记忆，在学习C语言的过程中，要注意养成正确和良好的习惯。用C语言来编写程序时，应该遵循以下的一些原则：

1. 大小写字母用在不同的场合，通常程序中的变量名、函数名等都采用小写字母，而所有的符号名和常数名用大写字母来定义。例如：

用　x　value　data_100　等表示变量名

用　MAX_INT　PI　LAST_DATA　等表示常量名

2. 把程序中的各语句组根据它们的功能和嵌套关系缩进编排，使程序的模块和复合关系都变得十分明了。比较以下两个程序，你就可以知道缩进编排的方法和好处了

(1)采用缩进编排方法

```
main()
  {
    int i,sum,k,l;
    scanf("%d",&l);
    if (l<=10)
        {  sum=0;
           k=1;
           for(i=1;i<=10;i++)
             {
               sum=sum+k;
               k=k*i;
             }
       }
    else  if  (l<=20)
       {

           sum=0;
           k=1;
           for(i=11;i<=20;i++)
             {
               sum=sum+k;
               k=k*i;
```

```
            }
        }
    else     sum=0;
    printf("The result is sum=%d\n",sum);
}
```

(2)不采用缩进编排方法

```
main()
{int i,sum,k,l;
scanf("%d",&l);
if (l<=10)
{sum=0;
k=1;
for(i=1;i<=10;i++)
{sum=sum+k;
k=k*i;}}
else if (l<=20)
{sum=0;
k=1;
for(i=11;i<=20;i++)
{sum=sum+k;
k=k*i;}}
else     sum=0;
printf("The result is sum=%d\n",sum);
```

显然,即使我们还不知道程序 1 中某些语句的意义,我们也能够从它的结构和英语句子中猜到它的功能。但如果把它写成程序 2 的形式,要读懂它就困难得多了。

3. 前面已经提到,在定义变量名的时侯应该注意把它们的功能和名字联系起来,这种方式明显地提高了程序的可读性。在实际应用中还应该注意到,虽然C语言规定变量名最长可以有 31 个字符,但在某些机器上只有前 8 个字符才有意义,即对这些机器来讲,FinalData1 和 FinalData2 是同一个变量。因此,为了保证程序的可移植性,变量名不宜取得太长,且变量名的前几个字符不要完全一样,例如上面两个标识符可以改写为:Data1Final 和 Data2Final。

4. 充分利用注释语句,在编写一个程序时,就把关于这个程序的目的、程序的使用方法、变量名的意义、各函数的作用以及一段复杂语句完成的功能等信息以注释的方式加到程序中。各注释的书写位置也应该统一起来,例如可以都放在语句、变量、函数的使用以前,或都放在语句行的后面。

5. 在程序中加上适当的空格或空行使程序更加清晰。一般在变量的定义与语句之间留一个空行,在不同的程序块之间也留出一个空行。

6. 用一对花括号{ 和 }标志一个程序功能块的开始和结束。C 语言对{ 和 }的书写位置没有做规定,但为了方便起见,一般总是让{ 和 }按列对齐。

以下是按上述要求书写的程序例子:

```
/* This program finds the row number and the collom number of the max value and
```

```
    min value in the array */
main()
  {
    int m[3][4],i,j,max_value,min_value,max_row,max_col,min_row,min_col;

    /* input array data */
      for(i=0;i<3;i++)
        for(j=0;j<4;j++)
         scanf("%d",&m[i][j]);

    /* find the max value and min value */
    max_value=-32767;
    min_value=32767;
    for(i=0;i<3;i++)
      for(j=0;j<4;j++)
        {
          if(m[i][j]>max_value)
            {
              max_value=m[i][j];
              max_row=i;
              max_col=j;
            }

          if(m[i][j]<min_value)
              {
                min_value=m[i][j];
                min_row=i;
                min_col=j;
              }
          }

  /* output the result */
  printf("The max value %d is located in row %d and col %d\n",max_value,max_row,
  max_col);
  printf("The min value %d is located in row %d and col %d\n",min_value,min_row,
  min_col);
}
```

问题与习题

1. 以下几个语句的显示结果是什么？

```
printf("first second third");
printf("first second third\n");
printf("first\nsecond\nthird\n");
printf("first\n\nsecond\n\nthird\n");
```

2. 以下程序完成的功能是什么？

```
main()
  {
    int i,j,k;
    printf("Please input the two integers:i and j\n");
    scanf("%d,%d",&i,&j);

    k=i*i+j*j;
    prinft("The result of i*i+j*j is%d\n",k);
  }
```

3. 以下哪些标识符是合法的？

int_do，　int，　do，　32data，

DWMax_Value，one_$，TWO_AND_THREE

4. 试编写一个C程序，它输入一个浮点数，计算它的倒数并将结果输出。

5. 试编写一个C程序，它输入三个整数，计算它们的和并将结果输出。