# C程序设计

李振立　张慧萍　主编

科学出版社

## 内 容 简 介

本书是理工类非计算机专业程序设计课程教材，主要用于培养学生利用计算机处理问题的能力，使学生掌握程序设计的基本方法，能够利用C语言进行简单的程序设计，为进一步学习和应用计算机知识打下良好基础。书中内容包括C语言程序设计基础，基本数据类型、语法规则、控制语句、数组与函数、指针、输入输出和文件处理，以及软件基础知识。

本书适用于理工科非计算机专业学生，也可供计算机编程爱好者参考使用。

**图书在版编目(CIP)数据**

C程序设计/李振立，张慧萍主编. —北京：科学出版社，2011.9

ISBN 978-7-03-032262-3

Ⅰ.C… Ⅱ.①李…②张… 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(2011)第178606号

责任编辑：张颖兵/责任校对：梅 莹

责任印制：彭 超/封面设计：苏 波

科学出版社 出版

北京东黄城根北街16号

邮政编码：100717

http://www.sciencep.com

武汉市首壹印务有限公司印刷

科学出版社发行 各地新华书店经销

*

2011年9月第 一 版 开本：787×1092 1/16

2011年9月第一次印刷 印张：19 3/4

印数：1—5 000 字数：465 000

**定价：33.00元**

（如有印装质量问题，我社负责调换）

# 前　言

以计算思维能力培养为新目标、新任务的计算机基础教学，受到国家教育主管部门的高度重视和大力支持。计算思维是运用计算机科学的基础概念进行问题求解、系统设计，以及人类行为理解的涵盖计算机科学之广度的一系列思维活动。教育部高等学校计算机基础课程教学指导委员会确定以培养学生计算机应用能力和计算思维能力为目标，大力培养学生计算思维能力，提高学生综合素质和创新能力，提高计算机课程教学质量。

思维内容是通过语言表述的，计算思维的内容也要通过程序设计语言来表述。对理工科大学生而言，学习程序设计语言是学习计算思维最好的工具，而C语言则是程序设计语言的首选。本教材《C程序设计》为学生提供较为全面的程序设计思想、程序分析过程和解题方法，提供分析和解决问题的基本过程和思路，能够体现语言层面上的问题求解方法，是训练学生计算思维的有效工具，培养计算思维能力的教学内容。

C语言具有高级语言的诸多特点，又具有汇编语言的特点；既适合用于开发操作系统和系统使用程序，又适合用于编写嵌入式系统等硬件系统的开发程序。C语言应用范围广，数据处理能力强，操作简单，易读性好，是最好的入门教材。

《C程序设计》系统介绍了计算机语言的词法、语法、语言规则、数据类型、数据存储、算法结构、函数模块、地址指针、数据文件等基本概念，深入地讨论了C语言程序设计中数据结构和经典算法，用C语言分析计算机语言模型，学习解题方法，培养计算思维能力。

本书根据理工类计算机基础课程教学指导分委员会颁布的《计算机基础课程教学基本要求》，从“算法基础与程序设计”领域中选择如下的知识单元，包括程序与程序设计语言、数据类型基础、基本控制结构、基本算法概念、程序设计过程、过程与函数、构造类型与指针、文件等知识单元组织教学内容。全书分为11章，内容包括C语言概述、数据类型与表达式、顺序结构程序设计、选择结构程序设计、循环结构程序设计、数组、函数、指针、结构体与共用体、文件和软件基础知识(选讲)。本书涉及C语言程序设计的全部内容和软件基础知识的主要内容。

本书由长期工作在教学一线的教师编写，全书各知识单元编排顺序得当，结构合理而严谨，内容丰富，由浅入深、循序渐进，详略度把握得体，书中配置了大量运行在VC环境下的例题，是一套理想的C语言程序设计的教材。本书即可作为各类高等院校本、专科非计算机专业的C语言程序设计的教材，也可以作为独立学院、高职高专、网络学院的教材，对于社会计算机学习者，尤其是具有一定计算机基础而又欲获得提高的广大计算机爱好者，本书无疑是一本极好的自学读物。

本书由李振立、张慧萍主编，其他编委还有张胜利、楚维善、高金兰、郭盛刚、顾梦霞、陈小娟、贺红艳、吴佩、罗宏芳、周雪琴、王世畅、李珺。在全书的策划、编写、出版过程中，王学文等同志给予了大力支持，在此深表谢意。

编　者

2011年8月8日

# 目　录

# 第1章 C语言概述

C语言是广泛使用的计算机程序设计语言。C语言具有一般高级语言的特性，程序不依赖计算机硬件，可读性和可移植性好，接近于自然语言或数学语言；又具有低级语言的特性，可以内嵌汇编指令，将汇编指令作为C语言的指令，可以直接对计算机硬件进行操作，如对内存地址的操作、位操作、I/O操作等。C语言集高级语言和低级语言的优点于一身，适用于作为系统描述语言，用于编写大型的操作系统、编译系统、应用软件，也可以作为DSP，EDA，ARM，单片机等嵌入式系统的开发语言。

C语言属于面向过程的程序设计语言，面向过程是一种以事件为中心的编程思想，将事件的产生、发展、变化和结果等事件运作过程作为研究的重点，采用模块化的方法设计源程序，由主控模块分级调用各子模块，各个模块依照事件运作的逻辑次序组织程序流程，用程序流程图描述程序的算法。C语言将要处理的信息数字化，表示成各种类型的数据。数据的类型、数据的组织和数据的传递合称为程序的数据结构，数据结构也是程序设计的重要内容，因此，可以说，面向过程的程序是由算法和数据结构共同组成的，即面向过程的程序＝算法＋数据结构。

## 1.1 C语言的发展史

### 1.1.1 C语言的起源

C语言是在B语言的基础上发展起来的。1970年美国贝尔实验室的学者肯·汤普森(Ken Thompson)以BCPL(basic combined programming language)语言为基础，经过简化，设计出简单而且接近于硬件的B语言，并用B语言写了第一个UNIX操作系统，在PDP-7上实现。1972年，贝尔实验室的学者D. M. Ritchie在B语言的基础上设计了C语言。1973年美国贝尔实验室的肯·汤普森和丹尼斯·利奇两人合作改写UNIX(第五版)，其中90%是用C改写的。1977年，出现不依赖于具体计算机的C语言编译文本《可移植C语言编译程序》，使C语言迅速移植到各类计算机上，从而推动了UNIX操作系统在各类计算机上的开发。UNIX的日益推广，推动了C语言的广泛应用。1978年由Brian W. Kernighan和D. M. Ritchie两人以UNIX(第七版)C语言为基础，合作发表了"The C Programming Language"这部名著，通常称为"K&R"，成为早期C语言的事实标准，称为经典C语言。1983年美国国家标准协会(ANSI)对C语言进行扩充和规范制定了新的标准，称为ANSI C。1987年公布了C语言的美国国家标准，1989年ISO/IEC提出了国际标准草案，1990年公布了C语言的正式标准称为C89，有时也称为C90。C89是目前广泛使用的标准，所有的主流编译器都支持C89。

对C90的修订工作开始于1994年，ISO/IEC于1999年12月16日，推出了C语言

标准：ISO/IEC 9899：1999(Programming languages-C)，称为 C99。C99 保持了 C 语言的基本特性，修订目标有三点：①支持国际化编程，引入了支持国际字符集 Unicode 的数据类型和库函数；②修正原有版本的明显缺点，如整数的移植方法，例如 int8_t、int16_t、int32_t；③针对科学和工程的需要，改进计算的实用性，添加了复数类型和新数学函数等。完全支持 C99 标准的编译器较少，使用不广泛。

### 1.1.2 C 语言的特点

(1) 语言简洁、紧凑，使用灵活、方便。语法限制不太严格，语言描述简单、明晰，没有复杂的控制结构，程序设计自由度大。

(2) 运算符丰富，表达能力强。C 语言将括号、赋值、强制类型转换等都作为运算符处理，使运算符的范围更广泛，表达式的类型多样化，运算更灵活方便，增强了语言的表达能力。

(3) 数据结构丰富，结构化程度高。它具有结构化程序的典型特征，程序代码与数据分离，具有现代语言的各种数据结构。C 语言的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型、枚举类型和自定义类型等。能实现链表、树、栈等复杂数据结构。具有结构化的流程控制语句，编制的程序具有模块化和结构化的特点。

(4) 兼有高级语言和低级语言的特点，允许直接访问物理地址，能进行位操作，能实现汇编语言的大部分功能。

(5) 生成目标代码质量高，程序执行效率高，一般高级语言程序执行效率低，C 语言的代码效率仅次于汇编语言，代码效率达到汇编语言的 80%～90%。

(6) 比汇编程序的可移植性好。程序基本上不做修改就能用于各种型号的计算机或各种操作系统。

### 1.1.3 C 语言的集成开发环境

C 语言是 AT&T 贝尔实验室发明的程序设计语言，贝尔实验室发布的集成开发环境 Dev-C++是一款自由软件，遵守 GPL 协议，用户在遵守 GNU 协议的前提下可以从 devpak.org 上取得最新版本的各种工具包，获取源代码，并拥有对这一切工具自由使用的权利。Dev-C++集合了 GCC，MinGW32 等众多自由软件，是一款非常实用的 C 开发工具，很多高水平的软件开发人员用它编写出许多著名软件。在全世界自由软件爱好者的研究和开发下，Dev-C++仍在不断地发展和进步。

常用的 C 与 C++集成开发环境有 Borland 公司的 Turbo C，Borland C++，C++ Builder；Microsoft 公司的 Microsoft C，Visual C++，Visual studio.NET；多平台的 C++集成开发环境有 GCC，MinGW，QT，eclipse+CDT 等软件。在 2008 年以前，全国计算机等级考试二级 C 语言使用的集成开发环境是 Turbo C(简称 TC)，初学者开始入门时使用简单的 TC 2.0 集成开发环境，有助于对 C 语言的操作和理解。2008 年以后，全国计算机等级考试使用的集成开发环境是 Visual C++6.0。在等级考试的带动下，Visual C++(简称 VC)成为最常用的 C 语言集成开发环境，但 VC 不支持 C99 标准。Dev-C++支持 C99 标准，是全国奥林匹克编程 C 语言指定的开发工具。

C语言集成开发环境都是由编辑器、编译器、连接器集合而成，用户编写的源程序通过编辑器输入到计算机内存，形成扩展名为c的程序文件，通过编译器编译，形成扩展名为obj的目标代码文件，通过连接器连接目标代码和库文件，形成扩展名为exe的执行文件，运行执行文件，在运行窗口输入数据，并输出计算结果。常用的集成开发环境TC与GCC是以命令行方式工作，VC和Dev-C++是在Windows平台下工作，由于Dev-C++执行完程序后立即关闭了输出窗口，用户看不见输出结果，因此在源程序结束之前应该加入"getch();"语句，暂停程序运行，让用户查看完输出内容后，按任一键结束。也可以用"system("pause");"语句，调用标准库中的系统函数，使用system函数时，要在源程序开始处用#include <stdlib.h>声明包含标准库头文件，因为system函数是定义在<stdlib.h>这个头文件之中。

## 1.2 C语言的程序架构

### 1.2.1 C语言程序的基本架构

C语言是一种函数型语言，C语言的源程序是由一系列函数构成的。C语言的函数模块一般形式如下：

```
编译预处理命令
函数类型 函数名(函数形式参数)
{
  声明语句
  执行语句
}
```

下面通过几个简单的例子讨论C程序的架构。

**例1.1** 在Dev-C++集成环境下编程，建立一个文件名为ex1_1的C源程序，在屏幕上输出"Compiler DEV C++"后回车换行(文件名为ex1_1.c)。

```
//注释内容:文件名 ex1_1.c
#include <stdio.h>                  /*包含标准输入/输出头文件*/
#include <stdlib.h>                 /*包含C语言标准库头文件*/
int main(void)                      /*主函数由主函数名和参数组成*/
{                                   /*函数体开始*/
    printf("Compiler DEV C++\n");   /*用格式输出函数printf输出字符串*/
    system("pause");                /*暂停程序运行,查看输出内容*/
    return 0;                       /*向操作系统返回0值,即程序正常退出*/
}                                   /*函数体结束*/
```

用Dev-C++集成环境编辑、调试该程序，根据C99标准，第1行可以使用"//"作为行注释；第7行要有"system("pause");"语句，确保程序运行结束前，暂停程序运行，查看输出内容。按快捷键F9键，集成环境编译、连接、运行该程序，在输出屏幕上显示字符串

"Compiler DEV C++",按任一键结束,关闭输出窗口。用TC或VC集成开发环境则应该将第1行的行注释改为块注释"/*注释内容:文件名ex1_1.c*/",去掉第7行"system("pause");"语句和第3行包含C语言标准库头文件。

**例1.2** 在VC集成环境下编程,编制输入两个整数,并输出较大的一个数的源程序(文件名为ex1_2.c)。

```
/*从两个数中找出较大的数,文件名为 ex1_2.c*/
#include<stdio.h>                    /*包含标准输入/输出头文件*/
main()                               /*主函数参数无返回值*/
{                                    /*函数体开始*/
   int iX,iY,iZ;                     /*定义函数中使用的变量 ix,iy,iz*/
   scanf("%d,%d",&iX,&iY);           /*用格式输入函数输入两个整数*/
   if(iX>iY)iZ=iX;                   /*如果 ix>iy 那么把 ix 的值赋给 iz*/
   else iZ=iY;                       /*否则把 iy 的值赋给 iz*/
   printf("max=%d\n",iZ);            /*用格式输出函数 printf 输出最大值*/
}                                    /*函数体结束*/
```

在VC集成环境中编辑该程序,选择"组建"菜单中"组建"命令,或按F7键,编译源形成目标模块,并组建成ex1_2.exe文件;选择"组建"菜单中"执行"命令,或按Ctrl+F5组合键,执行该程序打开输出窗口,用户输入"24,32"回车,屏幕显示"max=32"。

**例1.3** 先定义一个求两个数的较大值函数,然后由主函数调用该函数,试编写源程序,当输入两个整数时,输出较大的一个数(文件名为ex1_3.c)。

```
#include <stdio.h>                   /*包含标准输入/输出头文件*/
#include <stdlib.h>                  /*包含 C 语言标准库头文件*/
int iMax(int iX,int iY);             /*函数 iMax 前向说明*/

int main(void)                       /*主函数函数无返回值,参数无返回值*/
{                                    /*函数体开始*/
int iA,iB,iC;                        /*定义函数中使用的变量 iA,iB,iC*/
scanf("%d,%d",&iA,&iB);              /*用格式输入函数输入两个整数*/
iC=iMax(iA,iB);                      /*调用函数 iMax*/
printf("max=%d\n",iC);               /*用格式输出函数 printf 输出最大值*/
system("pause");                     /*暂停程序运行,查看输出内容*/
return 0;                            /*向操作系统返回 0 值,即程序正常退出*/
}                                    /*函数体结束*/

int iMax(int iX,int iY)              /*用户定义函数 iMax*/
{                                    /*函数体开始*/
int iZ;                              /*定义函数中使用的变量 iX,iY,iZ*/
if(iX>iY)iZ=iX;                      /*如果 iX>iY 那么把 iX 的值赋给 iZ*/
```

```
else iZ=iY;                    /*否则把 iy 的值赋给 iz*/
return(iZ);                    /*返回计算结果*/
}                              /*函数体结束*/
```

该程序的功能与程序例 1.2 相同，通过调用函数来计算两个数的较大值。

对照以上三个简单的 C 程序，可以看出 C 程序的基本架构由编译预处理命令、参数说明、函数说明、主函数、函数等成分组成。

### 1. 编译预处理

编译预处理命令有文件包含(.h 头文件)、宏定义和条件编译三种。在 C 语言中经常使用的包含文件有标准输入/输出头文件 stdio.h、标准库头文件 stdlib.h、字符串头文件 string.h、控制台输入/输出头文件 conio.h、数学函数库头文件 math.h 等。

包含文件的格式为

```
#include <filename.h>          例:#include <stdio.h>
```

或

```
#include "filename.h"          例:#include "stdlib.h"
```

用#include <filename.h>格式来引用标准库的头文件，编译器从标准库目录开始搜索。用#include "filename.h"格式来引用非标准库的头文件，编译器从用户工作目录开始搜索。

宏定义的作用是定义符号常量，用一个短的名字代表一个长的字符串。定义格式为

```
#define 标识符 字符串          例:#define PI 3.1415926
```

### 2. C 语言中的函数

C 语言是函数型语言，程序模块都是函数型模块，一个 C 程序可以由一个或多个具有相对独立功能的函数模块构成，其中有一个且仅有一个以 main 命名的主函数，其他函数模块为函数如 iMax。程序从 main 函数开始执行，直到执行完函数体程序才运行结束。主函数调用函数，执行完函数体后返回到主函数调用处继续执行后续语句。函数与函数之间用参数传递数据。函数只能被主函数或其他函数调用，不能单独运行。

### 3. 函数的组成

一个函数由函数头部和函数体两部分组成，函数的头部包括函数的类型、函数名和形式参数表。形式参数表可以无类型，参数表为无类型用类型名“void”表示。函数体用一对花括号“{ }”括起来，包括局部说明部分和语句部分。一般形式为

```
函数类型 函数名(参数表列)          int iMax(int iX,int iY)
{                                  {
局部说明语句;                      int iZ;
                                   if(iX>iY)iZ=iX;
执行语句;                          else iZ=iY;
                                   return(iZ);
}                                  }
```

### 4. 函数的调用

C函数分为标准函数和用户定义函数两类，程序模块中的函数调用语句可以调用标准函数，也可以调用用户定义函数。标准函数是由编译程序提供的，标准函数的定义是以编译后的目标代码形式存放在系统的函数库中，称为C函数库。用户编程时直接调用标准函数，在例1.2中的格式输入函数scanf和格式输出函数printf都是标准函数。格式输入函数scanf的格式为

```
scanf("输入格式",输入项地址表列);
```

其中："输入格式"是用双引号括起来的字符串，包括格式说明和普通字符，格式说明是由%字符开头后接格式字符，如%d表示十进制整型数，%f表示浮点型数，%c表示单个字符等；普通字符是按照需要输出的字符和控制符，包括可打印的字符和转义字符两种；"输入项地址表列"是由若干个地址所组成的表列，可以是变量的地址如&iA，&iB，或字符串的首地址str1。

格式输出函数printf的格式为

```
printf("输出格式",输出表列);
```

其中："输出格式"是用双引号括起来的字符串，包括格式说明和普通字符；输出表列是输出的常量、变量或表达式，用逗号分隔。

用户定义函数必须由用户在源程序中编写函数定义，根据模块功能，设计和编写用户定义的函数语句，供其他函数调用。用户定义函数与调用函数的函数名必须一致，两者参数的个数与参数的类型必须按位置相同放置，函数定义的参数为虚参，函数调用中的参数为实参，参数的传递按位置虚实对应。在例1.3中调用函数iMax的语句是赋值语句：

```
iC=iMax(iA,iB);     /*调用 iMax 函数,iA,iB 为实参已有确定的数值*/
```

其中："="为赋值号，执行该语句，将函数值赋给变量iC，保存在变量iC指向内存单元。

### 5. 函数说明

函数应该满足先定义后调用的规则，如果函数的定义在前，在主函数中对函数的调用语句在后，则满足先定义后调用的规则，不需要进行函数说明；如果主函数在前，函数的定义在后，则需要对函数进行前向说明。函数说明是在函数被使用之前的位置，用函数原型对函数进行前向说明，将函数的名称、类型和参数类型告知编译器。例1.3中的第3行

```
int iMax(int iX,int iY);
```

语句为函数原型，通知编译程序调用函数的类型及参数的类型，语句中的iX、iY可以缺省，例如写成

```
int iMax(int,int);
```

### 6. 参数说明

在程序的前面可以说明全局变量，在例1.3中的第4行可以用"int iD;"语句说明全局变量iD。在程序模块的说明部分可以说明局部参数，如例1.3中第7行"int iA,iB,iC;"语句和第17行"int iZ;"语句。在函数定义的参数表列中说明函数传递的参数，如例1.3中第

15行的 int iMax(int iX,int iY)语句中的 int iX,int iY。

### 7. 书写格式

C语言的书写格式自由,C语句用分号";"作结束符,一条语句可以写成多行,多条语句也可以写成一行。C语言的书写支持缩进格式,将同一层次的语句,语句的开头对齐在同一列,每下一个层次,语句开头通过加空格缩进若干列,从而形成层层缩进对齐的书写格式。

### 8. 注释

程序中可加必要的注释,注释分为块注释和行注释,块注释用一对符号"/* */"作程序中注释的定界符,表示"/*"和"*/"之间的内容是注释;行注释使用"//",支持C99的编译器支持在C语言中使用行注释。注释的内容不影响程序的编译和执行,增加程序的可读性,是一种良好的程序设计风格。

## 1.2.2 C语言程序逻辑顺序

C语言程序模块中,语句块的逻辑顺序是按局部说明、数据输入、数据处理、数据输出的次序依次排列的。

**例1.4** 已知圆的半径为 $r$,编写求圆的面积与周长的程序(文件名为 ex1_4.c)。

```
#include <stdio.h>                    /*包含标准输入/输出头文件*/
#define PI 3.1415926                  /*宏定义*/
main()                                /*经典C表示的主函数*/
{                                     /*函数体开始*/
float fR,fS,fW;                       /*定义函数中使用的变量 fR(局部说明)*/
scanf("%f",&fR);                      /*用格式输入函数输入半径 dR*/
fS=PI*fR*fR;                          /*求圆的面积(数据处理)*/
fW=2*PI*fR;                           /*求圆的周长(数据处理)*/
printf("fS=%f,fW=%f\n",fS,fW);        /*用格式输出函数 printf 输出面积和周长*/
}                                     /*函数体结束*/
```

说明:

(1) 程序开头的编译预处理命令由文件包含 #include <stdio.h>和宏定义 #define PI 3.1415926两部分组成,文件包含和宏定义是命令不是语句,命令结束没有分号,宏定义不是给PI赋值,而是定义符号常量,C程序在编译时会将非双引号内的所有PI都置换成3.1415926;

(2) 函数体内的局部说明"float fR,fS,fW;"说明变量fR,fS,fW是浮点型变量,在内存中为每个变量分配4个字节的存储单元;

(3) 在C语言中没有专门的输入语句,用格式输入函数"scanf("%f",&fR);"输入数据;

(4) 求圆的面积“fs=PI * fR * fR;”与求圆的周长“fW=2 * PI * fR;”是处理数据的语句块;

(5) 在C语言中没有专门的输出语句,用标准函数“printf("fS=%f,fW=%f\n",fS,fW);”输出数据。

## 1.2.3 C语言的风格

C语言的风格指C程序的风范格局,是程序内容与书写形式相互统一的表现形式,是由程序员的个性特征与其所受的教育、使用教材、编程经历、团队的规定等诸多条件影响下形成的编程习惯。C语言的风格是编程思想的开放性、程序开发的统一性、程序结构的层次性和程序内容的易读性的表现形式。也是程序员对程序书写形式一贯性的体现。

每一个软件开发团队可以自行规定表征C语言风格的书写形式,内容主要包括大括号的放置位置、缩进格式、空格的使用、注释方法、命名系统、函数等。

### 1. 主函数与大括号的放置位置

编写C语言源程序,主函数头部声明与大括号的放置位置要规范,以输出Hello!的源程序为例,经典C语言程序如下:

```
main()                      /*经典C语言声明的主函数*/
{                           /*函数体开始,标准C提倡单独占一行*/
printf("Hello!");           /*用标准输出函数printf输出字符串*/
}                           /*函数体结束,单独放在一行*/
```

经典C语言声明的主函数可以缺省函数类型,可以缺省参数。主函数中的大括号“{”表示函数开始,应该放在main()的下一行的第1列,其他语句另起一行;反大括号“}”也单独放在一行。C89或C99声明的主函数main的函数类型为整型,程序结束前要有return 0;语句,表示向操作系统返回0值,即程序正常退出。程序如下:

```
#include <stdio.h>              /*包含标准输入/输出头文件*/
int main(void)                  /*C89或C99表示的主函数*/
{                               /*函数体开始,提倡单独占一行*/
  printf("Hello!");             /*用标准输出函数printf输出字符串*/
  return 0;                     /*向操作系统返回0值,即程序正常退出*/
}                               /*函数体结束,单独放在一行*/
```

函数中的大括号也应该单独占一行。例如:

```
int iEX1(int iX,int iY)         /*函数*/
{                               /*函数体开始,单独占一行*/
  iX=iX+iY;iY=iX-iY;iX=iX-iY;/*函数处理多条语句可以写在一行*/
  printf("iX=%d,iY=%d",iX,iY);/*用标准输出函数printf输出整数iX,iY*/
}                               /*函数体结束,单独放在一行*/
```

不提倡将函数体开始的“{”与变量的类型说明写在一行。例如:

```
main()
{ int iA;
……
```

也不要用 void main(void)声明主函数，这是非标准的书写方法。

### 2. 缩进格式

在C的集成开发环境中，按一次 Tab 键默认跳过 8 个字符，C 语言默认每层控制结构缩进深度为 8 个字符。虽然有的用户设置的缩进深度为 4 个字符或 2 个字符，但这不是一个好的风格。因为控制结构的每一层构成一个语句块，块的开始和结束是缩进对齐的，缩进量设置为 8 个更加醒目，当用户连续编程 10 几个小时后(程序员经常这样)，缩进深度的大小就非常重要了。能帮助用户更清楚地划分结构层次，理解程序内容。同时，当用户程序中的控制结构层次太多，程序代码缩进到右边，看起来不舒服，也能起到提醒程序员修改程序的作用，嵌套这样多层的控制结构理解起来已经很困难了。

### 3. 空格

在C语言程序中使用空格能形成良好的程序风格，有些空格是作为语句的分隔符，在语句中是必须存在，例如，在 const，virtual，inline，case 等关键字之后至少要留一个或一个以上的空格，否则无法辨析关键字。另一类关键词如 if，for，while 之后应留一个空格再跟左括号“(”，以突出关键字，这类关键词添加的空格不是必须的。有些语言的集成开发环境会自动地给程序中的单词加空格，形成良好的程序风格。

良好程序风格使用的空格有如下规律：

(1) 函数名之后不要留空格，紧跟左括号“(”，关键字与“(”之间加空格，以体现两者的区别。在“(”之后紧跟数据。

(2) 符号“)”、“,”、“;”向前紧跟数据，紧跟处不留空格。“,”之后要留空格；如果“;”不是一行的结束符号，其后要留空格。

(3) 单目运算符(一元操作符)如“!”、“～”、“＋＋”、“－－”、“&”(取地址运算符)等前后不加空格。

(4) 双目运算符包括赋值与复合赋值运算符，如“＝”、“＋＝”、“－＝”、“＊＝”、“/＝”；比较操作符如“＞”、“＜”、“＞＝”、“＜＝”、“＝＝”、“！＝”；算术操作符如“＋”、“－”、“＊”、“/”、“%”；逻辑操作符如“&&”、“||”；位域操作符如“&”、“|”等，双目运算符的前后应当加空格。由两个字符组成的运算符两个字符之间不要加空格符，如“＋＝”不能写成“＋＝”。

(5) 数组运算符“[ ]”、域运算符“.”、指向运算符“－＞”的前后不加空格。

### 4. 注释

注释通常用于版本、版权声明，函数功能说明，函数的接口说明，结构提示，重要的代码行的说明，注释有助于理解代码，但过多地使用注释会使程序繁复，影响阅读程序代码。注释不能够嵌套。