



普通高等教育“十二五”重点规划教材 公共课系列

C语言程序设计应用教程

许勇 主编



科学出版社

普通高等教育“十二五”重点规划教材
公共课系列

C 语言程序设计应用教程

许 勇 主编

科学出版社

北 京

内 容 简 介

本书讲授 C 语言程序设计的基本思想、方法和解决实际问题的技巧,在体系结构安排上尽可能将概念、知识点与案例相结合进行案例教学,并且每一章都精心设计了实训指导、习题与上机实验(附习题和上机实验答案),突出实用性和操作性。全书共分 9 章,主要内容包括 C 语言概述与算法,基本数据类型,运算符与表达式,程序流程控制,数组和字符串,函数,指针,编译预处理,结构体与共用体,文件。

本书可作为高等院校 C 语言程序设计课程的教材,也可以作为等级考试的辅导教材,以及计算机爱好者自学用书和各类工程技术人员的参考书。

图书在版编目(CIP)数据

C 语言程序设计应用教程/许勇主编. —北京:科学出版社,2011
(普通高等教育“十二五”重点规划教材 公共课系列)
ISBN 978-7-03-030116-1

I. ①C… II. ①许… 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(2011)第 013779 号

责任编辑:赵丽欣/责任校对:刘玉靖

责任印制:吕春珉/封面设计:东方人华平面设计部

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>

百善印刷厂印刷

科学出版社发行 各地新华书店经销

*

2011 年 2 月第 一 版 开本:787×1092 1/16
2011 年 2 月第一次印刷 印张:22 1/2
印数:1—3 000 字数:504 000

定价:34.00 元

(如有印装质量问题,我社负责调换〈百善〉)

销售部电话 010-62142126 编辑部电话 010-62134021

版权所有,侵权必究

举报电话:010-64030229; 010-64034315; 13501151303

前 言

C 语言是一种在世界范围内被普遍采用的优秀的程序设计语言，功能丰富，使用灵活，可移植性好，深受广大用户欢迎。C 语言的数据类型丰富，既具有高级程序设计语言的优点，又具有低级程序设计语言的特点；既可以用来编写系统程序，又可以用来编写应用程序。

本书通过大量实例系统地介绍了 C 语言的语法结构。全书共分 9 章和 2 个附录。第 1 章为 C 语言概述与算法，主要内容包括 C 语言的发展、特点、组成、执行以及基本算法思想。第 2 章为基本数据类型、运算符与表达式，主要内容包括数据类型、数据类型转换、运算符与表达式、位运算、数据的输入与输出。第 3 章介绍了三种结构化程序结构，即顺序结构、选择结构和循环结构，选择结构主要包括 if 条件选择、switch 选择等，循环结构主要包括 while 型循环、do-while 型循环、for 型循环。第 4 章为数组和字符串，主要内容包括一维数组、二维数组、多维数组介绍、字符数组、数组综合实例。第 5 章为函数，主要内容包括函数概念引入、库函数、函数定义、函数间的参数传递、函数调用、数组与函数变量的作用域、变量的存储类型、函数的存储分类。第 6 章为指针，主要内容包括指针的含义、指针变量、指针与数组、指针与函数、指针与字符串、字符数组与字符指针的区别、指向指针的指针、指针类型小结。第 7 章为编译预处理，主要内容包括文件包含、宏定义、条件编译。第 8 章为结构体与共用体，主要内容包括结构体、共用体、枚举类型、类型定义 typedef。第 9 章为文件，主要内容包括 C 文件的概述、文件的打开与关闭、文件的读写、文件的定位与出错检测。

本书组织精练，例题简单，容易理解，突出应用，每章最后一节为“实训指导”，便于读者综合本章所学知识点，提高实际编程能力；同时配备了各种类型的练习和实践操作题，并增加了提示，便于学习掌握和上机练习。对于 C 语言中重要和较难理解、容易出错的内容，书中均特别加上注意和说明。此外，在介绍 C 语言的语法结构的同时，也强调了计算机算法和结构化设计方法的概念和作用。书后附有习题和上机实验的参考答案，可供读者学习时参考。

本书可作为高等学校 C 语言程序设计课程的教材，也可作为计算机等级考试的辅导教材。

本书以 ANSI 标准 C 语言为背景，有关内容不依赖于任何具体的 C 系统，介绍了国内使用较多的 Turbo C 系统和 Visual C++ 环境，程序以 Visual C++ 实现为主，但读者仍可自由选用其他符合 ANSI 标准的 C 系统编程环境作为学习工具。

全书由何光明策划和确定框架结构，最后由许勇统编定稿。孙玉香、杨明、杨萍、赵传审、许娟、史国川、史春联、赵明、陈智、王珊珊、吴涛涛、王程凌等在内容编写、程序测试、文字校对等过程中付出了辛勤劳动，在此一并表示衷心的感谢。

本书配有电子教案，并提供程序源代码，以方便读者自学，请到 www.abook.cn 下载。

限于作者水平，书中难免存在不当之处，恳请广大读者批评指正。如有批评和建议请发至 sciencepress@yeah.net。

目 录

前言

第 1 章 C 语言概述与算法	1
1.1 C 语言的发展史	1
1.2 C 语言的特点	2
1.3 C 程序的组成	3
1.3.1 简单的 C 程序分析	3
1.3.2 简单的 C 程序应用	5
1.3.3 简单的 C 程序种类	6
1.4 C 语言程序上机调试步骤和方法	8
1.4.1 C 语言的编译之 Turbo C 环境	8
1.4.2 C 语言的编译之 VC 环境	11
1.5 程序的灵魂——算法	16
1.5.1 算法的概念及特性	16
1.5.2 算法的表示方法	17
1.5.3 程序设计的基本算法	21
1.6 实训指导	24
习题	26
上机实验	26
第 2 章 基本数据类型、运算符与表达式	27
2.1 数据类型	27
2.1.1 常量与变量	27
2.1.2 整型数据	30
2.1.3 实型数据	32
2.1.4 字符型数据	33
2.2 数据类型转换	35
2.2.1 自动转换与强制转换	35
2.2.2 不同数据类型间的混合运算实例	37
2.3 运算符与表达式	38
2.3.1 算术运算符与算术表达式	39
2.3.2 赋值运算符与赋值表达式	42
2.3.3 逗号运算符与逗号表达式	45
2.3.4 关系运算符与关系表达式	46
2.3.5 逻辑运算符与逻辑表达式	48

2.4	位操作	51
2.4.1	二进制位运算概述	52
2.4.2	位运算符	52
2.4.3	位段	58
2.5	数据的输入与输出	59
2.5.1	格式输出函数——printf 函数	59
2.5.2	格式输入函数——scanf 函数	64
2.5.3	字符输出函数——putchar 函数	68
2.5.4	字符输入函数——getchar 函数	69
2.6	实训指导	70
	习题	70
	上机实验	73
第 3 章	程序流程控制	74
3.1	结构化程序设计的概念	74
3.2	顺序程序设计	75
3.3	选择程序设计	77
3.3.1	if 语句	77
3.3.2	条件运算符	83
3.3.3	switch 语句	85
3.3.4	标号语句和 goto 语句	88
3.3.5	选择结构程序设计举例	89
3.4	循环程序设计	90
3.4.1	while 型循环	90
3.4.2	do-while 型循环	94
3.4.3	for 型循环	96
3.4.4	循环的嵌套	100
3.4.5	其他控制语句	101
3.4.6	循环结构程序设计实例	103
3.5	实训指导	106
	习题	111
	上机实验	117
第 4 章	数组和字符串	119
4.1	一维数组	119
4.1.1	一维数组的定义与引用	119
4.1.2	一维数组的初始化与应用	120
4.2	二维数组	123
4.2.1	二维数组定义与引用	123
4.2.2	二维数组初始化	124
4.2.3	二维数组应用举例	125

4.3	多维数组介绍	127
4.4	字符数组	128
4.4.1	字符数组定义	128
4.4.2	字符数组初始化	129
4.4.3	字符串与字符数组	129
4.4.4	字符数组的输入与输出	130
4.4.5	字符数组处理函数	131
4.5	数组综合实例	136
4.6	实训指导	138
	习题	142
	上机实验	144
第5章	函数	145
5.1	函数概念引入	145
5.2	库函数	146
5.3	函数定义	148
5.3.1	函数定义的形式	148
5.3.2	函数的简单应用	151
5.4	函数间的参数传递	151
5.4.1	形式参数和实际参数	151
5.4.2	函数返回值	154
5.5	函数调用	156
5.5.1	调用的一般形式	156
5.5.2	函数调用方式	156
5.5.3	函数调用声明	158
5.5.4	函数的嵌套调用	159
5.5.5	函数的递归调用	160
5.6	数组与函数	163
5.7	变量的作用域	165
5.7.1	局部变量	165
5.7.2	全局变量	167
5.8	变量的存储类型	169
5.8.1	变量的动态和静态存储方式	170
5.8.2	局部变量的存储类别	170
5.8.3	全局变量的存储类别	174
5.8.4	存储类别小结	176
5.9	函数的存储分类	177
5.9.1	内部函数	177
5.9.2	外部函数	178
5.10	实训指导	179

习题	182
上机实验	187
第 6 章 指针	188
6.1 指针的含义	188
6.2 指针变量	189
6.2.1 指针变量的概念	189
6.2.2 指针变量的定义	189
6.2.3 指针变量赋值与初始化	190
6.2.4 指针的算术运算	191
6.2.5 指针变量的引用	192
6.3 指针与数组	194
6.3.1 一维数组的指针	195
6.3.2 二维数组的指针	198
6.4 指针与函数	203
6.4.1 变量的指针作为函数参数	204
6.4.2 指针型函数	207
6.4.3 一维数组的指针作为函数参数	212
6.4.4 二维数组的指针作为函数参数	216
6.5 指针与字符串	220
6.5.1 单字符串	220
6.5.2 多字符串	224
6.5.3 字符串指针作函数参数	226
6.6 字符数组与字符指针的区别	226
6.7 多级指针	227
6.7.1 定义形式	227
6.7.2 传给 main 函数的参数	230
6.7.3 void 指针类型	231
6.8 指针类型小结	232
6.9 综合实例	232
6.10 实训指导	234
习题	238
上机实验	242
第 7 章 编译预处理	243
7.1 文件包含	243
7.2 宏定义	246
7.2.1 无参数宏定义	246
7.2.2 有参数的宏定义	249
7.2.3 宏替换与函数调用的区别	252
7.3 条件编译	252

7.4 实训指导	255
习题	257
上机实验	258
第8章 结构体与共用体	259
8.1 结构体	259
8.1.1 结构体类型的声明	259
8.1.2 结构体变量	261
8.1.3 结构体变量的引用与初始化	263
8.1.4 结构体数组	265
8.1.5 结构体指针	268
8.1.6 结构体与函数	271
8.1.7 结构体嵌套	274
8.2 用结构体指针操作链表	275
8.2.1 链表概述	275
8.2.2 动态内存管理函数	276
8.2.3 链表的基本操作	277
8.2.4 结点的插入与删除	281
8.3 共用体	284
8.3.1 共用体类型定义	284
8.3.2 共用体变量的引用	286
8.4 枚举类型	288
8.5 类型定义 typedef	291
8.6 实训指导	292
习题	294
上机实验	298
第9章 文件	299
9.1 C文件的概述	299
9.1.1 文件的基本概念	299
9.1.2 文件指针	300
9.2 文件的打开与关闭	301
9.2.1 文件的打开	301
9.2.2 文件的关闭	304
9.3 文件的读写	304
9.3.1 单个字符的读写操作	304
9.3.2 字符串的读写操作	306
9.3.3 数据块的读写操作	308
9.3.4 格式化读写函数	310
9.4 文件的定位与出错检测	312
9.4.1 文件的定位	312

9.4.2 文件操作的出错检测	314
9.5 实训指导	315
习题	317
上机实验	320
附录 A 运算符的优先级及其结合性	321
附录 B 常用 ASCII 代码对照表	322
习题参考答案	323
上机实验参考答案	337
参考文献	350

第 1 章 C 语言概述与算法

计算机本身是无生命的机器，要使计算机能够运行起来，完成各种人们预想的任务，就得编制相应的程序让计算机来执行。而这些程序都是由语言编写而成的。

随着计算机技术的不断发展，计算机程序设计的研究开发也得到迅猛的发展。在各种各样的高级语言中，C 语言以其强大的功能和优良的特点，成为国际上公认的、最重要的、少数几种通用程序设计语言之一。

1.1 C 语言的发展史

C 语言由 Dennis Ritchie 设计，并首次在一台使用 UNIX 操作系统的 DEC PDP-11 计算机上实现。C 语言是由早期的编程语言 BCPL 发展演变而来的。Martin Richards 改进了 BCPL 语言，从而促进了 Ken Thompson 所设计的 B 语言的发展，最终促进了 20 世纪 70 年代 C 语言的问世。

1975 年 UNIX 第 6 版公布后，C 语言的突出优点引起了人们的普遍关注。1977 年出现了不依赖于具体机器的 C 语言编译文本“可移植 C 语言编译程序”，使 C 语言移植到其他机器时所需做的工作大大简化，这也推动了 UNIX 操作系统迅速在各种机器上的实现。例如，VAX、AT&T 等计算机系统都迅速相继开发了 UNIX。随着 UNIX 使用的日益广泛，C 语言也迅速得到推广。C 语言与 UNIX 可以说是孪生兄弟，在发展过程中相辅相成。1978 年以后，C 语言已先后移植到大型、中型、小型及微型机上，并独立于 UNIX。现在 C 语言已风靡全世界，成为世界上应用最广的几种计算机语言之一。

随着微型机的日益普及，出现了许多 C 语言版本。由于没有统一的标准，使得这些 C 语言之间出现了一些不一致的地方。为了改变这种情况，美国国家标准协会为 C 语言制定了一套 ANSI 标准，成为现行的 C 语言标准。

除了系统软件外，C 语言还成功地用于数值计算、文字处理、数据库、计算机网络和多媒体等。C 语言所呈现的强有力的表达能力和效率，使得它成为近 10 年来在计算机程序设计实践中做出重大贡献的一种语言。

目前，在计算机上广泛使用的 C 语言编译系统有 Microsoft C、Turbo C、Borland C 等，虽然它们的基本部分是相同的，但还存在差异，本书的上机环境采用 Microsoft Visual C++ 6.0。

1.2 C 语言的特点

作为目前应用最广泛的高级程序设计语言，C 语言具有明显的优点，但也存在一些缺陷。

1. C 语言的优点

(1) 语言描述简洁、灵活、高效

C 语言只有 32 个标准的关键字和 9 种控制语句，并且以易读易写的小写字母为基础，压缩了一切不必要的成分，使程序书写规整紧凑。

(2) 有丰富的数据类型

C 语言数据类型丰富，不仅具有 4 种最基本的数据类型（char, int, float, double），还有多种组合类型（数组和结构）以及复杂的导出类型，并允许使用简单的组合结构构造复杂的数据类型或直接由用户自定义数据类型。

(3) 运算符丰富

C 语言提供了 45 种标准的运算符和多种获取表达式值的方法，并提供了与地址密切相关的指针及其运算符。C 语言的运算符不仅具有优先级的概念，还有结合性的概念。因此，灵活使用各种运算符和表达式，不仅可以简化程序，还可以实现在其他语言中难以实现的运算。

(4) C 语言具有固定的标识符

C 语言的标识符主要用来表示常量、变量、函数和类型等的名字，它们只是起标识作用的一种符号。C 语言共有 32 个固定的标识符，它们都用小写字母表示，如图 1.1 所示。

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

图 1.1 C 语言的标识符

(5) 提供了功能齐全的函数库

C 语言标准函数库提供了功能极强的各类函数库。例如串、数组、结构乃至图形的处理等，只需要调用一下库函数即可实现，为编程者提供了极大的方便。

(6) 具有结构化的控制语句

C 语言提供了 9 种控制语句实现 3 种结构（顺序、分支和循环结构），如 if-else, while, switch, for 等。并用函数作为程序模块，是理想的结构化程序设计语言。C 程序由具有独立功能的函数构成，函数定义是平行、独立的，但函数调用可以是嵌套调用和递归调用，通过函数调用可以实现复杂的程序功能，多种存储属性的数据可共享同一段内存，从而保证了模块化的程序设计风格。另外，对于复杂的源程序文件，可以分割成多个较

小的源文件，分别编译和调试，最后组装，连接得到可执行的目标程序文件。

(7) 具有良好的通用性和程序的移植性

在C语言中，与硬件有关的操作都是通过调用系统提供的库函数实现的，这使得C语言具有很好的通用性，程序能容易地从一种计算机环境移植到另一种计算机环境中。

(8) 生成目标代码质量高，程序执行效率高

C语言生成目标代码的效率与汇编语言相比，一般低10%~20%。C语言可以像汇编语言一样对位、字节和地址，甚至对硬件进行直接操作。换句话说，C语言既具有汇编语言的强大功能，又没有汇编语言的难度，特别适合做底层开发。

(9) 语法限制不严格

C语言程序设计的自由度大，例如，对数组下标越界不作检查，各种类型的变量可以通用等。

2. C语言的缺点

当然，C语言的缺陷还是十分明显的。C语言缺乏一致公认的标准，语法限制不太严谨，运算符的优先级和结合性比较复杂，不容易记忆。C语言对数据类型缺乏一致性的检测，对数组、结构体等类型的整体运算存在一定的限制。C语言在程序设计方面灵活性有余，而安全性和可靠性不足。

对于上述问题，计算机科学家和工程技术人员正在不断寻求解决的办法，并不断提出改进的方案。

1.3 C程序的组成

1.3.1 简单的C程序分析

下面介绍一个最基本的C程序来说明C程序的特性。

例 1.1 显示：“What is your name?”。

```
main()                                /*主函数*/
{
    printf("What is your name? \n"); /*调用库函数 printf() 显示字符串*/
}
```

运行结果：

What is your name?

说明

本程序由一个main()函数构成。main是函数名，函数名后面的括号内是填写参数的，由于本程序主函数没有参数，所以是空的，但括号不能省略。main()后面有一对花括号，花括号内由语句组成，语句末尾一定要加上分号。本程序只有一个语句。

printf()函数是C语言的库函数，它的功能是在屏幕上输出双引号所包含的内容，“\n”是转义字符，它代表回车换行。

程序的注释部分应在 “/*” 与 “*/” 之间，“/” 和 “*” 之间不能有空格，注释部分允许出现在程序的任何位置。

下面我们再分析一个通用的 C 程序。

例 1.2 用户输入两个整数，计算机输出两个数的和。

```
main()                                /*主函数*/
{
    int x,y,z;                        /*定义变量*/
    scanf("%d,%d",&x,&y);             /*从键盘上输入两个变量*/
    z=add(x,y);                       /*计算两个数的和值*/
    printf("The result is %d",z);     /*输出*/
}

int add(int x,int y)
{
    int z;
    z=x+y;                           /*计算和值*/
    return(z);
}
```

输入: 3,5

运行结果:

The result is 8

说明



这个程序包括两个函数，一个是主函数，另一个 add 函数是求出两个整数的和。在执行时，先由 scanf() 函数从键盘上输入两个数字，将输入的数传入 add() 函数中，在 add() 函数中计算两个数的和值，并用 return 语句将和值作为函数的返回值。

以上实例及说明体现了 C 语言程序的如下特点。

- 程序至少要有个 main 函数，用户也可以自己根据需要设计函数，像上面的 add() 函数。
- 函数由两部分组成。第一部分是函数的说明部分，如函数的名称、函数的返回值类型、函数的参数及类型。第二部分是函数的实现部分，包括变量定义和执行语句。
- C 程序总是从 main 函数开始执行，不论 main 函数在程序的什么地方，也就是说，可以将 main 函数放在任何位置。
- C 程序的书写比较自由，可以在一行上写若干语句，也可以在多行上写一条语句。但每个语句后面都要有一个分号，这个分号是必不可少的。这一点必须注意。
- C 语言中没有专门的输入、输出语句。输入和输出是通过 scanf 和 printf 两个库函数实现的。

- C程序中可以用“/*.....*/”对任何部分进行注释,好的程序都要有必要的注释以提高程序的可读性。

从例1.2可以推出C语言程序的一般构成如下:

```

函数类型 函数名(参数类型 参数名) /*自定义函数*/
{
    函数体;
}
main()
{
    变量定义部分;
    语句执行部分;                /*语句执行部分包括调用自己定义的函数*/
}

```

说明



C语言的源程序由函数构成。每一个函数完成相对独立的功能,其中至少包括一个主函数(main()函数)。

一个自定义函数由两部分组成。

- ① 函数的首部,即函数的第一行,包括函数类型、函数名、参数类型和参数名。

例如:

```

int add(int x,int y);
  ↓      ↓      ↓      ↓      ↓      ↓
函数类型 函数名 函数参数类型 函数参数名 函数参数类型 函数参数名

```

一个函数名后面必须跟一对圆括号,函数参数可以没有。

- ② 函数体,即函数首部下面的大括号{.....}内的部分。如果一个函数内有多个大括号,则最外层的一对{}为函数的范围。

函数体一般包括:

声明部分:在这部分中定义所用到的变量,如add()函数中的int z。

执行部分:由若干个语句组成,如add()函数中的z=x+y。当然,在某些情况下也可以没有声明部分。甚至可以既无声明部分,也无执行部分。例如:

```

add()
{
}

```

它是一个空函数,什么也不做,但这是合法的。

1.3.2 简单的C程序应用

首先,书写程序应从书写清楚,便于阅读、理解和维护的角度出发,在书写程序时应遵循以下规则。

- 一条说明语句占一行。
- 在{}中的部分,通常表示程序的某一层结构。{}一般与该结构语句的第一个字母对齐,并单独占一行。

- 低一层次的语句说明可比高一层的语句说明缩进若干格后书写，以便看起来更加清晰，增加程序的可读性。

在编程时应力求遵循这些规则，以养成良好的编程风格。

例 1.3 由键盘输入两个整数，计算出它们的和与差，并在屏幕上输出。

```
main()
{
    int a,b,c,d;
    scanf("%d%d",&a,&b);
    c=a+b;
    d=a-b;
    printf("\n%d+%d=%d\n%d-%d=%d\n",a,b,c,a,b,d);
}
```

本例只有一个 main 函数。首先定义变量为整型，接着调用 scanf 函数，从键盘上输入两个整数，存入变量 a 和 b 的存储单元中，然后求和与差数，并分别存入变量 c 和 d 的存储单元中，最后执行 printf 语句输出计算结果。

输入

8 3

运行结果:

8+3=11

8-3=5

1.3.3 简单的 C 程序种类

一个高级语言程序是由若干条语句构成的，其中每条语句用来完成一个特定的操作。通过每条语句的执行，程序完成特定的功能。

C 语言提供的语句按功能特性分为 5 类。

1. 赋值语句

这是在程序中使用最频繁的语句，一般由赋值表达式加分号构成，完成将表达式结果赋给某一变量的任务，例如

```
x=1;
```

是一个典型的赋值语句。

而

```
x=1
```

是一个赋值表达式。

可以看到，一个表达式的最后加一个分号就构成了一条语句，分号是语句中不可缺少的一部分，例如

```
z=x+y    /*是表达式而不是语句*/  
z=x+y;   /*是语句*/
```

又如

```
x*y;
```

也是一个语句，作用是完成 $x*y$ 的操作，它是合法的，但并没有将 $x*y$ 的积赋给另一变量，所以它无实际意义。

说明



表达式是构成 C 语言的一个重要特色，其实函数调用也属于表达式，这是因为函数调用（如 $\text{abs}(x)$ ）属于表达式的一种。为了便于理解和使用，我们把函数调用语句和表达式语句分开来说。由于 C 程序中大多数语句是表达式语句，所以有人把 C 语言叫做表达式语言。

2. 控制语句

这类语句都具有控制和改变程序流向的功能，具体如下：

- | | |
|---------------|-------------------|
| ● if 语句 | 条件语句 |
| ● for 语句 | 循环语句 |
| ● while 语句 | 循环语句 |
| ● do-while 语句 | 循环语句 |
| ● continue 语句 | 跳出本次循环语句 |
| ● break 语句 | 终止执行循环或 switch 语句 |
| ● switch 语句 | 多分支选择语句 |
| ● goto 语句 | 转向语句 |
| ● return 语句 | 从函数返回主程序语句 |

3. 函数调用语句

这类语句由函数调用加分号构成。

例如

```
scanf("%d",&a);    /*调用 C 语言的标准库函数*/
```

就是一个函数调用语句。

4. 空语句

空语句只有一个分号“;”，它什么也不做，一般作为程序的转向点或作为循环语句中的空循环。下面是一个空语句：

又如

```
while(getchar()!='\n')
```