

高等职业教育

计算机类专业 规划教材

INFORMATION TECHNOLOGY

# 实用C语言程序 设计教程

李金祥 顾小晶 主 编

李爱军 陈 静 陈小英 副主编



中国电力出版社

<http://jc.cepp.com.cn>

高等职业教育

计算机类专业 规划教材

INFORMATION TECHNOLOGY

# 实用C语言程序 设计教程

主 编 李金祥 顾小晶  
副主编 李爱军 陈 静 陈小英  
参 编 陈 珂 熊志勇  
主 审 陈 雁



中国电力出版社

<http://jc.cepp.com.cn>

## 内 容 提 要

本书为高等职业教育计算机类专业规划教材。全书共7个模块,内容包括程序设计基础、结构化程序设计、数组与字符串、函数及应用、指针及应用、组合数据类型、位运算与文件等。每个模块含2~3个任务,每个任务又包含若干个案例,以案例讲解知识点。全书采用模块—任务—案例的方式组织内容,将理论与实践有机结合,由浅入深,突出重点、分散难点,结合实际,注重学生编程能力和设计风格的培养。

本书可作为高职高专院校计算机及相关专业的教材,也可作为备考计算机等级考试和其他从事计算机编程人员的参考用书。

## 图书在版编目(CIP)数据

实用C语言程序设计教程 / 李金祥, 顾小晶主编. —北京:  
中国电力出版社, 2010.6

高等职业教育计算机类专业规划教材

ISBN 978-7-5123-0411-6

I. ①实… II. ①李… ②顾… III. ①C语言—程序设计—  
高等学校: 技术学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2010)第083924号

中国电力出版社出版、发行

(北京市东城区北京站西街19号 100005 <http://jc.cepp.com.cn>)

汇鑫印务有限公司印刷

各地新华书店经售

\*

2010年6月第一版 2011年12月北京第二次印刷

787毫米×1092毫米 16开本 18印张 438千字

印数3001—6000册 定价28.80元

## 敬告读者

本书封面贴有防伪标签,加热后中心图案消失  
本书如有印装质量问题,我社发行部负责退换

版 权 专 有 翻 印 必 究

# 前言

C 语言程序设计是高等院校计算机及相关专业开设的一门重要的专业基础课,也是目前国内外广泛使用的一种程序设计语言,具有功能丰富、编程灵活、可移植性好,既可以用来编写系统程序,又可以用来编写应用程序等优点,普遍受到人们的欢迎,根据第一版《实用 C 语言简明教程》的使用情况和任课教师在多年教学工作中的经验和体会,以及目前职业教育的教学改革,对该教材进行了重新修订,与第一版相比,第二版从编写体例、组织结构、内容安排等方面都有了很大变动,目标更明确,针对性更强,新增了“学生成绩管理系统”的贯通案例,实验平台采用了 Visual C++环境,以便学生后续能更容易学习面向对象程序设计的其他语言。

具体来讲,本书具有以下特点:

(1) 内容组织上层次分明、结构清晰、实用性强,全面讲授 C 语言程序设计的基本思想、方法和解决实际问题的技巧。

(2) 采用项目、案例式教学,针对性更强,从以学生为中心的指导思想出发,结合高职高专学生的特点,采用源于现实生活的典型案例导入,融理论于案例之中,分散难点,突出重点,使理论与实践密切结合。

(3) 通过贯通案例,掌握模块化程序设计的基本思路,体现软件工程的基本思想和方法。为学生今后进一步开发中、大型软件系统奠定基础。

(4) 每个任务都给出明确的学习目标,通过案例讲解,归纳知识点,通过技能训练,培养学生的动手能力,通过拓展与练习,进一步提升编程技能。

(5) 以 Visual C++环境作为操作平台,给学生后续的学习与发展打好基础。书中全部案例源程序在该环境中调试通过。

全书包括 7 个模块和 5 个附录,模块 1 程序设计基础,模块 2 结构化程序设计,模块 3 数组与字符串,模块 4 函数及应用,模块 5 指针及应用,模块 6 组合数据类型,模块 7 位运算与文件。

本书由苏州市职业大学李金祥、顾小晶主编,李爱军、陈静、陈小英副主编,陈珂和熊志勇参编。其中顾小晶编写模块 2 和附录,李爱军编写模块 3 和模块 4,陈静编写模块 1,陈小英编写模块 6 和贯通案例,陈珂编写模块 5,熊志勇编写模块 7。全书由李金祥、顾小晶统稿,由陈雁主审。本书在编写过程中,得到苏州市职业大学教务处和中国电力出

版社的指导和帮助，在此表示衷心感谢！为了教师便于授课，与教材配套的教学课件可在 <http://jc.cepp.com.cn> 网站下载。

由于编者水平有限，书中难免有不足之处，欢迎各位读者提出宝贵意见。联系邮箱：[ljx@jssvc.edu.cn](mailto:ljx@jssvc.edu.cn)。

编 者

2010 年 5 月

# 目 录

## 前言

|                       |     |
|-----------------------|-----|
| 模块 1 程序设计基础           | 1   |
| 任务 1 程序结构与特征          | 1   |
| 任务 2 基本数据类型           | 17  |
| 自测题                   | 30  |
| 模块 2 结构化程序设计          | 35  |
| 任务 1 顺序结构程序设计和程序的基本结构 | 35  |
| 任务 2 选择结构程序设计         | 43  |
| 任务 3 循环结构程序设计         | 58  |
| 自测题                   | 78  |
| 模块 3 数组与字符串           | 81  |
| 任务 1 一维数组             | 81  |
| 任务 2 二维数组             | 93  |
| 任务 3 字符与字符串           | 100 |
| 自测题                   | 110 |
| 模块 4 函数及应用            | 117 |
| 任务 1 函数定义             | 117 |
| 任务 2 函数和数组、变量的作用域和生存期 | 132 |
| 任务 3 函数的嵌套调用与递归调用     | 146 |
| 自测题                   | 155 |
| 模块 5 指针及应用            | 159 |
| 任务 1 地址与指针            | 159 |
| 任务 2 指针与数组            | 167 |
| 任务 3 指针与函数            | 185 |
| 自测题                   | 202 |
| 模块 6 组合数据类型           | 204 |
| 任务 1 结构体              | 204 |
| 任务 2 共用体              | 216 |

|                             |     |
|-----------------------------|-----|
| 任务 3 枚举.....                | 222 |
| 自测题 .....                   | 233 |
| 模块 7 位运算与文件.....            | 235 |
| 任务 1 位运算.....               | 235 |
| 任务 2 文件.....                | 240 |
| 自测题 .....                   | 257 |
| 附录 A ASCII 代码表.....         | 259 |
| 附录 B C 语言的关键字 .....         | 263 |
| 附录 C 运算符的优先级和结合性 .....      | 263 |
| 附录 D 常用 C 库函数 .....         | 264 |
| 附录 E 编译、连接时常见的错误和警告信息 ..... | 275 |
| 参考文献 .....                  | 279 |

## 模块 1

## 程序设计基础

## 任务1 程序结构与特征

## 学习目标

了解 C 语言的发展背景,掌握 C 语言程序的基本结构,领会 C 语言程序设计的风格;掌握在 Visual C++环境中调试程序的方法。

## 1.1.1 案例讲解



## 案例1 菜单显示

## 1. 问题描述

模拟学校管理信息系统软件的菜单,利用 printf 函数按一定格式输出文字,可以在屏幕上显示如图 1-1 所示的简单程序界面。



图 1-1 程序界面

(1)启动 Visual C++。选择“开始”→“程序”→Microsoft Visual Studio 6.0→Microsoft Visual C++ 6.0 菜单,启动 Microsoft Visual C++ 6.0 编译系统。

(2)新建工程。选择“文件”→“新建”菜单,在出现的“新建”窗口的“工程”选项中选择 Win32 Console Application,在右侧“位置”栏中选择 D: 盘 Chapter1 文件夹,“工程名称”栏中输入“EX1\_1”,这时界面变为如图 1-2 所示。单击“确定”按钮后,在出现的应用框架选择向导窗口和新建工程信息窗口中分别单击“完成”和“确定”按钮,完成新工程的建立,如图 1-3 所示。

(3)编写源程序。重新选择“文件”→“新建”菜单命令,在出现的“新建”窗口的“文件”选项卡中选择“C++ Source File”项,在右侧“文件”栏中输入“EX1\_1.C”,如图 1-4 所示。C 语言程序的文件扩展名是.C。



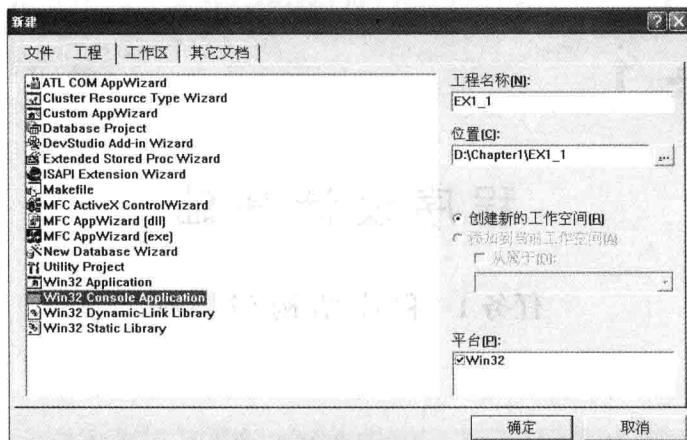


图 1-2 “新建”窗口的“工程”选项卡

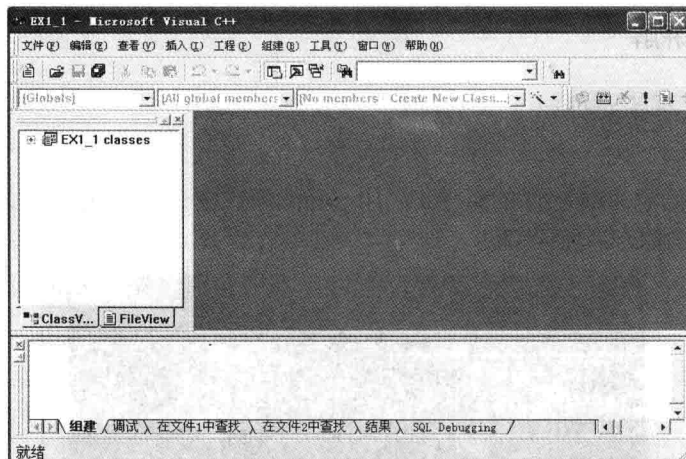


图 1-3 建立新工程后的窗口

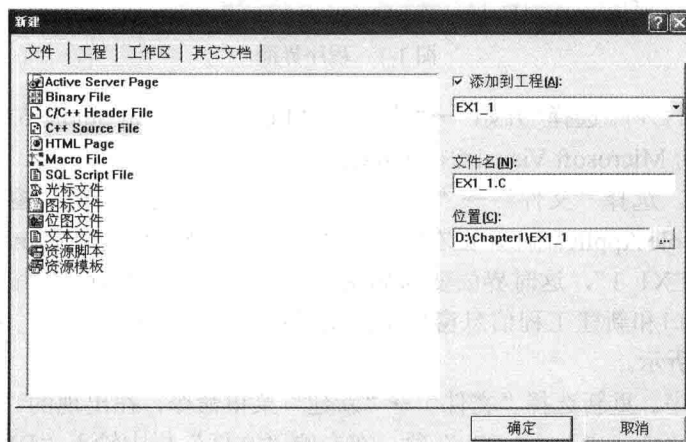


图 1-4 “新建”窗口的“文件”选项卡

单击“确定”按钮后，在出现的窗口（称为编辑窗口）右侧输入如下代码，如图 1-5 所示。



图 1-5 编写源程序

```
/* EX1_1.C */
#include <stdio.h>
main( )
{
    printf("*****\n");
    printf("***** 校园管理信息系统 *****\n");
    printf("*****\n");
    printf("**      1.学生管理      *\n");
    printf("**      2.教师管理      *\n");
    printf("**      3.课程管理      *\n");
    printf("**      4.成绩管理      *\n");
    printf("**      5.退出系统      *\n");
    printf("*****\n");
    printf("请按 1-5 按钮选择菜单项: ");
}
```

(4) 保存程序。选择“文件”→“保存”菜单命令或单击“保存”按钮进行保存。为了防止意外丢失程序代码，应养成及时存盘的好习惯。

(5) 编译。选择“组建”→“编译 EX1\_1.C”菜单命令或单击“编译”按钮进行编译。系统在编译前会自动将程序保存，然后进行编译。如果系统在编译的过程中发现错误，将在下方窗口中列出所有错误和警告。双击显示错误或警告的第一行，则光标自动跳到代码的错误行。修改程序中的该错误后，重新进行编译，若还有错误或警告，继续修改和编译，直到没有错误为止。本实例没有发现任何错误和警告，所以错误和警告数都为 0，如图 1-6 所示。

(6) 连接。选择“组建”→“组建 EX1\_1.exe”菜单命令或单击“连接”按钮，与编译时一样，如果系统在连接的过程中发现错误，将在下面窗口中列出所有错误和警告。修改错误后重新编译和连接，直到编译和连接都没有错误为止。

(7) 运行。选择“组建”→“执行 EX1\_1.exe”菜单命令或单击“运行”按钮，在出现的黑屏中显示运行结果，如图 1-7 所示。单击任意键回到编辑窗口。

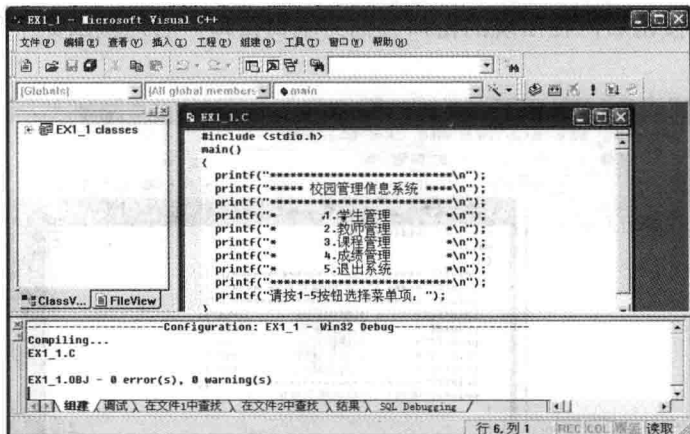


图 1-6 编译后的情况



图 1-7 案例 1 运行结果

## 2. 归纳分析

(1) 任何一个 C 程序是由若干个函数构成的，C 语言中用函数来实现特定的功能。一个 C 源程序至少包含一个 main 主函数，也可以包含一个 main 主函数和若干个其他函数。

主函数的一般框架为：

```
main()
{ 定义变量部分
  功能语句部分
}
```

(2) 在屏幕上显示内容使用 printf 函数，printf 函数显示字符串的格式为：

```
printf("一串字符");
```

在函数后面加分号构成输出语句，语句是程序执行的基本单位。程序中“\n”是换行符。

(3) C 语言系统提供丰富的标准库函数，而且为了方便使用，对这些函数进行分类并存放在对应的头文件中。程序 EX1\_1.C 的第 1 行 #include <stdio.h> 是一条预处理命令，作用是将头文件 stdio.h 包含入本程序，如果程序中需要输入输出数据，就必须包含头文件 stdio.h，而数学函数则存放在“math.h”文件中。在附录中列出了系统常用的库函数。

(4) C 语言程序必须经过编辑、编译、连接的过程后才能运行。C 语言的上机环境很多，如 Turbo C 2.0、Visual C++6.0、Borland C++等。本书选用 Visual C++6.0，主要是方便读者使

用鼠标编辑，也便于过渡到 C++ 的学习。



## 案例2 销售额的计算

### 1. 问题描述

某玩具店为了促销某商品，周日举办了一场多买多优惠的活动，即买 1 件 58.5 元、2 件 108.5 元、3 件 150 元。编写程序，输入周日买 1 件的人数、买 2 件的人数、买 3 件的人数，并计算当天的总销售额和平均单价。

### 2. 编程分析

main()

{

定义整型变量 n1、n2、n3 和 n

定义双精度实型变量 sum、ave

分别输入买 1 件、2 件、3 件的人数存入 n1、n2、n3 中

计算总的卖出件数存入 n 中

计算总销售额放在 sum 中

计算平均单价放在 ave 中

显示总销售额和平均单价

}

### 3. 编写源程序

```
/* EX1_2.C */
```

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

```
main()
```

```
{
```

```
int n1,n2,n3,n;
```

```
double sum,ave;
```

```
printf("请输入三种人数 n1,n2,n3:");
```

```
scanf("%d,%d,%d",&n1,&n2,&n3);
```

```
n=1*n1+2*n2+3*n3;
```

```
sum=58.5*n1+108.5*n2+150*n3;
```

```
ave=sum/n;
```

```
printf("总销售额:%lf, 平均单价:%lf\n",sum,ave);
```

```
}
```

特别提示：源程序编写好后要存盘。例如以“EX1\_2.C”为文件名存盘。

4. 编译、连接后运行程序，等待输入时输入“3, 4, 5<回车>”，则运行结果如图 1-8 所示

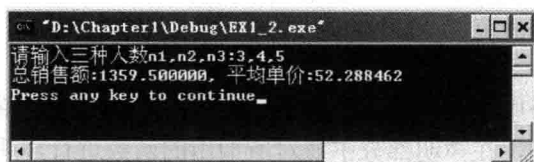


图 1-8 案例2 运行结果

## 5. 归纳分析

(1) 程序中的 `printf` 函数输出一条信息“请输入三种人数 `n1, n2, n3`：”，用于提示用户输入三种数据，并说明输入的格式（用逗号分隔三个数），和 `scanf` 函数配合使用以实现用户和计算机之间的信息交互。

(2) 从键盘输入数据使用 `scanf` 函数，该函数的一般形式为

`scanf(<格式控制字符串>, <地址列表>)`

<格式控制字符串>是用双引号括起来的字符串，也称“转换控制字符串”，它包括两种信息：一部分是普通字符，这些字符将按原样输出；另一部分是格式说明，以“%”开始，后跟一个或几个规定字符，用来确定输出内容格式，例如 `%d`、`%f` 等，它的作用是将输出的数据转换为指定的格式输出。

<地址列表>是由若干个地址组成的列表，可以是变量的地址或字符串的首地址。

程序中字符串“`&n1, &n2, &n3`”中的“&”是“地址运算符”，`&a` 指 `a` 在内存中的地址。上面 `scanf` 函数的作用是：按照 `n1`、`n2`、`n3` 在内存的地址将 `n1`、`n2`、`n3` 的值存进去。变量 `n1`、`n2`、`n3` 的地址是在编译连接阶段分配的。

“`%d, %d, %d`”表示按基本整型输入数据。

### 1.1.2 基础理论

#### 1. C 程序的基本结构

通过以上几个案例，可以看到：

(1) C 程序由函数构成。C 程序是由函数构成的，C 语言中用函数来实现特定的功能。一个 C 源程序至少包含一个 `main` 函数，也可以包含一个 `main` 函数和若干个其他函数。因此，函数是 C 程序的基本单位，C 语言的函数相当于其他语言中的子程序，可以说 C 语言是函数式的语言，程序中的全部工作都是由各个函数分别完成的，编写 C 程序就是编写一个个函数。

C 语言的这种特点使得容易实现程序的模块化。

(2) 函数由两部分组成。C 语言中的函数由两部分组成：

1) 函数的首部，即函数的第一行。包括函数属性、函数类型、函数名、函数参数（形参）名、参数类型。

程序 EX1\_1.C 中的 `main` 函数的首部为

`main()`

在此例中，只定义了函数名，没有给出函数的类型、参数等内容，这是允许的，但一个函数名后面必须跟一对圆括弧。

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

函数体一般包括：

声明部分：在这部分中定义所用到的变量，如程序 EX1\_2.C 中的“`int n1, n2, n3, n;`”。在后面课程中还将看到，在声明部分中要对所调用的函数进行声明。

执行部分：由若干个语句组成。

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。

(3) C 程序从 `main` 函数开始执行。一个 C 程序总是从 `main` 函数开始执行的,而不论 `main` 函数在整个程序中的位置如何(`main` 函数可以放在程序最前头,也可以放在程序最后,或在一些函数之前,在另一些函数之后)。

(4) 程序书写格式自由。C 程序书写格式自由,一行内可以写几个语句,一个语句可以分写在多行上。C 程序没有行号,也不像 Fortran 或 COBOL 那样严格规定书写格式(语句必须从某一列开始书写)。

(5) 分号是语句的结束符。C 语言中,每个语句和数据定义的最后必须有一个分号,分号是 C 语句的必要组成部分。如

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

(6) 程序中可以使用注释。可以用 `/*.....*/` 对 C 程序中的任何部分作注释。一个好的、有使用价值的源程序都应当加上必要的注释,以增加程序的可读性。

## 2. printf 函数

(1) printf 函数的一般格式。

`printf(<格式控制字符串>,<参数列表>)`

如 `printf("i=%d,c=%c\n",i,c);`

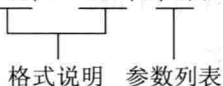
括弧内包括两部分:

<格式控制字符串>是用双引号括起来的字符串,也称“转换控制字符串”,它包括两种信息:一部分是普通字符,这些字符将按原样输出,例如“`i=c`”;另一部分是格式说明,以“`%`”开始,例如`%d`、`%f`等,它的作用是将输出的数据转换为指定的格式输出。

<参数列表>是需要输出的一些数据,可以是表达式,例如上面 `printf` 函数中的“`i,c`”部分,其个数必须与格式化字符串所说明的输出参数个数一样多,各参数之间用逗号分开,且顺序一一对应,否则将会出现意想不到的错误。

下面是另一个例子:

```
printf("a=%d,b=%d",a,b);
```



在上面双引号中的字符除了“`%d`”和“`%d`”以外,还有非格式说明的普通字符,它们按原样输出。如果 `a`、`b` 的值分别为 10、20,则输出为:

a=10,b=20

其中下划线的字符是 `printf` 函数中的“格式控制字符串”中的普通字符按原样输出的结果。10 和 20 是 `a` 和 `b` 的值(注意 10 和 20 无前导空格和尾随空格)。

(2) 格式说明。格式说明的一般形式为:

`%[标志][输出最小宽度][.精度][长度] 类型格式字符`

其中方括号 `[]` 中的项为可选项。各项的意义如下:

1) 类型格式字符。类型格式字符用以表示输出数据的类型,注意不同的类型数据要用相应的类型字符输出,其格式符和意义如表 1-1 所示。

表 1-1 类型格式字符

| 符号 | 作用       | 符号  | 作用                           |
|----|----------|-----|------------------------------|
| d  | 十进制有符号整数 | x、X | 无符号以十六进制表示的整数                |
| u  | 十进制无符号整数 | o   | 无符号以八进制表示的整数                 |
| f  | 浮点数      | e   | 指数形式的浮点数                     |
| c  | 单个字符     | g   | 浮点数, 选用 f 或 e 格式中输出宽度较短的一种格式 |
| s  | 字符串      | p   | 指针的值                         |

2) 标志。标志字符为一、+、#、空格四种, 其意义如表 1-2 所示。

表 1-2 标志字符

| 标志 | 意义           | 标志 | 意义                                                                                   |
|----|--------------|----|--------------------------------------------------------------------------------------|
| -  | 结果左对齐, 右边填充格 | 空格 | 输出值为正时冠以空格, 为负时冠以负号                                                                  |
| +  | 输出符号 (正号或负号) | #  | 对 c、s、d、u 类无影响;<br>对 o 类, 在输出时加前缀 0;<br>对 x 类, 在输出时加前缀 0x;<br>对 e、g、f 类当结果有小数时才给出小数点 |

3) 输出最小宽度。用十进制整数来表示输出的最少位数。若实际位数多于定义的宽度, 则按实际位数输出, 若实际位数少于定义的宽度则补以空格或 0。

4) 精度。精度格式符以 “.” 开头, 后跟十进制整数。本项的意义是: 如果输出数值, 则表示小数的位数; 如果输出的是字符, 则表示输出字符的个数; 若实际位数大于所定义的精度位数, 则截去超过的部分。

5) 长度。长度格式符为 h、l 两种, h 表示按短整型量输出, l 表示按长整型量输出。

3. scanf 函数

(1) 格式说明。和 print 函数中的格式说明相似, 以 % 开始, 以一个类型格式字符结束, 中间可以插入附加的字符。表 1-3 列出 scanf 用到的类型格式字符。

表 1-3 scanf 函数中类型格式字符

| 格式字符      | 说明                                            |
|-----------|-----------------------------------------------|
| d、i       | 用来输入有符号的十进制整数                                 |
| u         | 用来输入无符号的十进制整数                                 |
| o         | 用来输入无符号的八进制整数                                 |
| x、X       | 用来输入无符号的十六进制整数 (大小写作用相同)                      |
| c         | 用来输入单个字符                                      |
| s         | 用来输入字符串, 将字符串送到一个字符数组中, 在输入时以非空白字符开始第一个空白字符结束 |
| f、e、E、g、G | 用来输入实数, 可以用小数形式或指数形式输入                        |
| e、E、g、G   | 与 f 作用相同, e 与 f、g 可以互相替换 (大小写作用相同)            |

表 1-4 列出 scanf 可以用的附加说明字符 (修饰符)。

表 1-4

scanf 的附加格式说明符

| 字 符   | 说 明                                              |
|-------|--------------------------------------------------|
| 字母 l  | 用于输入长整型数据 (%ld、%lo、%lx) 和 double 型数据 (%lf 或 %le) |
| 字母 h  | 用于输入短整型数据 (%hd、%ho、%hx)                          |
| 正整数 m | 域宽, 指定输入数据所占宽度 (列数)                              |
| 字符*   | 表示本输入项在读入后不赋给相应的变量                               |

## (2) 注意事项:

1) 可以用整数 m 指定输入数据所占列数, 系统自动按它截取所需数据, 如

```
scanf ("%3d%3d", &a, &b);
```

输入: 123456✓

系统自动将 123 赋给 a, 456 赋给 b。

此方法也可用于字符型:

```
scanf ("%3c", &ch);
```

如果从键盘连续输入三个字符 abc, 由于 ch 只能容纳一个字符, 系统就把第一个字符 a 赋给 ch。

2) 如果在<格式控制>字符串中除了格式说明以外还有其他字符, 则在输入数据时应输入与这些字符相同的字符。如

```
scanf ("%d,%d", &a, &b);
```

输入时应用如下形式:

1,2✓

注意 1 后面是逗号, 它与 scanf 函数中的<格式控制>中的逗号对应。如果输入时不用逗号而用空格或其他字符是不对的, 如

1:2✓ 系统只能读入 1 存入变量 a, 然后等待输入逗号

如果是

```
scanf ("%d:%d:%d", &hour, &mintue, &second);
```

输入应该用以下形式:

11:43:15✓

如果是

```
scanf ("a=%d,b=%d,c=%d", &a, &b, &c);
```

输入应为以下形式:

a=10,b=100,c=-1

3) 在用 "%c" 格式输入字符时, 空格字符和“转义字符”都作为有效字符输入:

```
scanf ("%c%c%c", &ch1, &ch2, &ch3);
```

如输入

x □ y □ z✓

字符 x 存入 ch1, 字符空格存入 ch2, 字符 y 存入 ch3, 因为%c 只要求读入一个字符,



后面不需要用空格作为两个字符的间隔, 因此  $x$  和  $y$  之间的空格作为下一个字符送给 `ch2`。

4) 在输入数据时, 遇以下情况时该数据认为结束:

遇空格, 或按“回车”或“跳格”(Tab)键。

按指定的宽度结束, 如“%3d”, 只取三列。

遇非法输入。

输入输出是程序中最基本的操作, 而 C 语言的格式化输入输出函数的规定又比较烦琐, 用得不对就得不到预期的结果, 所以本章作了较为详细的介绍。但读者在学习时不必花许多精力在每一个细节上, 只要重点掌握最常用的一些规则即可。其他部分可以通过编写和调试程序来逐步掌握。

#### 4. getchar 函数

此函数的作用是从终端(或系统隐含指定的输入设备)输入一个字符。`getchar` 函数没有参数, 其一般形式为

```
getchar( )
```

函数的值就是从输入设备得到的字符, 如

```
a=getchar( );
```

该语句作用是将从键盘输入的字符存放在变量  $a$  中。

#### 5. putchar 函数

`putchar` 函数的作用是向终端输出一个字符, 如

```
putchar(c);
```

其中  $c$  为一个字符变量或常量。

#### 6. getch 函数

此函数和 `getchar` 函数类似, 格式为 `getch( )`; 也是从终端输入一个字符, 但两者不同的是 `getch` 函数不会回显输入的字符, 而且也不需要用回车键确认输入。

### 1.1.3 技能训练

**【实验 1.1.1】** 运行下面的程序, 分析运行结果。

```
/* EX1_3.C */
#include<stdio.h>
main( )
{
    int i;
    long j;
    float f;

    i = 123;
    j = 123456;
    printf("%d,%5d,%05d \n",i,i,i);
    printf("%ld,%8ld,%08ld \n",j,j,j);
    f=123.4;
    printf("%f\n", f);
    printf("%l0f\n",f);
    printf("%l0.2f\n",f);
    printf("%.2f\n",f);
}
```