

第3章

顺序结构程序设计

通过第2章的学习,已经基本了解和掌握了C语言程序设计的基本知识,本章对编程中遇到的其他问题进行描述。

3.1 C语言的基本语句

C语言中的语句可分为表达式语句、函数调用语句、控制语句、复合语句和空语句5种。

1. 表达式语句

表达式语句由表达式加上分号“;”组合而成。任意一个表达式加上分号后,都构成一条合法的表达式语句。例如, $x * y + z$ 是表达式,而“ $x * y + z$;”是表达式语句; $i++$ 是表达式,而“ $i++$;”是表达式语句。

2. 函数调用语句

合法的函数调用加上分号“;”构成函数调用语句,其一般形式为

函数名(实际参数列表);

例如: `printf("Hello!");` // `printf` 是系统库函数,实现输出功能,把字符串 `Hello!` 输出

3. 控制语句

控制语句用于控制程序的执行流程,以实现程序的结构需要。C语言中主要包括9种控制语句,如选择语句与循环语句等,将在后面的章节中详细介绍。

4. 复合语句

将单条语句或多条语句用花括号“{}”括起来构成的语句块称为复合语句。

例如: `{ t = a; a = b; b = t; }`

注意: 复合语句在语法上相当于单条语句,花括号“{}”中的语句被看成一个整体,要么全执行,要么全不执行。

5. 空语句

在C语言中分号“;”是语句的结束符,单独一个“;”也构成一条合法的语句,称为空语

句。空语句是不执行任何操作的语句。

3.1.1 赋值语句

赋值语句是由赋值表达式加上分号构成的表达式语句。其一般形式为

赋值表达式;

赋值语句的功能和特点与赋值表达式相同,它是程序中使用最多的语句之一。当执行赋值语句时,会完成计算和赋值的操作。

在赋值语句的使用过程中需要注意以下几点。

(1) 在赋值符“=”右边的表达式可以又是一个赋值表达式,即有如下形式。

变量 = 变量 = ... = 表达式;

例如,“x=y=z=3;”是一个合法的赋值语句。按照赋值运算符的右结合性,该语句实际上等价于“z=3; y=z; x=y;”。

(2) 在变量定义中,不允许连续给多个变量赋初值。“int x=y=z=3;”是错误的,应该写为“int x=3,y=3,z=3;”,而赋值语句允许连续赋值,如“x=y=z=3;”是正确的。

3.1.2 顺序结构程序特点

C 语言程序设计中包含顺序结构、选择结构、循环结构三种结构。顺序结构程序的特点如下。

(1) 程序在执行过程中严格按照语句书写的先后顺序执行。

(2) 程序一般由定义变量、输入数据或赋值、中间处理、输出结果 4 部分组成。

【例 3.1】 顺序结构程序设计举例。

```
#include "stdio.h"
main()
{   int n1,n2,sum;           //定义变量
    scanf("%d%d",&n1,&n2);   //输入数据
    sum = n1 + n2;           //运算
    printf("sum = %d\n",sum); //输出结果
}
```

3.2 数据的输入和输出

C 语言本身不提供输入和输出语句,其功能是由系统函数来完成的。常用到的标准输入输出函数有 printf()、scanf()、putchar()、getchar()等。

在编写 C 语言源程序的过程中,如果使用了系统库函数,要用文件包含预编译命令将有关的“头文件”包含到用户的源文件中,且放在源文件的最前面。文件包含预处理命令格式为 #include“文件名.h”。例如,scanf、printf 等输入输出函数头文件是 stdio.h,所以文件包含预处理命令为 #include "stdio.h",是先从当前目录查找头文件,然后到系统约定的路径查看。

3.2.1 printf 输出函数

```
printf("格式字符串",输出列表);
```

功能：将列表中的各项值按照指定格式依次输出显示到屏幕上。

```
例如：printf("a = %d,b = %d\n",a,b);
```

1. 输出列表

输出列表是要输出的数据,可以没有。当有两个或两个以上输出项时,要用逗号(,)分隔。输出列表中的输出项可以是常量、变量或表达式。下面的 printf()函数都是合法的。

```
printf("I am a student.\n");
printf(" %d",3+2);
printf("a = %d,b = %d\n", a, a+3);
```

2. 格式字符串

格式字符串也称转换控制字符串,由普通字符和格式说明符两部分组成。

(1) 普通字符,即需要原样输出的字符(包括转义字符)。格式字符串中的普通字符原样输出。

```
例如：printf("a = %d,b = %d\n",a,b); //语句中的",""a = ""b = "等都是普通字符。
```

(2) 格式说明符以“%”开始,以一个格式字符结束,中间可以插入附加说明符,它的作用是将输出的数据转换为指定的格式输出。其一般形式为：

```
%[附加说明符]格式字符
```

3. 格式字符

使用 printf()实现整型数据的输出时,应该根据数据的类型和输入输出的形式,使用合适的格式字符(如表 3.1 和表 3.2 所示)。

表 3.1 整型数据格式字符

数据类型	输入输出形式			
	十进制	八进制	十六进制	格式符的含义
int	%d	%o	%x	以十进制、八进制、十六进制输入/输出一个整数
long	%ld	%lo	%lx	以十进制、八进制、十六进制输入/输出一个长整数
unsigned	%u	%o	%x	以十进制、八进制、十六进制输入/输出一个无符号整数

注：在 VC++ 环境中,由于 int 也是 4B,所以它和长整型之间没有区别,都可以使用 %d； %ld 可以忽略不用。

表 3.2 常用的附加说明符

附加说明符	说 明
m(列宽)	按宽度 m 输出。若 m>数据长度,左补空格,否则按实际位数输出
-m(列宽)	按宽度 m 输出。若 m>数据长度,右补空格,否则按实际位数输出

【例 3.2】 格式字符 %d 的使用。

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

```
main()
{
    int a = 123;
    printf("a = %d\n",a);
}
```

程序运行结果为

```
a = 123
```

对于整型数据,还可以输出八进制、十六进制、无符号十进制类型的数据。

【例 3.3】 格式字符 %d、%ld 和列宽的使用。

```
#include "stdio.h"
main()
{
    int a = 123;
    long b = 123456;
    printf("a = %d,a = %5d,a = %-5d,a = %2d\n",a,a,a,a);
    printf("b = %ld,b = %8ld,b = %5ld\n",b,b,b);
}
```

程序运行结果为

```
a = 123,a = 123,a = 123,a = 123
b = 123456,b = 123456,b = 123456
```

【例 3.4】 格式字符 %d、%o、%x、%u 的使用。

```
#include "stdio.h"
main()
{
    int a = 20;
    int b = -1;
    printf(" %d, %o, %x, %u\n",a,a,a,a);
    printf(" %d, %o, %x, %u\n",b,b,b,b);
}
```

程序运行结果为

```
20,24,14,20
1,3777777777,fffffff,4294967295
```

对于整数,当使用八进制、十六进制和无符号形式输出数据时,一律按照无符号形式。无符号形式是指,不论是正数还是负数,系统一律当作无符号整数来输出。不论采用哪种输出形式,数据在内存的二进制序列是确定的。

使用 printf()实现实型数据的输出时,应该根据数据的类型和输入输出的形式,使用合适的格式字符(如表 3.3 和表 3.4 所示)。

表 3.3 实型数据格式字符

函数	数据类型	格式	含 义
printf	float,double	%f	以小数形式输出实数(保留 6 位小数)
	float,double	%e	以指数形式输出实数(小数点前有且仅有一位非 0 的数字)
scanf	float	%f	以小数或指数形式输入一个单精度实数
		%e	
	double	%lf	以小数或指数形式输入一个双精度实数
		%le	

表 3.4 常用的附加说明符

附加说明符	说 明
m(列宽)	按宽度 m 输出。若 m>数据长度,左补空格,否则按实际位数输出
-m(列宽)	按宽度 m 输出。若 m>数据长度,右补空格,否则按实际位数输出
.n(小数位数)	在 f 前,指定 n 位小数
	在 e 或 E 前,指定 n-1 位小数(VC 环境下,是 n 位小数)

【例 3.5】 格式字符 %m.nf 的使用。

```
#include "stdio.h"
main()
{   float f = 123.456;
    double d1,d2;
    d1 = 111111111111.111111111;
    d2 = 222222222222.222222222;
    printf(" %f, %12f, %12.2f, % -12.2f, % .2f\n",f,f,f,f,f);
    printf("d1 + d2 = %f\n",d1 + d2);
    printf(" %e, %10.3e, % -10.3e, % .2e\n",f,f,f,f,f);
}
```

程序运行结果为

```
123.456001, □□123.456001, □□□□□123.46, 123.46□□□□□, 123.46
d1 + d2 = 333333333333.333000
1.234560e+002, 1.235e+002, 1.235e+002, 1.23e+002
```

3.2.2 scanf 输入函数

scanf()函数的作用：按指定的格式从键盘读入数据，并将数据存入地址列表指定的内存单元中。一般调用格式：

```
scanf("格式字符串",地址列表);
例如：scanf("a = %d,b = %d\n",&a,&b);
```

1. 地址列表

地址列表是由若干地址组成的列表，可以是变量的地址或其他地址。C 语言中变量的地址通过取地址运算符“&”得到，表示形式为：&变量名。例如，变量 a 的地址为 &a。

2. 格式字符串

格式字符串同 printf()函数类似，由普通字符和格式说明符组成。普通字符，即需原样输入的字符。格式说明符同 printf()函数相似。

【例 3.6】 输入函数 scanf 的应用举例。

```
#include "stdio.h"
main()
{   int a,b,c;
    scanf(" %d, %d",&a,&b);
    c = a + b;
    printf("c = %d\n",c);
}
```

则在执行程序时,在执行的界面上输入 3,4 ↵,这样 a 和 b 的值分别为 3 和 4。

说明:

(1) 格式字符串中的普通字符必须原样输入。例如:

```
scanf("a = %d, b = %d", &a, &b); //输入时应用如下形式:a = 3, b = 4 ↵
```

但如果没有任何间隔,输入数据时需要以空格、Enter 键、Tab 键间隔,例如:

```
scanf(" %d %d", &a, &b); //输入时应用如下形式:3 4 ↵
```

这样,a 和 b 就会得到 3 和 4。

(2) 地址列表中的每一项必须为地址。例如:

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

不能写成:

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

虽然在编译时不会出错,但是得不到正确的输入数据。

3.2.3 字符和字符串的输入和输出

上面已经介绍了格式化输入函数 scanf()和格式化输出函数 printf()的基本使用,下面介绍字符数据的输出和输入函数: putchar()和 getchar()。

1. 字符型数据的输入和输出(%c)

字符型数据的输入输出既可以使用%c 格式符,也可以使用 putchar()和 getchar()函数。

【例 3.7】 格式字符%c 的使用。

```
#include "stdio.h"
main()
{   char c = 'A';
    int i = 65;
    printf("c = %c, %5c, %d\n", c, c, c);
    printf("i = %d, %c", i, i);
}
```

程序运行结果为

```
c = A, □□□□A, 65
i = 65, A
```

需要强调的是,在 C 语言中,整数可以用字符形式输出,字符数据也可以用整数形式输出。将整数用字符形式输出时,系统首先求该数与 256 的余数,然后将余数作为 ASCII 码,转换成相应的字符输出。

2. 字符串数据输出(%s)

【例 3.8】 格式字符%s 的使用。

```
#include "stdio.h"
main()
```

```
{ printf(" %s, %5s, % -10s", "Internet", "Internet", "Internet");  
  printf(" %10.5s, % -10.5s, % 4.5s\n", "Internet", "Internet", "Internet");  
}
```

程序运行结果为

Internet, Internet, Internet□□, □□□□□Inter, Inter□□□□□, Inter

可以使用%s 格式符输出字符串常量。

注意：系统输出字符和字符串时,不输出单引号和双引号。

3. 字符输入函数 getchar

字符型数据的输入输出还可以使用 putchar()和 getchar()函数完成。使用它们时一定要使用文件包含: #include "stdio. h"或#include <stdio. h>。

getchar()函数的功能是从键盘读入一个字符,是无参函数。一般调用格式:

getchar()

说明:

(1) getchar()函数一次只能接收一个字符,即使从键盘输入多个字符,也只接收第一个。空格和转义字符都作为有效字符接收。从键盘上输入的字符不能带单引号,输入以Enter 键结束。

(2) 接收的字符可以赋给字符型变量或整型变量,也可以不赋给任何变量,作为表达式的一部分。

4. 字符输出函数 putchar

putchar()函数的功能是向显示器输出一个字符。一般调用格式:

putchar(参数)

其中,参数可以是整数类型表达式,一般为算术表达式。例如:

```
putchar('a')           //输出字符 a  
putchar(65)            //输出 ASCII 码为 65 的字符 A  
putchar('a' + 2)       //输出字符 c  
putchar('\n')          //输出一个换行符
```

说明:

(1) putchar()函数一次只能输出一个字符,即该函数有且只有一个参数。

(2) putchar()函数可以输出转义字符。

【例 3.9】 从键盘上输入一个大写字符,输出对应的小写字符。

```
#include "stdio. h"  
main()  
{ char c1, c2;  
  c1 = getchar();           //从键盘输入字符直到按 Enter 键结束  
  c2 = c1 + 32;  
  putchar(c2);             //输出运算结果  
}
```

【例 3.10】 程序举例,输入三角形边长,求三角形面积。

```
#include "math.h"
#include "stdio.h"
main()
{   float a,b,c,s,area;
    scanf("%f,%f,%f",&a,&b,&c);
    s = 1.0/2 * (a + b + c);
    area = sqrt(s * (s - a) * (s - b) * (s - c));
    printf("a = %7.2f, b = %7.2f, c = %7.2f, s = %7.2f\n",a,b,c,s);
    printf("area = %7.2f\n",area);
}
```

输入: 3,4,6 ✓

输出: a= 3.00,b= 4.00,c= 6.00 s= 6.50
area= 5.33

在使用 scanf() 和 printf() 实现数据的输入和输出时,还要注意以下几点。

(1) 格式字符一定要小写(e、x 除外),否则将不是格式字符,而是作为普通字符处理。

例如: printf("%D",123); //输出结果为: %D

(2) 格式说明与输出项从左向右一一对应,两者的个数可以不相同,若输出项个数多于格式说明个数,输出项右边多出的部分不被输出,若格式说明个数多于输出项个数,格式控制字符串中右边多出的格式说明部分将输出与其类型对应的随机值。例如:

```
printf("%d %d",1,2,3); //输出结果为 1 2
printf("%d %d %d",1,2); //输出结果为 1 2 随机值
```

(3) 格式控制字符串可以分解成几个格式控制字符串。例如:

```
printf("%d%d\n",1,2); 等价于 printf("%d" "%d"\n",1,2);
```

(4) 在格式控制字符串中,两个连续的%只输出一个%。例如:

```
printf("%f %",1.0/6); //输出结果为 0.166667 %
```

(5) 格式说明与输出的数据类型要匹配,否则得到的输出结果可能不是原值。

(6) 格式字符串中的普通字符必须原样输入。例如:

```
scanf("a = %d,b = %d",&a,&b); //输入时应用如下形式:a = 3,b = 4 ✓
```

(7) 地址表列中的每一项必须为地址。例如:

```
scanf("a = %d,b = %d",&a,&b); //不能写成:scanf("a = %d,b = %d",a,b);
```

虽然在编译时不会出错,但是得不到正确的输入数据。

(8) 在用"%c"格式输入字符时,空格和转义字符都作为有效字符输入。例如:

```
scanf("%c%c%c",&ch1,&ch2,&ch3); //输入:A□✓
```

字符 A 赋给变量 ch1,空格赋给变量 ch2,回车赋给变量 ch3。

(9) 输入数据时不能指定精度。例如:

```
scanf("%lf,%lf",&x,&y); //不能写成:scanf("%8.3lf,%4lf",&x,&y);
```

(10) 输入数据时,遇空格、回车、跳格(Tab)、宽度结束或非法输入时该数据输入结束。

例如:

```
scanf("%d%c %lf",&a,&ch1,&x);    //输入:1234w12h.234
```

变量 a 的值为 1234,变量 ch1 的值为 w,变量 x 的值为 12.00。

由于遇空格数据输入结束,所以用 scanf()函数不能输入含有空格的字符串。

3.3 上机实践

1. 上机实践的目的要求

- (1) 进一步熟悉 C 语言的基本语句。
- (2) 掌握赋值语句的使用。
- (3) 掌握输入输出函数的格式及应用。

2. 上机实践内容

- (1) 格式符的使用。

```
#include "stdio.h"
main()
{   int a = 1234;
    float b = 123.456;
    printf("%2d, %2.1f",a,b );
}
```

程序运行结果为

1234,123.5

- (2) 整型数据和字符型数据在一定范围内可通用。

```
#include "stdio.h"
main()
{   int i = 65;
    char ch = 'A';
    printf("i = %d ch = %c\n",i,ch);
    printf("i = %c ch = %d\n",i,ch);
}
```

- (3) 格式输入输出函数的使用。

```
#include "stdio.h"
main()
{   int a,b,c;
    scanf("%d%d%d",&a,&b,&c);
    printf("a = %d, %d == %d",a,b,c);
}
```

运行时按以下方式输入 a,b,c 的值: 3□4□5 ✓

A = 3,4 == 5(输出 a,b,c 的值)

习题

一、选择题

1. putchar()函数可以向终端输出一个()。
 - A. 整型变量的值
 - B. 实型变量的值
 - C. 字符串
 - D. 字符或字符型变量的值
2. printf()函数中用到格式符%5s,其中,数字5表示输出的字符串占用5列,如果字符串长度大于5,则输出方式为();如果字符串长度小于5,则输出方式为()。
 - A. 从左起输出该字符串,右补空格
 - B. 按原字符长从左向右全部输出
 - C. 右对齐输出该字符串,左补空格
 - D. 输出错误信息
3. 已有定义“int a=-2”和输出语句“printf("%8lx",a);”,以下叙述正确的是()。
 - A. 整型变量的输出格式只有%d一种
 - B. %x是格式符的一种,它使用于任何一种类型数据
 - C. %x是格式符的一种,其变量的值按十六进制输出,但%8lx是错误的
 - D. %8lx不是错误的格式符,其中数字8规定了输出字符的宽度
4. 已有如下定义和输入语句,若要求a1,a2,c1,c2的值分别为10,20,A和B,当从第一列开始输入数据时,正确的数据输入方式是()。

```
int a1,a2;char c1,c2;
scanf("%d%c%d%c",&a1,&c1,&a2,&c2);
```

- A. 10A□20B ✓
 - B. 10□A□20□B ✓
 - C. 10A20B ✓
 - D. 10A20□B ✓
5. 已有如下定义和输入语句,若要求“a1,a2,c1,c2”的值分别为“10,20,A,B”,当从第一列开始输入数据时,正确的数据输入方式是()。

```
int a1,a2;char c1,c2;
scanf("%d%d",&a1,&a2);
scanf("%c%c",&c1,&c2);
```

- A. 1020AB ✓
 - B. 10□20 ✓ AB ✓
 - C. 10□□20□□B ✓
 - D. 1020□AB ✓
6. 有输入语句“scanf("a=%d,b=%d,c=%d",&a,&b,&c);”,为使变量a的值为1,b的值为3,c的值为2,从键盘输入数据的正确形式应当是()。
- A. 132 ✓
 - B. 1,3,2 ✓
 - C. a=1□b=3□c=2 ✓
 - D. a=1,b=3,c=2 ✓
7. 已有定义“int x; float y;”且执行“scanf("%3d%f",&x,&y);”语句时,从第一列开始输入数据,1234□678 ✓,则x的值为(),y的值为()。
- A. 1234
 - B. 123
 - C. 45
 - D. 345
 - E. 4.000000
 - F. 46.000000
 - G. 678.000000
 - H. 123.000000

8. 根据定义和数据输入方式,输入语句的正确形式为()。

已有定义: float f1,f2;

数据的输入方式: 4.52 ✓
3.5 ✓

- A. scanf("%f,%f",&f1,&f2); B. scanf("%f%f",&f1,&f2);
C. scanf("%3.2f %2.1f",&f1,&f2); D. scanf("%3.2f%2.1f",&f1,&f2);

9. 以下说法正确的是()。

- A. 输入项可以是一个实型常量,如“scanf("%f",=3.5);”
B. 只有格式控制,没有输入项,也可以进行正确的输入,如 scanf("a=%d,b=%d");
C. 当输入一个实型数据时,格式部分应规定小数点后的位数,如 scanf("%5.1f",&x);
D. 当输入一个数据时,必须指明变量的地址,如“scanf("%d",&x);”

二、程序设计题

1. 从键盘输入半径,计算圆的面积和周长,输出时要求取小数点后两位数字。
2. 输入一个华氏温度,要求输出摄氏温度,公式为 $c = 5(f - 32)/9$,输出时要求有文字说明。
3. 用 getchar()函数读入两个字符给 c1,c2,然后分别用 putchar()函数和 printf()函数输出这两个字符,并思考以下问题。
 - (1) 变量 c1,c2 应定义为字符型还是整型? 或二者皆可?
 - (2) 要求输出 c1 和 c2 值的 ASCII 码,应如何处理? 用 putchar()函数还是 printf()函数?
 - (3) 整型变量与字符型变量是否在任何情况下都可以互相代替? 如“char c1,c2;”与“int c1,c2;”是否无条件等价?