

## 第3章 顺序结构程序设计

C 语言程序是由 3 种基本结构组成的，其中最简单的结构就是顺序结构。不管是什么样的控制结构，在 C 语言中的具体实现操作都是通过语句完成的。按照程序的语句书写次序依次执行的程序结构即为顺序结构。

### 3.1 C 语 句

C 程序对数据的处理是通过语句的执行来实现的，一条语句完成一项操作或功能。而构成 C 程序的函数体内应包含若干条语句，以实现相应的功能。

#### 3.1.1 变量定义语句

C 语言规定：任何变量要先定义后使用。如果不定义就使用，则程序将会出错。定义变量的标准形式：

变量类型 变量名 1, 变量名 2, ..., 变量名 n;

例如：

```
int a,b;          /*定义 a, b 为整型变量*/  
float x;         /*定义 x 为单精度实型变量*/  
char ch;        /*定义 ch 为字符型变量*/  
double y;       /*定义 y 为双精度实型变量*/
```

#### 3.1.2 表达式语句

由表达式组成的语句称为表达式语句。语句格式：

表达式;

即在表达式的尾部加上一个分号“;”构成。

功能：

计算表达式或改变变量的值。

表达式语句根据功能的不同又分为普通表达式语句和赋值语句。

##### (1) 普通表达式语句

普通表达式语句就是在普通的常量、变量、函数及由它们构成的表达式后加一个分号构成。例如，9;、a+b;、x++;、a=3,b=8;、sin(x);等。

##### (2) 赋值语句

赋值语句是在赋值表达式的尾部加上一个分号“;”构成的。

例如, $x=y+z$	赋值表达式
$x=y+z;$	赋值语句
$x=2$	赋值表达式
$x=2;$	赋值语句

C 语言的赋值语句在程序设计中可以实现给变量赋初值,它是程序设计中的基本可执行语句。

### 3.1.3 复合语句

C 语言的复合语句的形式:

```
{ 语句 1 语句 2 ..., 语句 n }
```

其作用就是将 C 语言的若干语句用一对花括号括起来构成一个语句组。

例如:

```
{ a=b;a++;c=a*b;printf("c=%d\n",b); }
```

说明:

- ① 复合语句的花括号内的语句数量不限。
- ② 复合语句的花括号内的语句可以不在一行,但花括号不能缺省,必须成对出现。
- ③ 复合语句的花括号内可以有定义语句,但定义语句必须置于所有可执行语句之前。

### 3.1.4 空语句

分号“;”是 C 语句最重要的一个组成成分,它置于 C 语句的最后。而分号本身在 C 语言中也可以单独构成一个语句,由分号单独构成的语句即为空语句,它不产生任何操作。

例如:

```
main()  
{  
;  
}
```

是一个合法的程序。

说明:

- ① 程序设计时有时需要加一条空语句来表示某条语句的存在。
- ② 随意加空语句有时候会造成逻辑上的错误,需慎重。

## 3.2 数据的输入和输出

输入/输出操作是程序的两项基本操作。任何一个程序只要是实现对数据的加工处理,就必然需要有输入数据,同时加工结果也要通过输出告知外界。

C 语言本身没有提供专门的输入和输出语句,它的输入/输出操作均是由 C 的标准函数

实现。而这些函数均被定义在 `stdio.h` 的头文件中，因此在 Turbo C 及 VC6.0 中要使用要输入/输出函数时，一定要在源程序中使用包含头文件 `stdio.h`：

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

### 3.2.1 printf 函数（格式输出函数）

`printf` 函数是 C 语言提供的标准格式输出函数，用来在终端上（显示器或打印机）按指定格式进行输出。

(1) `printf` 函数的一般格式：

```
printf("格式控制", 输出项 1, 输出项 2, ..., 输出项 n)
```

它的作用是按照格式控制符的指定格式，将各输出项在终端设备上输出。

例如：

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

其中，“格式控制”中的 `a=`、`,`和 `b=`为普通字符，`%d`为格式说明，而 `a,b`为两个输出项，它们分别与“格式控制”中的两个`%d`格式说明相对应。

说明：

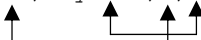
① “格式控制”中的普通字符（如上例中的 `a=`、`,`和 `b=`）原样输出。在“格式控制”中加上普通字符是为了使输出结果更为完整。

② “格式控制”中的格式符是为后面的输出项指定输出格式。格式说明由“%”开头，后面跟一个格式符。不同的数据类型其格式符不同。

③ 各输出项必须与“格式控制”中的格式说明个数和对应数据类型相容。

例如：

```
int x;
float y;
x=4;
y=x+4;
printf("x=%d,x+y=%f", x, x+y);
```



④ `printf` 函数的调用则必须在其后加上一个分号“;”，使其变成表达式语句才能执行。

(2) `printf` 函数的常用格式说明

C 语言的标准输入函数 `scanf` 和标准输出函数 `printf` 均需要使用格式说明来完成数据的输入和输出。表 3-1 为 C 语言允许使用的格式符和它们的作用。

表 3-1 格式符及其作用

格式符	作用说明
d (或 i)	输出带符号的十进制整数
o	输出八进制无符号整数（不输出前导 0）

续表

格式符	作用说明
u	输出无符号的十进制整数
x (或 X)	输出无符号的十六进制的整数 (不输出前导 0x)
f	输出带小数点的单精度数和双精度数
e (或 E)	以指数形式输出单精度数和双精度数
g (或 G)	系统自动选定%f和%e输出宽度较小的一种,输出单精度数和双精度数
c	输出单个字符
s	输出字符串,遇到'\0'结束输出
%	输出一个%

例如:

```

int a;
float x;
char ch;
a=16;
x=12.3;
ch='D';
printf("a=%d",a) /* 输出结果: a=16*/
printf("a=%o,a=%x",a,a) /* 输出结果: a=20,a=10*/
printf("x=%f,x=%e",x,x) /* 输出结果: x=12.300000,x=1.23e+001*/
printf("x=%g",x) /* 输出结果: x=12.3*/
printf("ch=%c",ch) /* 输出结果: ch=D*/
printf("%s","HELLO\0How Are you! ") /* 输出结果 HELLO*/

```

在%和格式符之间还可以插入诸如“宽度说明”、“左对齐符-”等格式附加符,如表 3-2 所示。

表 3-2 格式附加符

格式附加符	作用说明
l	输出长整型数据或双精度数据,可作用在 d、o、x、u、f、e、g 前
负号-	以左对齐的方式输出数据
整数 m	确定数据的输出宽度。若 m>数据宽度,则以右对齐的方式输出,左端补空格;若 m<数据宽度,则按数据的实际宽度输出
.n (n 为一个整数)	输出的实数保留 n 位小数,第 n+1 位四舍五入;或截取字符串的前 n 位字符

**知识的延伸:**

printf 函数中可以取消格式说明吗?

在 C 语言中 printf 函数实际上还有一个比较单纯的应用,即实现对字符串的输出,此时可以取消格式说明,仅有一个字符串。这种格式常常用于输出提示信息。

例如:

```
#include "stdio.h"
main( )
{
    printf("How are you!");
}
```

程序的运行结果: How are you!

```
#include "stdio.h"
main( )
{
    printf("I am a");
    printf("student");
}
```

程序的运行结果: I am a student

同时在字符串中可以使用转义字符。

例如:

```
#include "stdio.h"
main( )
{
    printf("Hello!\n");
    printf("How are you!");
}
```

程序的运行结果: Hello!

How are you!

**思维拓展:**

① 对照表 2-2, 你能写出下列程序的运行结果吗?

```
main()
{
    printf(" ab c\t de\rftg\n");
    printf("h\ti\b\bj k");
}
```

② 若 printf() 函数中格式符与输出项对应出错或数据类型出现错误, 如以下 3 种情况: 假设 int i,j;

```
float x;
i=3, j=4, x=5.0;
```

① printf("%d,%d",i);

② printf("%d",i,j);

③ printf("%d,%f",x,i);

程序运行时会出现什么情况?

### 3.2.2 scanf 函数（格式输入函数）

scanf 函数是 C 语言提供的最常用的标准输入函数，在其尾部加上“;”则构成输入语句。scanf 函数的一般格式：

```
scanf("格式控制", &变量名 1, &变量名 2, ..., &变量名 n)
```

它的作用是将从键盘上输入的数据，按照格式控制符的指定格式放入相应的变量中。

例如：

```
scanf("x=%d,y=%d",&x,&y);
```

其中，“格式控制”中的“x=”和“y=”为普通字符，而%d为格式说明，而&为 C 语言的地址运算符，所以&x,&y是x和y这两个变量的地址，它们分别与“格式控制”中的两个%d格式说明相对应。

说明：

① “格式控制”中的普通字符（如上例中的 x=、,和 y=）要原样输入。不过在“格式控制”中最好不要加普通字符，以免输入时因粗心而造成不必要的错误。

② “格式控制”中的格式符是为后面的变量指定输入格式。格式说明由“%”开头，后面跟一个格式符。scanf()的格式符与printf()的基本一致，只是没有u和g格式符且双精度变量值的输入要用格式符lf或le。

③ “格式控制”若有多个连续的格式说明符，则除%c和%s外，其他的格式说明符以回车、空格或tab键作为分隔符。

例如：

```
int a,b;
float x,y;
scanf("%d%d%f%f",&a,&b,&x,&y);
```

则输入数据时可以输完每个数据后回车，如：

```
1
23
6.7
8
```

保证 a 得 1, b 得 23, x 得 6.7, y 得 8.0

也可以输完每个数据后单击空格键，如：

```
1 23 6.7 8
```

保证 4 个变量得到同样的值。若使用空格符作为分隔符，则空格数可以不止 1 个。

④ 变量地址必须与“格式控制”中的格式说明个数、对应数据类型一致。

例如：

```
float x;
double y;
scanf("%f%lf",&x,&y);
```

**思维拓展:**

若 scanf()函数中变量名前漏了符号&:

假设 int i,j;

```
scanf("%d,%d",&i,j);
```

程序运行时会出现什么情况?

### 3.3 顺序结构程序案例

**【案例 3.1】** 求任意两个数之和。

案例分析:

求两个数之和,显然需要用到 2 个变量,为了使得整个程序流程清晰,可以再定义一个变量装和值。同时数据应该是实数,为了节省空间,在此定义其数据类型为 float。由于题目中没有给出具体是哪两个数,所以这两个变量值的输入要利用函数 scanf 实现,再利用 printf 函数实现输出。

具体程序如下:

```
# include "stdio.h"
main( )
{
    float x,y,z;
    printf("Enter x and y:\n");
    scanf("%f%f",&x,&y);
    z=x+y;
    printf("z=%f\n",z);
}
```

程序的运行情况:

```
Enter x and y:
2.3 6.7<CR>
c=9.000000
```

**【案例 3.2】** 交换两个整型变量 a、b 的值。

案例分析:

要想交换两个变量 a 和 b 的值,不能简单地用 a=b;b=a;两条语句实现。因为当执行了语句 a=b;后, a 中原有值就被 b 的值替换, a 的值就丢失了,从而无法实现两个数值的交换。为了不丢失 a 中原有的值,必须在执行 a=b;之前,将 a 的值保存到一个临时变量中,在执行了 a=b;后再将临时变量的值赋给 b。

具体程序如下:

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

```
int a,b,c;
printf("enter a and b:\n");
scanf("%d%d",&a,&b);
printf("a=%d,b=%d",a,b);
c=a;
a=b;
b=c;
printf("a=%d,b=%d",a,b);
}
```

程序的运行情况:

```
enter a and b:
15 51<cr>
a=15,b=51
a=51,b=15
```

**【案例 3.3】** 求 17 除以 4 的余数。

案例分析:

因除数和被除数都很清楚,所以只需定义 1 个变量  $x$  用来装余数就行了。利用 C 语言提供的求余运算符 %。

具体程序如下:

```
# include "stdio.h"
main()
{
    int x;
    x=17%4;
    printf("x=%d",x);
}
```

程序的运行情况:

```
x=1
```

**【案例 3.4】** 求整数  $m$  除以  $n$  的余数。

案例分析:

因除数和被除数需要随机输入,同时还要求余数,所以需定义 3 个变量。利用 C 语言提供的求余运算符 %。

具体程序如下:

```
# include "stdio.h"
main()
{
    int m,n,t;
    printf("enter m and n:\n");
    scanf("%d%d",&m,&n);
    t=m%n;
```



```
printf("t=%d",t);
}
```

程序的运行情况:

```
enter m and n:
17 3<cr>
t=2
```

**【案例 3.5】** 任意输入一个小写英文字母，输出其大写英文字母。

案例分析:

由于要输入小写字母，输出大写字母，所以需要两个字符型变量。要注意在 ASCII 码表中大写英文字母的码值比小写英文字母小 32。比如，'a' 为 97，'A' 为 65；'c' 为 99，'C' 为 67，所以只需用小写的 ASCII 值减去 32 即为大写字母。

具体程序如下:

```
# include "stdio.h"
main()
{
char ch1,ch2;
printf("Enter ch1:\n");
scanf("%c",&ch1);
ch2=ch1-32;
printf("ch1=%c,ch2=%c",ch1,ch2);
}
```

程序的运行情况:

```
Enter ch1:
h<cr>
ch1=h,ch2=H
```

**知识的延伸:**

C 语言中是否只有 printf 函数和 scanf 函数实现数据的输入和输出呢?

C 语言中 printf 函数和 scanf 函数是最为通用的格式输出和格式输入函数，但它们不是唯一的。比如，单个字符的输入和输出在 C 语言中有专门的函数实现，它们是 getchar 函数和 putchar 函数

getchar() 为单个字符的输入函数。

一般调用形式:

```
变量= getchar( );
```

功能: 从键盘上输入一个字符，赋给赋值号左侧的变量。

putchar() 为单个字符的输出函数。

一般调用形式:

```
putchar(变量);
```

功能: 将变量的值输出到显示器上并换行。

`getchar()`和 `putchar()`函数均被定义在 `stdio.h` 的头文件中。

用 `getchar` 函数和 `putchar` 函数改写案例 3.5。

具体程序如下:

```
# include "stdio.h"
main()
{
char ch1,ch2;
printf("Enter ch1:\n");
ch1=getchar( );
ch2=ch1-32;
putchar(ch1); putchar(ch2);
}
```

程序的运行情况:

```
Enter ch1:
h<cr>
hH
```

### 3.4 顺序结构程序设计实验指导

#### 1. 实验目的

- ① 理解顺序结构程序设计的基本思想。
- ② 熟练掌握各种数据类型的输入/输出格式符。
- ③ 掌握 C 语言程序设计中最重要的一种语句——赋值语句的使用方法。
- ④ 熟练掌握格式输入与格式输出函数的使用。
- ⑤ 进一步练习 C 程序的输入、编译连接与运行的过程。

#### 2. 实验内容

(1) 计算并输出两数的和

用 `scanf` 函数输入两个整数 `x`, `y`, 然后用一个赋值语句计算这两个数的和 `z`, 最后用格式输出函数 `printf` 输出 `z` 的值。

具体要求如下:

- ① 所有变量为整数。
- ② 输入前要有提示。
- ③ 输出结果时要有文字说明。

(2) 计算并输出面积、体积。

设一个圆柱的底面半径 `r` 为 2.5 厘米, 高 `h` 为 3.5 厘米。分别计算并输出该圆柱体的底



- A. 15                      B. 15,16                      C. 15 16 17                      D. 15 16  
                                  16    17    17  
                                  17

3.7 以下程序段的输出结果是 ( )。

```
#include "stdio.h"
main()
{
    int a=1,b=0;
    printf("%d,",b=a+b);
    printf("%d\n",a=2*b);
}
```

- A. 0,0                      B. 1,0                      C. 3,2                      D. 1,2

3.8 有以下程序, 其中 k 的初值为八进制数:

```
#include
main()
{
    int k=011;
    printf("%d\n", k++);
}
```

程序运行后的输出结果是 ( )。

- A. 9                      B. 10                      C. 11                      D. 12

3.9 若变量 x、y 已正确定义并赋值, 以下符合 C 语言语法的表达式是 ( )。

- A. ++x,y=x--                      B. x+1=y                      C. x=x+10=x+y                      D. double(x)/10

3.10 有以下程序

```
main()
{
    int x, y, z;
    x=y=1;
    z=x++,y++,++y;
    printf("%d,%d,%d\n",x,y,z);
}
```

程序运行后的输出结果是 ( )。

- A. 2,3,3                      B. 2,3,2                      C. 2,3,1                      D. 2,2,1

3.11 有下列程序:

```
main()
{
    char a1='M',a2='m';
    printf("%c\n", (a1,a2));
}
```

程序运行后的输出结果是 ( )。

- A. M    B. m

C. 格式说明符不足, 编译出错

D. 程序运行时产生出错信息

## 二、上机实战

3.12 编程求  $30^\circ$  的弧度值。

3.13 编程计算任意两个数之积。

3.14 输入一个3位正整数, 分别输出它的个位、十位和百位数字。

3.15 编程求  $15/4$  的商。

3.16 输入一个华氏温度, 要求输出摄氏温度, 公式为:  $9c=5(F-32)$ 。其中,  $c$  代表摄氏温度,  $F$  代表华氏温度。

电子工业出版社版权所有  
盗版必究