第 3 章　顺序结构程序设计

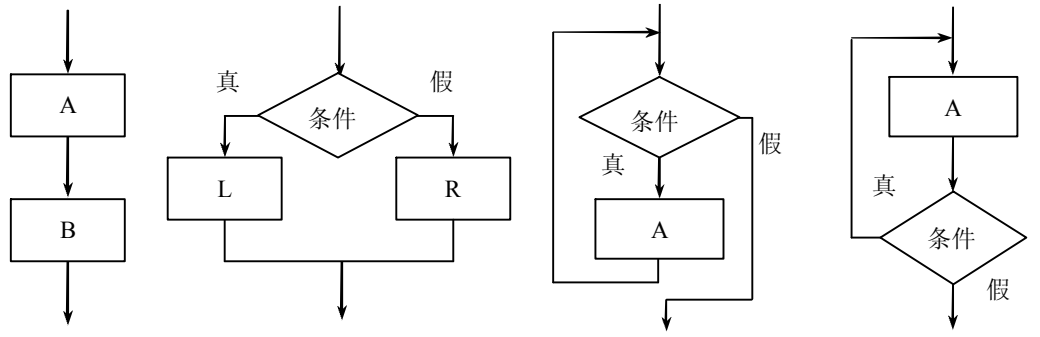
随着计算机应用的不断发展，软件变得越来越复杂。软件的可读性、可理解性等问题变得十分突出。经过不断的研究和实践，确定了程序设计的基本方法，这就是结构化程序设计方法。结构化程序设计方法使得程序的逻辑结构清晰，层次分明，有效地改善了程序的可靠性，提高了软件的质量。本章所介绍的语句，将按它们在程序中出现的顺序逐条执行，由这样的语句构成的程序结构称为顺序结构。本章主要介绍以下内容：

- 结构化程序的三种基本结构
- C 语言的语句
- 赋值语句
- 数据的输入与输出
- 顺序结构程序设计

3.1　C 程序概述

3.1.1　结构化程序设计

近年来广泛采用结构化程序设计方法，使程序结构清晰、易读性强，以提高程序设计的质量和效率。结构化程序设计的基本思想是：用顺序结构、选择结构和循环结构这三种基本结构来构造程序。结构化程序的三种基本结构如图 3-1 所示。



（a）顺序结构　（b）选择结构　（c）当型循环　（d）直到型循环

图 3-1　结构化程序的三种基本结构

（1）顺序结构。程序执行流程是按语句顺序依次执行。先执行 A 操作，再执行 B 操作，两者是顺序执行的关系。

（2）选择结构。根据给定的条件进行判断，由判断结果决定执行程序中的哪一个分支。当条件成立时执行 L 操作，否则执行 R 操作。L 或 R 只能执行之一。

（3）循环结构。在给定的条件成立的情况下，反复执行某个程序段。有两种循环结构：

1）当型循环结构。当条件成立时，反复执行 A 操作。直到条件不成立时才停止循环。

2）直到型循环结构。先执行 A 操作，再判断条件，若条件成立，再执行 A 操作，如此反复，直到条件不成立时才停止循环。

由上述三种程序结构组成的程序称作结构化程序，形成的软件称作结构化软件。

3.1.2　C 程序结构

C 语言以文件为编译单位。一个 C 程序可以由一个或若干个源程序文件构成，一个源程序文件可以由一个或若干个 C 函数组成，一个函数由数据定义部分和执行语句组成。C 程序结构如图 3-2 所示。语句是 C 程序的基本组成部分，它用来向计算机系统发出操作指令。一个语句经编译后产生若干条机器指令。

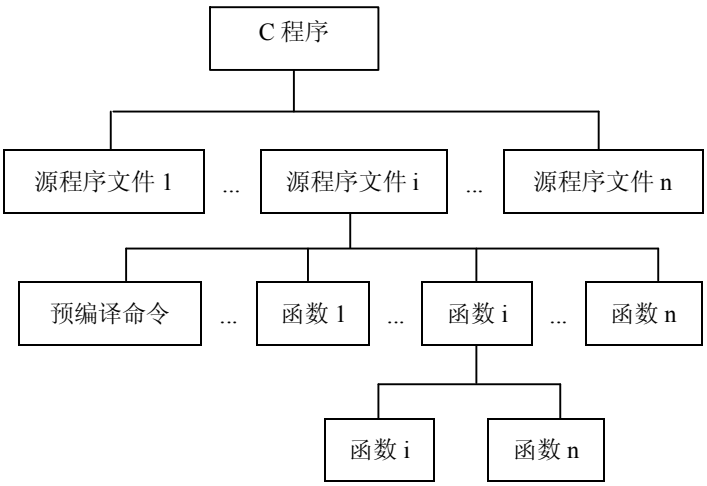


图 3-2　C 程序结构

3.1.3　C 语言的语句

在 C 语言中，一个表达式的后面跟一个分号就构成了语句。如：

```
a=3;
x=15+a/b;
```

等都是语句。这种语句称为表达式语句。注意，C 语言中的分号（;）是语句的终结符，而不是语句之间的分隔符。换句话说，分号是 C 语句的最末组成部分。C 语句可以分为四类：

1．控制语句

控制语句完成一定的控制功能。如：

```
if(a==b)x=a;else x=a/b;
```

就是一个条件语句。当 a 与 b 相等时，将 a 的值赋给 x，否则，将 a/b 的值赋给 x。

C 语言的控制语句分流程控制语句和辅助控制语句，共有 9 种，它们是：

（1）if…else…　　条件语句
（2）for…　　循环语句
（3）do…while…　　循环语句
（4）while…　　循环语句
（5）switch…case…　　多分支选择语句

（6）continue　　结束本次循环语句
（7）break　　终止执行 switch 或循环语句
（8）goto　　转向语句
（9）return　　从函数返回语句

2. 表达式语句

表达式语句指由表达式加分号构成的一个语句。例如：

```
a=3
```

是一个赋值表达式，而

```
a=3;
```

是一个赋值语句。

任何一个表达式都可以加上分号而成为语句，例如：

```
++n;
```

是一个语句，作用是使 n 值加 1。又如：

```
a+b/c;
```

也是一个语句，作用是完成 a+b/c 的操作，它是合法的，但是并不把 a+b/c 的值赋给另一变量，所以它并无实际意义。

因为函数调用也属于表达式的一种，所以，由一次函数调用加一个分号也构成一个语句，例如：

```
scanf("%d%d",&a,&b);
printf("a=%d,b=%d\n",a,b);
```

都是函数调用语句，作用分别是调用输入函数给变量 a 和 b 赋值；调用输出函数输出变量 a 和 b 的值。

3. 空语句

空语句指只有一个分号的语句。如：

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

这个分号也是一条语句，称为“空语句”，空语句在语法上占一个语句的位置，但是它不具备任何执行功能。空语句有时用作转向点，或循环语句中的循环体。如：

```
for(j=0;j<10;j++);
```

在这样的循环中，循环体是空语句，表示循环体本身什么也不做。在实时控制中，它经常用来实现延时功能。

4. 复合语句

复合语句指由花括号括起来的语句。其一般形式如下：

```
{说明部分;语句部分;}
```

其中说明部分是一系列用分号隔开的说明，包括变量说明等；语句部分是由一系列可执行语句组成的。例如：

```
{   int x,y,z;
    z=(x+y)/(x-y);
    printf("%d",z);
}
```

是一个复合语句。注意，复合语句是以左右花括号括起来的，因此，在复合语句中左右花括号是必需的，且右面的花括号后不必加分号。

一个复合语句在语法上等同于一个语句，因此，凡是单个语句能够出现的地方都可以出现复合语句。复合语句作为一个语句又可以出现在其他复合语句的内部。

从 C 语言提供的语句可以看出，C 语言具有与结构化程序设计相对应的三种语句，所以，C 语言支持结构化程序设计的三种结构。因此，用 C 语言编制的程序，模块之间界面清晰，易读性好，能够有效地提高程序的可靠性。

例 3.1　顺序结构举例。

```
#include   "stdio.h"
main( )
{   int a=5,b=6;
    printf("a=%d\n",a);
    {   int c=7;
        printf("c=%d\n",c);
    }
    printf("b=%d\n",b);
}
```

运行结果为：

```
a=5
c=7
b=6
```

可以看出，程序是按语句顺序依次执行。其中，

```
{   int c=7;
    printf("c=%d\n",c);
}
```

是复合语句。复合语句在语法上等价于一个语句。

3.2　赋值语句

在赋值表达式的尾部加上一个分号（;）号，就构成了赋值语句，也称表达式语句。例如：

```
a=b+c
```

是赋值表达式，而

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

则是赋值语句；再如：

```
i=l,j=2
```

是逗号表达式，而

```
i=(l,j=2);
```

则是一条赋值语句。赋值语句是一种可执行语句，应当出现在函数的可执行部分。在程序中要注意把赋值语句和赋值表达式区别开来。例如：

```
if((a=b)>0)t=a;
```

按语法规定 if 后面的括号内是一个条件表达式，例如可以是：“if(x>0)…”。现在在 x 的位置上换上一个赋值表达式“a=b”，其作用是：先进行赋值运算，将 b 的值赋给 a，然后判断表达式“a=b”是否大于 0，如大于 0，执行 t=a。在 if 语句中的“a=b”不是赋值语句而是赋值表达式，这样写是合法的。如果写成

```
if((a=b;)>0)t=a;
```

就错了。在 if 的条件中不能包含赋值语句。

C 语言中可由形式多样的赋值表达式构成赋值语句，用法灵活，因此读者首先应当掌握好赋值表达式的运算规律才能写出正确的赋值语句。

3.3 数据的输出

把数据从计算机内部送到计算机的外部设备上的操作称为“输出”。数据的输出是以终端（即系统隐含指定的输出设备）为对象。

C 语言本身不提供用于输入和输出的语句。在 C 语言程序中，可以通过调用标准库函数提供的输入和输出函数来实现数据的输入和输出。本节介绍两个最基本的输出函数，它们一般都以终端显示器为处理对象。

3.3.1 字符输出函数 putchar

putchar 函数的作用是向终端显示器输出一个字符。其一般调用形式如下：

```
putchar(ch);
```

其作用是将变量 ch 中的字符输出到屏幕上当前光标的位置。其中 ch 可以是字符型变量，也可以是整型变量。在使用标准 I/O 库函数时，要用预编译命令#include 将 stdio.h 文件包括到用户源文件中。即

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

stdio.h 是 standard input&output 的缩写，它包含了与标准 I/O 库有关的变量定义和宏定义。在需要使用标准 I/O 库中的函数时，应在程序前使用上述预编译命令，但 printf 函数和 scanf 函数例外。

例 3.2 输出字符 A（变量为整型）。

```
#include "stdio.h"
main()
{   int c;
    c=65;
    putchar(c);
}
```

例 3.3 输出字符 A（变量为一字符）。

```
#include   "stdio.h"
main()
{   char c;
    c='A';
    putchar(c);
}
```

这两个程序的运行结果都是显示变量 c 的值——字母 A。例 3.2 中变量 c 是整型变量，例 3.3 中变量 c 是字符型变量。

例 3.4 输出字符串“Hello”。

```
#include "stdio.h"
main()
{   char a='H',b='e',c='l',d='l',e='o';
    putchar(a); putchar(b); putchar(c); putchar(d); putchar(e);
}
```

运行结果为：

```
Hello
```

putchar 函数也可以输出控制字符和其他转义字符，如：

```
putchar('\n');
```

输出一个换行符。

```
putchar ("\101");
```

输出字符'A'。

3.3.2　格式输出函数 printf

putchar 函数只能输出一个字符，而不能输出整型和实型数据。printf 函数是 C 语言提供的标准输出函数，它的作用是在终端设备上按指定格式进行输出。它可以用来输出 C 语言中任意类型的数据，而且可以同时输出多个同类型或不同类型的数据。

printf 函数的一般调用形式如下：

```
printf (格式控制,输出表列);
```

1．格式控制

“格式控制”部分是由双引号括起来的字符串，也称“转换控制字符串”，它主要包括两种信息：格式说明符和普通字符。

（1）普通字符，即需要按原样输出的字符。如：

```
printf("The C Language");
```

运行结果为输出下列字符串：

```
The C Language
```

其中，“The C Language”是普通字符。

（2）格式说明符是由“%”和格式字符组成，如，%d、%c 和%f 等，其中%是格式表示符，d、c 和 f 是格式控制符，也称为格式字符。它的作用是将要输出的数据转换为指定的格式输出。格式说明总是由“%”字符开始。格式说明的一般形式为：

%+/-0m.nl 格式字符

其中+、-、0、m、n、l 称为附加格式说明符，说明输出数据精度，左右对齐等。

①printf 函数中使用的格式字符主要包括如下几个：

- d：以十进制形式输出带符号的整型数，输出长整型数据时，使用 ld。
- o：以八进制无符号形式输出整型数（不带前导 0），输出长整型数据时，使用 lo。
- x：以十六进制无符号形式输出整型数（不带前导 0x 或 0X），输出长整型数据时，使用 lx。
- u：以无符号的十进制形式输出整型数，输出长整型数据时，使用 lu。
- c：以字符形式输出一个字符。
- s：输出以 '\0' 结尾的字符串。
- f：以十进制小数形式输出单精度数和双精度数，隐含输出 6 位小数。
- e：以指数形式输出单精度数和双精度数，数字部分小数位为 6 位。
- g：由系统决定采用%f 格式还是采用%e 格式，以使输出宽度最小。

②printf 函数中使用的附加格式字符主要包括如下几个：

- l：表示输出的是长整型整数，可以加在 d、o、x、u 前面。
- m：表示输出数据的最小宽度。

- .n：对实数，表示输出 n 位小数；对字符串，表示截取 n 个字符。
- 0：表示左边补 0。
- +：转换后的数据右对齐。
- -：转换后的数据左对齐。

2. 输出表列

printf 函数中的“输出表列”是一些表达式，可以是常量、变量等，这些表达式应当与“格式控制”字符串中的格式说明符的类型一一对应，如果“输出表列”中有多个变量，则每个变量之间应由逗号隔开。

例 3.5

```
#include   "stdio.h"
main()
{   char a='A';   int b=10;   float c=1.23;
    printf ("a=%c,b=%d,,c=%f\n",a,b,c);
}
```

运行结果为：

```
a=A, b=10, c=1.230000
```

例 3.6　求算术运算的值。

```
#include   "stdio.h"
main()
{   int a,b;
    a=10; b=5;
    printf ("a=%d,b=%d\n",a,b);
    printf ("a+b=%d\na-b=%d\na*b=%d\na/b=%d\n",a+b,a-b,a*b,a/b);
}
```

运行结果为：

```
a= 10, b=5
a+b=15
a-b=5
a*b=50
a/b=2
```

3. 输出数据所占的宽度

当使用%d、%c、%f、%e、……格式说明时，输出数据所占的宽度由系统决定（通常取数据本身的宽度，不加空格），并采用右对齐的形式。可以用附加格式字符人为控制输出数据所占的宽度。

（1）%md：m 为指定输出字段的宽度。如果数据的位数大于 m，则按实际位数输出，否则输出数据右对齐，左边补以空格。如：%4d。

（2）%mc：m 为指定输出字段的宽度。如果 m 大于一个字符的宽度，则输出时向右对齐，左边补以空格。如：%3c。

（3）%ms：m 为输出字符串所占的列数。如果字符串的长度大于 m，则按字符串的本身长度输出，否则，输出时，字符串向右对齐，左边补以空格。如：%8s。

（4）%-ms：m 为输出字符串所占的列数。如果字符串的长度大于 m，则按字符串的本身长度输出，否则，输出时，字符串向左对齐，右边补以空格。如：%-8s。

（5）%m.nf：m 为实型数据所占的总列数（包括小数点），n 为小数点后面的位数。如果数据的总位数小于 m，则输出时向右对齐，左边补以空格。如：%5.2f。

（6）%-m.nf：m 为实型数据所占的总列数（包括小数点），n 为小数点后面的位数。如果数据的长度小于 m，则输出时向左对齐，右边补以空格。如：%-5.2f。

4. 调用 printf 函数时的注意事项

（1）在格式控制串中，格式说明与输出项从左到右在类型上必须一一对应匹配。如不匹配，将导致数据不能正确输出，这时，系统并不报错。特别要提醒读者的是：在输出 long 整型数据时，一定要使用%ld 格式说明，如果遗漏了字母 l，只用了%d，将输出错误的数据。如例 3.2 中%c 与 a 对应，%d 与 b 对应，%f 与 c 对应。

（2）在格式控制串中，格式说明与输出项的个数应该相同。如果格式说明的个数少于输出项的个数，多余的输出项不予输出；如果格式说明的个数多于输出项的个数，则对于多余的格式将输出不定值（或 0 值）。

（3）在格式控制串中，除了合法的格式说明外，可以包含任意的合法字符（包括转义字符），这些字符在输出时将“原样照印”。如例 3.2 中的“a=”、“b=”、“c=”和双引号里的逗号。

（4）如果需要输出百分号%，则应该在格式控制串中用两个连续的百分号%%来表示。

（5）格式字符必须用小写字母，如%d 不能写成%D。

（6）printf 函数的返回值通常是本次调用中输出字符的个数。

例 3.7　printf 函数的使用。

```
main()
{   int j=1234;
    float f=12345678.901234;
    char c='c';
    char s[5]="test";
    printf("12345678901234567890123456789O\n\n");
    printf("%%d             ##%d\n",j);
    printf("%%2d            ##%2d\n",j);
    printf("%%10d           ##%10d\n",j);
    printf("%%-10d          ##%-10d\n\n",j);
    printf("%%f             ##%f\n",f);
    printf("%%4.4f          ##%4.4f\n",f);
    printf("%%10.10f        ##%10.10f\n",f);
    printf("%%-10.10f       ##%-10.10f\n\n",f);
    printf("%%c             ##%c\n",c);
    printf("%%2c            ##%2c\n",c);
    printf("%%10c           ##%10c\n",c);
    printf("%%-10c          ##%-10c\n\n",c);
    printf("%%s             ##%s\n",s);
    printf("%%2s            ##%2s\n",s);
    printf("%%10s           ##%10s\n",s);
    printf("%%-10s          ##%-10s",s);
}
```

程序运行结果为：

```
12345678901234567890123456789O

%d          ##1234
%2d         ##1234
%10d        ##         1234
```

```
%-10d       ##1234

%f          ##12345679.000000
%4.4f       ## 12345679.0000
%10.10f     ##12345679.0000000000
%-10.10f    ##12345679.0000000000

%c          ##c
%2c         ## c
%10c        ##         c
%-10c       ##c

%s          ##test
%2s         ##test
%10s        ##      test
%-10s       ##test
```

3.4 数据的输入

数据的输入是以终端（或系统隐含指定的输入设备）为处理对象。本节介绍两个输入函数，它们一般以终端键盘为输入设备。

3.4.1 字符输入函数 getchar

此函数的作用是接收从键盘输入的一个字符。当程序执行到 getchar 函数时，将等待用户从键盘输入一个字符，然后程序再继续执行。函数的值就是从键盘得到的字符。getchar 函数没有参数，其一般形式为：

```
ch=getchar();
```

其中 ch 是字符型或整型变量，它将接收从键盘输入的一个字符。getchar()是有回显的字符输入函数，也就是说当执行到此函数时输入的字符自动显示在终端上。

例 3.8 从键盘输入一个字符，并把它显示出来。

```
#include   "stdio.h"
main()
{   char x;          /*或 int x;*/
    x=getchar();
    putchar(x);
}
```

在运行时，如果从键盘输入字符'A'：

```
A <CR>     （输入'A'后，按“回车”键）
A          （输出变量 x 的值'A'）
```

注意 getchar 函数只能接收一个字符。函数得到的字符可以赋给一个字符变量或整型变量，也可以不赋给任何变量，只作为表达式的一部分。如上例可以改为

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

```
    putchar(getchar());
}
```

因为 getchar()的值为'A'，因此输出'A'。也可以用 printf 函数：

```
printf("%c",getchar());
```

和 putchar 函数一样，在一个程序中要用 getchar 函数时，应该在程序的前面用

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

将"stdio.h"文件包括到用户源程序文件中。

3.4.2　格式输入函数 scanf

getchar 函数只能用来输入一个字符，而不能输入整型和实型数据。scanf 函数是 C 语言提供的标准输入函数，它的作用是在终端设备上按指定格式进行输入。它可以用来输入 C 语言中任意类型的数据，而且可以同时输入多个同类型的或不同类型的数据。

scanf 函数的一般调用形式如下：

```
scanf (格式控制,地址表列);
```

1．格式控制

“格式控制”部分与printf函数相似，也是由双引号括起来的字符串，它主要是由“%”和格式字符组成，中间可以插入附加格式字符，如，%d、%3d、%c和%f等，它的作用是将要输入的数据转换为指定的格式存入到由地址表列所指向的相应的变量中。

（1）scanf 函数中使用的格式字符主要包括如下几个：

d：用来输入带符号的十进制形式整数，输入长整型数据时，使用 ld；输入短整型数据时，使用 hd。

u：用来输入无符号的十进制形式整数。

o：用来输入八进制无符号形式整数（可带前导 0，也可不带），输入长整型数据时，使用 lo；输入短整型数据时，使用 ho。

x：用来输入十六进制无符号形式整数（可带前导 0x 或 0X，也可不带），输入长整型数据时，使用 lx；输入短整型数据时，使用 hx。

c：用来输入单个字符。

s：用来输入字符串。以非空格字符开始，以第一个空字符结束。

f：用来输入实数，以带小数点的形式或指数形式输入。

e：用来输入实数，与 f 的作用相同。

（2）scanf 函数中使用的附加格式字符主要包括如下几个：

l：表示输入的是长整型整数或 double 型数据，可以加在 d、o、x、f、e 前面。

h：表示输入的是短整型整数，可以加在 d、o、x 前面。

m：表示输入数据的最小宽度。

*：表示本输入项在读入后不赋给相应的变量。

注意输入数据时不能规定精度。

2．地址表列

scanf 函数中的“地址表列”部分是由变量的地址组成，如果“地址表列”中有多个变量，则每个变量之间应由逗号隔开。在 C 语言中，变量的地址可由取地址运算符“&”得到，如变量 x 的地址可以写为&x。

例 3.9 从键盘输入一个字符、一个整型数和一个实型数，并显示出来。

```
#include "stdio.h"
main()
{   char c; int i;
    float f;
    scanf("%c%d%f",&c,&i,&f);
    printf("%c,%d,%f\n",c,i,f);
}
```

上述程序执行到 scanf 函数时，需要用户从键盘输入一个字符、一个整型数和一个实型数后，才能够继续执行。运行时按以下方式输入 c、i、f 的值：

```
A  23  4.56 <CR>    （输入 c、i、f 的值）
A,23,4.56           （输出 c、i、f 的值）
```

3. 调用 scanf 函数时需要注意的问题

（1）在“格式控制”中的每个格式说明符，都必须在“地址表列”中有一个变量与之对应，如上例中%c 与&c 对应，%i 与&i 对应，%f 与&f 对应。而且，格式说明符必须与相应变量的类型一致。若类型不匹配，系统并不给出出错信息，但不可能得到正确的数据。

（2）当格式说明符之间没有任何字符时，在输入数据时，两个数据之间使用“空格”、“Tab”或“回车”键作间隔；如果格式说明符之间包含其他字符，则输入数据时，应输入与这些字符相同的字符作间隔。 如：

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

在输入数据时，应采用如下形式：

```
12,234<CR>
```

在输入字符型数据时，由于“空格”也作为有效字符输入，因此，不需要用“空格”作间隔，只要输入一个字符即可。例如：

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

当输入

```
a b c<CR>
```

时，字符'a'送给变量ch1，字符空格' '送给变量ch2，字符'b'送给变量ch3。

（3）在格式说明符前可以用一个整数指定输入数据所占列数，系统将自动按此列数来截取所需的数据。如：

```
scanf("%2d%3d",&a,&b);
```

若运行时输入12345，系统将自动地把12赋给变量a，把345赋给变量b。这种方式也可以用于字符型数据的输入，但不可以对实型数指定小数位的宽度。

（4）当输入的数据少于输入项时，程序等待输入，直到满足要求为止。当输入的数据多于输入项时，多余的数据并不消失，而是留作下一个输入操作时的输入数据。

（5）可以在格式字符和“%”之间加一个“*”号来跳过对应的输入数据。例如：

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

当输入以下数据时：

```
10  20  30  40<CR>
```

将把10赋给a，跳过20，把30赋给b，把40赋给c。

（6）应该强调指出，scanf 函数中的“地址列表”部分应当是变量的地址，而不是变量

名，如果只写变量名，编译阶段检查不出错误，但执行程序时，就会出现混乱，变量得不到相应的值。

（7）scanf 函数在调用结束后将返回一个函数值，其值等于得到输入值的输入项的个数，一般情况下 scanf 函数的返回值无用。

（8）在输入数据（常量）遇到以下情况时认为数据输入结束：

- 遇空格，或按“回车”或“跳格”（Tab）键。
- 遇宽度结束，如"%3d"，只取 3 列。
- 遇非法输入。如

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

若输入

```
1234a123o.56
```

第一个数据对应%d 格式输入 1234 之后遇字母'a'，因此认为数值 1234 后已没有数字了，第一个数据到此结束，把 1234 送给变量 a，字符'a'送给变量 b，由于%c 只要求输入一个字符，因此 a 后面不需要空格，后面的数值应送给变量 c。如果由于疏忽把本来应为 1230.56 错写成 123o.56，由于 123 后面出现字母'o'，就认为此数值结束，将 123 送给 c。

（9）在实际使用 scanf 函数时，不要在“格式控制”里加入与格式控制符不相关的其他符号，以免使实际的输入复杂，如下例。

例 3.10　求 a*b 和 a/b 的值。

```
main()
{   float a,b,x,y;
    scanf("a=%f,b=%f",&a,&b);
    x=a*b;   y=a/b;
    printf("\n\nx=%3.2f\ny=%3.2f\n",x,y);
}
```

程序运行情况如下：

```
a=2,b=3<CR>     （输入 a=2,b =3 后按回车键）

x=6.00          （输出结果）
y=0.67
.
```

3.5　顺序结构程序设计举例

顺序结构程序是按语句在程序中出现的顺序逐条执行。在 C 语言中，实现顺序结构的语句有表达式语句、空语句和复合语句。

例 3.11　以下程序由终端输入两个整数给变量 x 和 y，然后输出 x 和 y，交换 x 和 y 中的值后，再输出 x 和 y，验证两个变量中的数是否正确地进行了交换。

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

```
    t=x;x=y; y=t;
    printf("x=%d y=%d\n",x,y);
}
```

程序运行情况：

```
Enter x&y:              （由第 4 行的 printf 输出）
123  456<CR>            （从键盘输入两个整数，按 Enter 键）
x=123  y=456            （由第 6 行的 printf 输出）
x=456  y=123            （由第 8 行的 printf 输出）
```

在程序中交换 x 和 y 两个变量中的数，不可以简单地用“x=y;y=x;”两条语句来实现，语句“x=y;”执行的结果将把 y 中的值复制到 x 中，使 y 和 x 变量中具有相同的值，x 中原有的值丢失，因此无法再实现两数的交换。为了不丢失 x 中原有的值，必须在执行“x=y;”之前，把 x 中的值放到一个临时变量中保存起来（在此，通过 t=x;来实现），在执行了 x=y;之后，再把保存在临时变量中的值赋给 y（通过 y=t;来实现）。

例 3.12 输入三角形的三边长，求三角形面积（假定输入的三边长 a、b、c 能构成三角形）。

```
#include "stdio.h"
#include "math.h"
main()
{   float a,b,c,s,area;
    printf("Enter a,b&c:\n");
    scanf("%f,%f,%f",&a,&b,&c);
    s=l.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);
}
```

程序运行情况如下：

```
3,4,6↙                  （输入 3，4，6 后按回车键）
a=     3.00,b=     4.00,c=     6.00,s=     6.50
area=     5.33
```

例 3.13 从键盘输入一个小写字母，要求改用大写字母输出。

```
#include "stdio.h"
main()
{   char ch1,ch2;
    ch1=getcher();
    printf("%c,%d\n",chl,ch1);
    ch2=ch1-32;
    printf("%c,%d\n",ch2,ch2);
}
```

程序运行情况如下：

```
a<CR>               （输入 a 后按回车键）
a,97
A,65
```

习　题

一、简答题

3.1　C 语言中的语句有哪几类？

3.2　怎样区分表达式和表达式语句？C 语言为什么要设表达式语句？什么时候用表达式，什么时候用表达式语句？

二、选择题

3.3　若 a、b、c、d 都是 int 类型变量且初值为 0，以下选项中不正确的赋值语句是（　　）。

A．a=b=c=100;　　B．d++;

C．c+b;　　D．d=(c=22)-(b++);

3.4　以下选项中不是 C 语句的是（　　）。

A．{int j;　j++;　printf("%d\n",j);　}

B．;

C．a=5,c=10

D．{　;　}

3.5　以下合法的 C 语言赋值语句是（　　）。

A．a=b=58　　B．k=int(a+b)!

C．a=58,b=58　　D．--n;

3.6　以下程序的输出结果是（　　）。

A．0　　B．1　　C3　　D．不确定的值

```
main()
{   int x=10,y=3;
    printf("%d\n",y=x/y);
}
```

3.7　若变量已正确说明为 int 类型，要给 a、b、c 输入数据，以下正确的输入语句是（　　）。

A．read(a,b,c);　　B．scanf("%d%d%d",a,b,c);

C．scanf("%D%D%D",&a,%b,%c);　　D．scanf("%d%d%d",&a,&b,&c);

3.8　若变量已正确说明为 float 类型，要通过以下赋值语句给 a 赋值 10、b 赋值 22、c 赋值 33，以下不正确的输入形式是（　　）。

A．10　　B．10.0,22.0,33.0　　C．10.0　　D．10　22

22　　22.0　33.0　　33

33

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

3.9　若变量已正确定义，要将 a 和 b 中的数进行交换，下面不正确的语句组是（　　）。

A．a=a+b,b=a-b,a=a-b;　　B．t=a,a=b,b=t;

C．a=t;t=b;b=a;　　D．t=b;b=a;a=t;

3.10　若变量已正确定义，以下程序段的输出结果是（　　）。

A．输出格式说明与输出项不匹配，输出无定值

B．5.17

C．5.168

D．5.169

```
x=5.16894;
printf("%f",(int)(x*1000+0.5)/(float)1000);
```

3.11 若有以下程序段，c3 中的值是（ ）。

A．0 B．1/2 C．0.5 D．1

```
int  c1=1, c2=2,  c3;
c3=c1/c2;
```

3.12 若有以下程序段，其输出结果是（ ）。

A．3,0,-10 B．0,0,3

C．-10,3,-10 D．3,0,3

```
int  a=0,b=0,c=0;
c=(a-=a-5),(a=b,b+3);
printf("%d,%d,%d\n",a, b,c);
```

3.13 当运行以下程序时，在键盘上从第一列开始输入 987654321<CR>，则程序的输出结果是（ ）。

A．a=98,b=765,c=4321

B．a=10,b=432,c=8765

C．a=98,b=765.000000,c=321.000000

D．a=98,b=765.0,c=321.0

```
main()
{  int a;  float b,c;
   scanf("%2d%3f%4f",&a,&b,&c);
   printf("\na=%d,b=%f,c=%f",a,b,c);
}
```

3.14 以下程序的输出结果是（ ）。

A．a=%2, b=%5 B．a=2,b=5

C．a=%%d,b=%%d D．a=%d,b=%d

```
main()
{  int a=2,b=5;
   printf("a=%%d,b=%%d\n",a,b);
}
```

3.15 若 int 类型占两个字节，则以下程序段的输出结果是（ ）。

A．-1,-1 B．-1,32767

C．-1,32768 D．-1,65535

```
int  a=-1;
printf("%d,%u\n",a);
```

3.16 以下程序段的输出结果是（ ）。

A．*496 * B．* 496*

C．*000496* D．输出格式符不合法

```
int x=496;
printf("*%-06d*\n",x);
```

3.17　以下程序段的输出结果是（　　）。

A．|3.1415|　　B．|　　3.0|

C．|　　3|　　D．|　　3.|

```
float   a=3.1415;
printf("|%6.0f|\n",a);
```

3.18　以下程序段的输出结果是（　　）。

A．|2345.67800|　　B．|12345.6780|

C．|12345.67800|　　D．|12345.678|

```
printf("|%10.5f|\n",12345.678);
```

3.19　以下程序段的输出结果是（　　）。

A．*0000057.66*　　B．*　　57.66*

C．*0000057.67*　　D．*　　57.67*

```
float       a=57.666;
printf("*%010.2f*\n",a);
```

3.20　若从终端输入以下数据，要给变量 c 赋值 283.19，则正确的输入语句是（　　）。

A．scanf("%f",c)；　　B．scanf("%8.4f",&c)；

C．scanf("%6.2f",&c)；　　D．scanf("%8f",&c)；

```
283.1900↙
```

3.21　若变量已正确说明，要求用以下语句给 a 赋值 3.12、给 b 赋值 9.0，则正确的输入形式是（　　）。

A．3.12　9.0↙　　B．a=　3.12b=　9↙

C．a=3.12,b=9↙　　D．a=3.12　,b=9　　↙

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

3.22　以下程序的输出结果是（　　）。

A．9　　8　　B．8　　9

C．6　　6　　D．以上三个都不对

```
#include   "math.h"
main()
{   double   a=-3.0,b2;
    printf("%3.0f%3.0f\n",pow(b,fabs(a)),pow(fabs(a),b));
}
```

三、填空题

3.23　若有以下定义，请写出以下程序段中输出语句执行后的输出结果。

（1）_____　（2）_____　（3）

```
int         i=-200,j=2500;
prritf("(1)   %d   %d\n",i,j);
prritf("(2)   i=%d   j=%d\n",i,j);
prritf("(3)   i=%d\n   j=%d\n",i,j);
```

3.24　变量 i、j、k 已定义为 int 类型并有初值 0，用以下语句进行输入时

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

当执行第一个输入语句，从键盘输入：

```
12.3<CR>
```

则变量 i、j、k 的值分别是_____、_____、_____。

3.25 复合语句在语法上被认为是_____。空语句的形式是_____。

3.26 C 语句的最后用_____结束。

3.27 以下程序段，要求通过 scanf 语句给变量赋值，然后输出变量的值。运行时给 k 输入 100，给 a 输入 25.81，给 x 输入 1.89234，写出三种可能的输入形式_____、_____、_____。

```
int k;  float a;      double x;
scanf("%d%f%lf",&k,&a,&x);
printf("k=%d,a=%f,x=%f\n",k,a,x);
```

3.28 以下程序段的输出结果是_____。

```
int x=0177;
printf("x=%3d,x=%6d,x=%6o,x=%6x,x=%6u\n",x,x,x,x,x);
```

3.29 以下程序段的输出结果是_____。

```
int x=0177;
printf("x=%-3d,x=%-6d,x=$%-06d,x=$%06d,x=%%06d\n",x,x,x,x,x);
```

3.30 以下程序段的输出结果是_____。

```
double a=513.789215 ;
printf("a=%8.6f,a=%8.2f,a=%14.8f,a=%-14.8f\n",a,a,a,a);
```

四、读程序

3.31 分析下面的程序，给出模拟运行结果。

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

3.32 分析下面的程序，给出模拟运行结果。

```
#include  "stdio.h"
main()
{ char c1,c2;
   scanf("%c%c",&c1,&c2);
   ++c1;  c2--;
   printf("C1=%c,C2=%c\n",c1,c2);
}
```

3.33 写出下面程序的输出结果。

```
main()
{  int a=5,b=7,c=-1;
   float x=67.8564,y=-789.124;
   char c='A';
   long n=1234567;
   unsigned u=65535;
   printf("%d,%d\n",a,b);
   printf("%3d,%3d\n",a,b);
```

```
    printf("%f,%f\n",x,y);
    printf("%-10f,%-lOf\n",x,y);
    printf("%8.2f,%8.2f,%.4f,%.4f,%3f,%3f\n",x,y,x,y,x,y);
    printf("%e,%10.2e\n",x,y);
    printf("%c,%d,%o,%x\n",c,c,c,c);
    printf("%ld,%lo,%lx\n",n,n,n);
    printf("%u,%o,%x,%d\n",u,u,u,u);
    printf("%s,%5.3s\n","COMPUTER","COMPUTER");
}
```

3.34　用下面的 scanf 函数输入数据，使得 a=3，b=7，x=8.5，y=71.82，cl='A'，c2='a'，问在键盘上如何输入?

```
main()
{   int a,b;    float x,y;    char c1,c2;
    scanf("a=%d b=%d",&a,&b);
    scanf("x=%f y=%e",&x,&y);
    scanf("c1=%c c2=%c",&c1,&c2);
}
```

五、改错题

3.35　以下程序多处有错。要按下面指定的形式输入数据和输出数据，请对该程序做相应的修改。

```
main()
{   double     a, b,c,s,v;
    printf ("input a,b,c:\n");
    scanf ("%d%d%d",a,b,c);
    s= a*b;    v=a*b*c;
    printf("%d    %d    %d".,a,b,c);
    printf ("s=%f\n",s, "v=%f\n",v);
}
```

当程序执行时，屏幕的显示和要求输入形式如下：

```
input    a, b, c: 1.0      2.0      3.0                ←此处的 1.0    2.0    3.0 是用户输入的数据
a=2.000000,b=2,000000,c=3.000000                     ←此处是要求的输出形式
s=4.000000,    v=12.000000
```

六、编程题

3.36　编写程序，把 560 分钟换算成用小时和分钟表示，然后进行输出。

3.37　编写程序，输入两个整数 1500 和 350，求出它们的商数和余数并进行输出。

3.38　编写程序，读入三个双精度数，求它们的平均值并保留此平均值小数点后一位数，对小数点后第二位数进行四舍五入，最后输出结果。

3.39　编写程序，读入三个整数给 a、b、c，然后交换它们中的数，把 a 中原来的值给 b，把 b 中原来的值给 c，把 c 中原来的值给 a。

3.40　设圆半径 r=1.5，圆柱高 h=3，求圆周长、圆面积、圆球表面积、圆球体积、圆柱体积。用 scanf 输入数据，输出计算结果，输出时要求有文字说明，取小数点后两位数字。请编程序。

3.41　编程序，用 getchar 函数读入两个字符给 cl、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 cl、c2 应定义为字符型还是整型或二者皆可？②要求输出 cl 和 c2 值的 ASCII 码，应如何处理？用 putchar 函数还是 printf 函数？③整型变量与字符变量是否在任何情况下都可以互相代替？如：

```
char c1, c2;
```

与　int　cl, c2；

是否无条件地等价？

3.42　编程序，输入输出现在的时间、今天的日期及星期几。