第3章 顺序结构程序设计

学习时长：1周

学习目标

知识目标：

- 理解标准库函数。
- 理解数据的输入和输出。
- 理解C语言的基本语句类型。
- 理解C程序的一般结构。

能力目标：

- 掌握运用输入输出库函数进行数据的输入和输出。
- 掌握C语言各类基本语句。
- 掌握顺序结构的程序设计流程。

本章内容概要

3.1 数据的输入与输出

- 数据输出函数：putchar()、printf()
- 数据输入函数：getchar()、scanf()

3.2 顺序结构的流程

- C语言的基本语句
- C程序的一般结构

3.3 顺序结构综合应用实例

3.1 数据的输入与输出

计算机理论和实践证明，任何一个计算机程序均可由顺序结构、分支结构和循环结构3种基本结构组成。对于顺序结构的程序，计算机将按照语句的顺序逐条执行，是最基本的操作方式。

计算机程序在运行时，往往会涉及数据的输入和输出。C语言是通过调用系统提供的标准库函数来完成输入/输出操作。C语言的输入/输出标准函数以及相关的说明通常都包含在一个名称为stdio.h的头文件中。因此，调用这些标准函数之前，必须使用预编译命令#include，把包含有关函数信息的头文件包含到用户源文件中。在文件开头应有以下预编译命令：

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

3.1.1 数据输出函数

1. 字符输出函数：putchar()

函数原型：int putchar(int)

功能：putchar()是字符输出函数，其功能是在显示器上输出单个字符。

函数调用形式：putchar(ch)；

使用说明：

① 对应头文件"stdio.h"。

② 其中 putchar 是函数名，后面圆括号中的 ch 是函数参数，可以是字符型或整型的常量、变量或表达式。

例如：以下各语句均是在屏幕上输出大写字母 A。

```
char x = 'A';
putchar('A');
putchar(65);
putchar(x);
```

③ putchar(ch)通常不需要使用其返回值。

【例 3-1】 利用 putchar 函数输出字符。

```
#include <stdio.h>
main()
{
    char c1,c2;
    c1 = 'a';
    c2 = 'B';
    putchar(c1);
    putchar(c2);
    putchar('\n');              /* 输出换行 */
    c1 -= 'a' - 'A';            /* 相当于 c1 = c1 - 'a' + 'A'; */
    c2 -= 'A' - 'a';            /* 相当于 c2 = c2 - 'A' + 'a'; */
    putchar(c1);
    putchar(c2);
    putchar('\n');
}
运行结果:
aB
Ab
```

2. 格式输出函数：printf()

函数原型：int printf(char * format [,argument,…])

功能：按格式控制所指定的格式，在标准输出设备上输出列表项。

函数调用形式：printf("格式控制字符串",输出表列)；

(1) 使用说明：

① printf 是标准函数名，圆括号中函数参数包括格式控制字符和输出表列两部分。

第一部分是格式控制字符，是用双引号(英文)引起来的字符序列，用于指定输出格式。字符串可以包括“格式字符串”和“非格式字符串”两种字符。

- 格式字符串是以%开头的字符串，在%后面跟有格式字符，以说明输出表列对应变量的输出方式。
- 非格式字符串也就是普通的字符，输出时原样输出。

第二部分是输出表列，即要输出的变量，表达式等的列表，多个输出项之间必须用逗号分隔。

例如：

```
int i = 5;
double x = 1.5;
printf("i = %d  x = %f", i, x) ;              /* 输出结果为:i = 5  x = 1.500000 */
```

注意：格式控制符"i=%d x=%f"中的输出次序。

第一步：“i=”是普通字符原样输出。

第二步：遇到%d，到后面的输出表列取对应的第 1 个表达式 i 的值 5，即 5 取代%d 的位置。

第三步：“ x=”是普通字符原样输出，请注意 x 前面有两个英文空格，也需原样输出。

第四步：遇到%f，到后面的输出表列取对应的第 2 个表达式 x 的值 1.5，输出为 1.500000。

② 注意以下几点：

- 使用 printf 函数输出数据时，格式控制中的格式说明符与输出参数的个数和类型必须一一对应。有几个输出参数，就应该写几个格式说明符，并且类型、排列顺序不得有误，否则会产生错误的输出结果。
- 如果格式控制说明项数多于输出表列个数，则会输出错误数据。
- 如果输出表列个数多于格式控制说明数，则多出数不被输出。

例如：

```
int i = 23;
float y = 123.456, x = 22.222;
printf(" % - 4d, % g", i, y, x);
```

输出结果：

```
23  ,123.456
```

(2) 格式控制字符格式：

printf()中格式控制字符串的一般形式为：

```
%[flags][width][.prec][F|N|h|L]type
```

对应的含义：

```
%[标志][输出最小宽度][.精度][长度] 类型字符
```

其中方括号[ ]中的项为可选项。各项的意义介绍如下。

① type 用于表示输出数据的类型,其格式符和意义如表 3-1 所示。

表 3-1　格式符及其意义

输 出 类 型	表示输出类型的格式符	格式符意义
整型数据	d,i	以十进制形式输出带符号整数(正数不输出符号)
	o	以八进制形式输出无符号整数(不输出前缀 0)
	x,X	以十六进制形式输出无符号整数(不输出前缀 0x 或 0X)
	u	以十进制形式输出无符号整数
实型数据	f	以小数形式输出单、双精度实数
	e	以指数形式输出单、双精度实数
	g	以%f%e 中较短的输出宽度输出单、双精度实数
字符型数据	c	输出单个字符
	s	输出字符串

② flags 标志字符常用的有 3 种。

－:为左对齐,否则右对齐。

＋:正数输出＋,负数输出－。

空格:正数输出空格,负数输出－。

③ width 用于指定数据的最小输出宽度(称为域宽)。对于实型数据,m 指定的域宽包括整数位、小数点、小数位和符号位所占的总位数。如果输出数据位数小于域宽,不足部分用空格补齐;如果超出域宽,则按实际宽度输出(此时也可以省略)。

④ prec 为精度指示符。用小数点加十进制正整数构成,用来限制输出数的精度。

⑤ [F|N|h|L]为长度修饰符,

F:远程指针地址。

N:近程指针地址。

h:短整数据的值。

L:长整(双精度)数据的值。

例如:

设 int a=12; float b=1234.5678; 分析不同格式的输出结果:

```
a 的格式说明:          a 的输出方式:
      %d               12
      %8d              ____12(_代表空格,前空 6 个空格)
      %o               14     (八进制)
      %x               c      (十六进制)
      %u               12     (无符号)
      % -8d            12 ____(后空 6 个空格,左靠齐)
b 的格式说明:          b 的输出方式:
      %f               1234.567749(小数位为 6 位,浮点数保存时有误差)
      %e               1.234568e+003 (指数形式)
      %8.2f            1234.57 (小数两位,长 8 位)
      %10.2e            1.23e+003(前有 1 空格)
      % -10.2e         1.23e+003 (后有 1 空格)
```

(3) 注意以下几点:

① %d 按整型数据的实际长度输出,%md,m 为指定的输出字段的宽度。如果数据的位数小于 m,则左端补以空格,若大于 m,则按实际位数输出。

② 八进制和十六进制形式输出整型，由于将内存单元中各位的值按八进制形式输出，因此输出的数据不带符号，即将符号也一起作为八进制数的一部分输出。

例如：

```
int x = 123;
printf("x = %d,x = %o",x,x);
```

输出结果：

```
x = 123,x = 173
```

③ %f 输出浮点数时不指定字段宽度，由系统自动指定，使整数部分全部如数输出，并输出 6 位小数。注意并非全部数字都是有效数字。单精度实数一般的有效数字为 7 位。

④ %e 输出指数时不指定输出数据所占的宽度和数字部分的小数位数。由系统自动指定给出 6 位小数，指数部分占 5 位(如 e+002)，其中“e”占一位，指数符号占一位，指数占 3 位。数值按标准化指数形式输出(即小数前必须有而且只有一位非零数字)。

例如：

```
printf("%e",123.456);
```

输出：1.234560e+002。%e 格式输出的实数共占 13 列宽度。

⑤ g 格式符用来输出实数，系统根据数值的大小，自动选 f 格式或 e 格式(选择输出时占宽度较小的一种)，且不输出无意义的 0。

例如：

```
若 f = 123.456,则
printf("%f %e %g",f,f,f);
```

输出的结果如下：

```
123.456001 1.234560e+002 123.456
```

⑥ %s 格式符用来输出字符串。%ms 输出的字符串占 m 列，如字符串本身长度大于 m，则突破 m 的限制，将字符串全部输出，如串小于 m，则左补空格。%-ms 如果串长小于 m，则在 m 列范围内，字符串向左靠，右补空格。%m.ns 输出占 m 列，但只取字符串中左端 n 个字符，这 n 个字符输出在 m 列的右侧，左补空格。%-m.n 其中 m，n 含义同上，n 个字符输出在 m 列范围的左侧，右补空格。如果 n>m，则 m 自动取 n 值，即保证 n 个字符的正常输出。

例如：

```
printf("%3s,%7.2s,%.4s%,%-5.3s,","china", "china", "china", "china");
```

输出结果：

```
china,     ch,chin,chi  ,
```

⑦ 格式说明符的%和后面的描述符之间不能有空格。除%X、%E、%G 外类型描述符必须是小写字母。

⑧ 长整型数无论是按十进制、八进制还是十六进制形式输出，都一定要使用小写字母“l”进行修饰，否则会出现输出错误。

⑨ printf 函数的参数可以是常量、变量或表达式。

【例 3-2】 分析以下程序运行结果。

```
#include <stdio.h>
main()
{
    int i=8;
    printf("%d\t%d\n", i, ++i);
}
```

因为 printf 函数对输出表中各量求值的顺序是自右至左进行的，因此先对最后一项“++i”求值，然后 i 自增 1 后为 9，取值结果为 9。

再对“i”项求值，此时 i 为 9。

但是必须注意，求值顺序虽是自右至左，但是输出顺序还是从左至右，因此程序的输出结果是：

```
9  9
```

请注意，输出表中如果有多个表达式时，不同 C 语言编译器的运行结果可能不同。

3.1.2 数据输入函数

1. 键盘输入函数：getchar()

函数的原型为：int getchar(void) /* stdio.h */

函数功能：getchar 函数的功能是从键盘上接收一个字符。该函数通过返回值带回接收字符的 ASCII 码。void 表示没有参数。

(1) getchar 函数调用的基本形式有以下几种。

① 作为独立的函数调用语句使用，例如：

```
getchar();
```

其作用是等待用户从键盘输入一个字符，之后继续执行程序。这种情况相当于程序暂停，之后按任意键继续执行程序。

② 作为表达式在赋值语句中使用，例如：

```
c=getchar();
```

其作用是等待用户从键盘输入一个字符，将其赋给 char 型(或 int 型)变量 c，之后继续执行程序。

③ 作为表达式出现在其他语句中，例如：

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

其作用是等待用户输入一个字符，之后输出该字符。

(2) getchar 函数使用说明。

① getchar 是一个无参数的函数调用，函数名后面的圆括号不能省略。

② getchar 函数需要交互输入，接收到输入的字符之后才继续执行程序。getchar 函数

只能接收单个字符，输入数字也按字符处理。输入多于一个字符时，只接收第一个字符。特别注意的是，空格、回车和 Tab 键都将作为字符接收。

③ getchar 和 putchar 函数一样，都是标准函数，包含在 C 语言标准 I/O 函数库中，使用时必须在程序开始使用 #include <stdio.h>预处理命令，否则编译出错。

【例 3-3】 从键盘输入任一数字字符'0'-'9'，将其转换为对应的数字。

```
#include <stdio.h>
main()
{
    char ch;
    ch = getchar();
    printf("字符 %c,数字 %d\n",ch,ch - '0');
}
运行结果:
输入:6↙
输出:字符 6,数字 6
```

2. 格式输入函数：scanf()

函数原型：int scanf(char * format [,argument,…])　　/* stdio.h */

函数功能：按格式控制指定的格式，从标准输入设备(即键盘)交互输入数据，并依次存放到对应的地址参数指定的变量中(即将输入的值赋给变量)。函数可以实现整型、实型和字符型等数据的赋值。

函数的一般形式为：

```
scanf("格式控制字符串",地址表列);
```

(1) 使用说明。

① scanf()函数的参数包括格式控制字符串和输入地址表列两部分。

- 格式控制字符包含格式说明符和普通字符，用于指定输入数据的类型和输入形式。
- 地址表列是由若干地址组成的表列。可以是变量的地址或字符串的首地址。

② “&”是取地址运算符。表示取一个变量的地址。

例如：

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

&a,&b,&c 分别表示取变量 a、b、c 的地址。

③ scanf("%d%d%d",&a,&b,&c)；语句执行时，表示从键盘接收三个整型数据，分别放到变量 a、b、c 的地址中。即通过键盘给变量 a、b、c 进行了赋值。使用时必须避免重复赋值。

例如：

```
a = b = c = 5;                    /* 给 a、b、c 赋值 */
scanf("%d%d%d",&a,&b,&c);   /* 从键盘再次给 a、b、c 赋值 */
```

请注意区分用 scanf 函数赋值和在程序中直接赋值的意义。

④ 用 scanf()函数接收数据时请注意格式控制字符包含的格式说明符和普通字符都是

用于输入的，不会显示在屏幕上。执行 scanf 语句时屏幕会有一个闪烁的光标等待用户输入数据，这时：

• 如果格式控制字符包含普通字符，输入数据时必须将普通字符原样给出。

例如：

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

给 a、b 赋值为 3，4 时的格式必须为："a＝3，b＝4 <回车>"。再次强调，格式控制字符中的","是普通字符。

• 如果格式控制字符不包含普通字符，输入多个数据时 C 语言系统提供了三种默认的分隔符：空格、Tab、Enter(<回车>)。

例如：

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

如果 a、b、c 的值分别为 7，8，9，这时必须输入 7 8 9 <回车>，或者把空格用 Tab、Enter 取代。

(2) 格式控制字符。

格式控制字符串构成的内容与 printf 函数类似，由％和格式字符组成。中间可以插入附加的字符。

```
%[*][width][F|N][h|L]type
```

其中有方括号的项为任选项。各项的意义如下：

① type 表示输入数据的类型，其格式字符的意义如表 3-2 所示。

表 3-2　scanf 函数中使用的格式说明符

输入类型	格式说明符	说　明
整型数据	d(％ld)	输入十进制整型数(长整型数)
	u(％lu)	输入无符号的十进制整型数(无符号长整型数)
	o(％lo)	输入八进制整型数(八进制长整型数)
	x(％lx)	输入十六进制整数(十六进制长整型数)
实型数据	f(％lf)	输入小数形式的单精度实型数(双精度实型数)
	e(％le)	输入指数形式的单精度实型数(双精度实型数)
字符型数据	％c	输入单个字符
	％s	输入字符串，将字符串送到一个字符数组

② [*]输入赋值抑制字符，输入数据但不赋值。

③ [width] 用十进制整数指定输入的宽度，系统自动按它截取所需数据。

④ [F|N][h|L]与 printf()函数相同。

【例 3-4】 scanf 函数的使用示例。

```
#include <stdio.h>
main()
{
    int i;
    float x;
    char name[50];
```

```
    scanf("i = %2d, x = %f, %*dname = %2s", &i, &x, name);
}
运行结果:
输入:56789 0123 45a72↙
输出:i = 56, x = 789.000000, name = 45
```

(3) 注意以下几点:

① %f 和%e 是输入 float 型数据的格式说明符,如果输入 double 型数据(无论是以小数形式还是以指数形式输入),都必须用小写字母"l"进行附加说明,否则不能正确输入。

② %d、%u、%o 和%x 分别可以输入十进制、八进制、和十六进制基本整型数。如果要输入 long 型数据,还应在相应格式说明符中加入小写字母"l"进行修饰说明。

③ scanf()的格式控制字符串中的字符不是用于输出的,都是要求输入的。如语句 scanf("x=%d",&x);输入时必须加"x="。

④ 参数的第二部分一定是地址列表,不能是表达式,请记得在变量前加上 &。

⑤ 执行 scanf()输入数据时,在两个数据之间允许以一个或多个空格间隔,也可以用回车键、Tab 键分隔。

⑥ 输入实数时不允许规定精度,如%10.4f 是不合法的。

⑦ %后面有 * 号时,该数据会被禁止使用。

⑧ 如果输入时类型不匹配则停止处理,函数返回 0。

3.2 顺序结构的流程

在计算机系统中,按语句编写的自然顺序逐条执行即为顺序结构。顺序结构是由若干语句串接而成的,程序的流程从上至下依次执行每一条语句。顺序结构的程序通常不能完成复杂的工作,其流程一般是先输入一些数据,然后对数据进行某种处理或运算,最后输出结果,如图 3-1 所示。

3.2.1 C 语言的基本语句

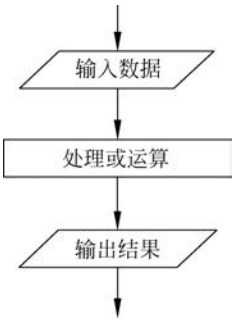

图 3-1 顺序流程图

C 程序是由若干条语句组成的,通常一条语句命令计算机完成一个基本动作。C 语句主要由说明语句、表达式语句、函数调用语句、复合语句、控制语句和空语句组成。

在汉语和英语文章中,分别使用"。"和"."表示一句话的结束。与此类似,在 C 语言中,使用分号";"表示一个语句的结束,即两个";"之间就是一条语句内容,在书写上可以一条语句占一行,也可以一行并列写多条语句。

1. 说明语句

变量的说明和函数的说明后面加分号结束,统称说明语句。

例如:

```
int a,b,c;                /*变量说明语句*/
int x[8], *xy;            /*数组变量和指针说明语句*/
```

```
int num(int a,int b);          /* 函数声明语句 */
```

2. 表达式语句

表达式语句由表达式末尾加上分号“;”组成。任何一个表达式都可以加上分号而成为表达式语句。

一般形式为：

```
表达式;
```

执行表达式语句就是计算表达式的值。

例如：

```
y = a + b * x;                 /* 表达式赋值语句 */
y *  = 325 + a;                /* 表达式复合赋值语句 */
a = 4,b = 7,c = 2;             /* 逗号表达式语句 */
i++;                           /* 算术表达式语句 */
```

3. 函数调用语句

一般形式：

```
函数名(实际参数表);
```

函数在 C 语言中就是指一段具有特定功能的程序。这段程序通常需要完成规定的功能，在任务执行完毕以后可以返回执行结果，即函数的返回值。

在 C 语言中，函数分两种，标准库函数和用户自定义函数。

- 标准库函数是 C 语言自带的，需要时可以直接调用。
- 用户自定义函数是 C 语言原来没有，用户根据自己的需要编写的。

不管是标准库函数还是用户自定义函数，使用时都需要调用函数。

函数调用语句有两种形式：

① 如果需要函数的返回值。这时通常需要用一个变量记录下函数的返回值。

例如：

```
a = getchar(); /* 从键盘接收一个字符赋给变量 a */
```

② 不需要函数的返回值。这时可以直接调用。

例如：

```
printf("C Program"); /* 调用输出函数 printf,输出字符串"C Program" */
```

因此，在使用函数时必须首先明确是需要用到函数的返回值，还是只需要函数的特定功能，来决定函数的调用方法。

4. 复合语句

把若干连续的语句用花括号括起来组成的一个语句称复合语句。在程序中应把复合语句看成单条语句，而不是多条语句。

一般形式为：

```
{
说明部分;                  /* 可以没有 */
可执行语句;
}
```

例如：

```
#include < stdio.h >
main()
{
    int a = 5;
    {
        int b = 3;
        b++;
    }
    a++;
}
```

注意：

① 复合语句是一条语句，如上面程序中 main 函数中只有 3 条语句。

② 复合语句内的各条语句都必须以分号(;)结尾，注意：在括号(})外不能加分号。

③ 复合语句内又可以定义变量，该变量只在复合语句中有效。

5. 控制语句

控制语句用于控制和改变程序的流向，以实现程序的各种结构方式。

它们由特定的语句定义符组成。C 语言有 9 种控制语句，可分成以下 3 类：

① 条件判断语句：

```
if 语句,switch 语句
```

② 循环执行语句：

```
do while 语句,while 语句,for 语句
```

③ 转向语句：

```
break 语句,goto 语句,continue 语句,return 语句
```

控制语句将用专章分别讲解。

6. 空语句

只有分号(;)组成的语句称为空语句。空语句是什么也不执行的语句，在程序中可用来作空循环体；另外，为了程序的结构清楚，可读性好，以及扩充新功能方便也常用空语句来表示。

例如：用于延时的空循环，空语句作循环体。

```
for(i = 1;i < = 500;i++);
```

3.2.2 C 程序的一般结构

C 语言程序一般由三部分组成：预处理命令部分、主函数部分及自定义函数部分。以下是简化的程序格式，不包含自定义函数：

```
#include <头文件>
main()
{
    说明部分;
    语句序列;
}
```

include 是编译预处理命令，由编译程序处理，对源程序进行翻译前的一些操作。使用时以＃号开头，它的完整用法参见 7.4 节；自定义函数的有关内容见第 7 章。

3.3 顺序结构综合应用实例

【例 3-5】 从键盘输入 2 个变量的值，交换后输出。

分析：

S1：定义 2 个变量 a，b。

S2：从键盘给 2 个变量赋值。

S3：交换其值。

S4：输出变量 a，b。

```
#include <stdio.h>
void main()
{
    int a,b, c;
    printf("\nInput a,b:");
    scanf("%d,%d",&a,&b);
    c = a;
    a = b;
    b = c;
    printf("After exchange: a = %d, b = %d\n",a,b);
}
运行结果:
输入:Input a,b:32,57↙
输出:After exchange: a = 57, b = 32
```

交换 a，b 变量的值有多种方法。

方法一：如上面的程序，设置中间变量，进行交换。c＝a；a＝b；b＝c；这时注意赋值的先后次序不能更改。

方法二：不借助第三个变量实现交换。a＝a＋b；b＝a－b；a＝a－b；请仔细体会这种思路。

【例 3-6】 输入长和宽，计算长方形的面积。

分析：

S1：定义变量 length 和 wide、area，用于存放长方形的长、宽和面积。

S2：输入长方形的长 length 和宽 wide。

S3：计算“area＝length * wide”。

S4：输出其面积 area。

```
#include <stdio.h>
main()
{
    float length,wide,area;
    printf("input length and wide :");
```

```
        scanf("%f,%f",&length,&wide);
        area=length*wide;
        printf("area=%6.2f\n",area);
    }
```

运行结果:

输入:3.2,4.0↙

输出:area= 12.80

【例 3-7】 输入三角形的三条边,求三角形的面积。

分析:

S1:定义三个变量代表三角形的三边。

S2:给变量赋值。注意输入的三边必须是任意两边之和大于第三边,满足构成三角形的条件。

S3:计算三角形面积的计算公式如下:

area=$\sqrt{s(s-a)(s-b)(s-c)}$ 其中 s=(a+b+c)/2

S4:输出面积。

```
#include<stdio.h>
#include<math.h>
main()
{
    float a,b,c,s,area;
    printf("input three numbers:");
    scanf("%f,%f,%f",&a,&b,&c);
    s=0.5*(a+b+c);
    area=sqrt(s*(s-a)*(s-b)*(s-c)); /* sqrt 函数是平方根函数,需要用到 math.h */
    printf("a=%.2f,b=%.2f,c=%.2f\n",a,b,c);
    printf("area=%.2f\n",area);
}
```

运行结果:

输入:3,4,5↙

输出:a=3.00, b=4.00,c=5.00

area=6.00

【例 3-8】 从键盘输入一个小写字母,输出其相应的大写字母。

分析:

S1:定义变量 ch。

S2:给变量 ch 赋值。

S3:转换大小写。

S4:输出 ch。

```
#include<stdio.h>
main()
{
    char ch;
    printf("input a character:");
    ch=getchar();
    ch=ch-'a'+'A';
```

```
    putchar(ch);
    printf("\n");
}
运行结果:
输入:input a character:a↙
输出:A
```

大小写转换算法有多种,一种如程序中,另一种是大小写字符中间相差 32,即大写字母+32=小写字母。

【例 3-9】 编写一个程序,输入一个 3 位正整数,要求逆序输出对应的数,如输入 123,则输出 321。

分析:

S1:从键盘接收一个 3 位整数。

S2:分别取其个、十、百位分别放到 3 个变量中。

S3:重新组合成新数。

S4:输出该数。

```
#include <stdio.h>
main()
{
    int n,i,j,k,m;
    printf("输入一个 3 位正整数:");
    scanf("%3d",&n);                    /*n 为原 3 位数*/
    i=n/100;                            /*百位数*/
    j=n/10%10;                          /*十位数*/
    k=n%10;                             /*个位数*/
    m=100*k+10*j+i;                     /*重新组合*/
    printf("%d=  =>%d\n\n",n,m);
}
```

【例 3-10】 输入一个 3 位整数,依次将其符号位、个位、十位、百位转换为数字字符输出。

分析:

S1:定义 4 个字符型变量 c1、c2、c3、c4,定义 1 个整型变量 x。

S2:用键盘输入一个 3 位数的整数,赋值给变量 x。

S3:用条件运算符对 x 的符号位进行取值,赋值给 c4。

S4:对 x 取绝对值。

S5:用表达式 x%10+'0',分别取出个位,十位和百位的数值,并转换为字符型数据。

S6:依次输出各位的字符型数据。

```
#include <stdio.h>
#include <math.h>
main()
{
    char c1,c2,c3,c4;
    int x;
    printf("please input a number:");
```

```
    scanf("%d",&x);                                  /* 输入一个 3 位整数 x */
    c4 = (x>=0 ? '+' : '-');                         /* c4 为 x 的符号位 */
    x = abs(x);                                      /* abs()取整数的绝对值,头文件 math.h */
    c1 = x%10 + '0';                                 /* c3 取个位,并转换为字符 */
    c2 = x/10%10 + '0';                              /* c2 取十位,并转换为字符 */
    c3 = x/100%10 + '0';                             /* c1 取百位,并转换为字符 */
    printf("%c\n%c\n%c\n%c",c4,c3,c2,c1);   /* 输出 */
}
```

习题与思考

一、选择题

1. 能正确表示 a 和 b 同时为正或同时为负的逻辑表达式是_____。

A. (a>=0 || b>=0) && (a<0 || b<0)

B. (a>=0 && b>=0) && (a<0 && b<0)

C. (a+b>0 && a+b<=0)

D. a*b>0

2. 以下条件表达式中能完全等价于条件表达式 x 的是_____。

A. (x==0)　　B. (x!=0)　　C. (x==1)　　D. (x!=1)

3. 设 ch 是 char 型变量,值为'A',则表达式"ch=(ch>='A' && ch<='Z')? ch+32: ch"的值是_____。

A. Z　　B. a　　C. z　　D. A

4. 设 a, b 和 c 都是 int 型变量,且 a=3,b=4,c=5,则下面的表达式中,值为 0 的表达式是_____。

A. 'a'&&'b'　　B. a<=b

C. a||b+c&&b-c　　D. !((a<b)&&!c||1)

5. 设 a=5,b=6,c=5,d=8,m=2,n=2,执行 (m=a>b)&&(n=c>d) 后 n 的值为_____。

A. 1　　B. 2　　C. 3　　D. 0

6. 设 x、y、z、t 均为 int 型变量,则执行以下语句后,t 的值为_____。

```
x = y = z = 1;
t = ++x || ++y && ++z;
```

A. 不定值　　B. 4　　C. 1　　D. 0

7. 已知 int x=10,y=20,z=30,则执行

```
if (x > y)
z = x;x = y;y = z;
```

语句后,x、y、z 的值是_____。

A. x=10,y=20,z=30　　B. x=20,y=30,z=30

C. x=20,y=30,z=10　　D. x=20,y=30,z=20

8. 设 a 为整型变量,不能正确表达数学关系 10<a<15 的 C 语言表达式是_____。

A. 10<a<15　　B. a==11||a==12||a==13||a==14

C. a>10&&a<15　　D. !(a<=10)&&!(a>=15)

9. 语句："printf("%d",(a=2) && (b=-2));"的输出结果是_____。

A. 无输出　　B. 结果不确定　　C. -1　　D. 1

10. 执行下列程序片段时的输出结果是_____。

```
int x = 13,y = 5;
printf(" % d",x % = (y/ = 2));
```

A. 3　　B. 2　　C. 1　　D. 0

11. 下列程序的输出结果是_____。

```
main ( )
{
    int x = 023;
    printf(" % d", -- x);
}
```

A. 17　　B. 18　　C. 23　　D. 24

12. 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 AB↙　　D. 10 20AB↙

13. 执行下列程序片段时的输出结果是_____。

```
int x = 5,y;
y = 2 + (x += x++,x + 8,++x);
printf(" % d",y);
```

A. 13　　B. 14　　C. 15　　D. 16

14. 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，则正确的数据输入方式是_____。

A. 132↙　　B. 1,3,2↙

C. a=1 b=3 c=2↙　　D. a=1,b=3,c=2↙

二、填空题

1. 在一个 C 程序中依次执行完所有语句是 C 语言的_____结构程序设计思想。

2. C 语言中语句可以分为_____、_____、_____、_____和_____等 5 种类型。

3. C 控制语句有_____种。

4. 一个表达式要构成一个 C 语句，必须在表达式尾端加_____符号。

5. 复合语句是用一对_____界定的语句块。

6. printf 函数和 scanf 函数的格式说明都是使用_____字符开始。

7. scanf 处理输入数据时，遇到下列三种情况时该数据认为结束：(1)_____，

(2)_____,(3)_____。

8. 设“int i,j; float x;”为将－10 赋给 i,12 赋给 j,410.34 赋给 x,对应 scanf 函数调用语句为

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

则数据输入形式是_____。

9. C 语言本身不提供输入输出语句,其输入输出操作是由系统提供的_____来实现的。

10. 一般地,调用标准字符或格式输入输出库函数时,文件开头应有以下预编译命令:_____。

三、读程序写结果

1.
```
char c1 = 'a',c2 = 'c';
printf(" %d, %c",c2 - c1,c2 - 'a' + 'C');
```

2.
```
#include <stdio.h>
main()
{
    char ch1,ch2;
    ch1 = 'A' + '5' - '3' ;
    ch2 = 'A' + '6' - '3';
    printf(" %d, %c",ch1,ch2);
}
```

3.
```
#include <stdio.h>
main()
{
    int x;float y;
    scanf(" %3d %f",&x,&y);
    printf("x = %d,y = %f",x,y);
}
```
当执行时输入数据:1234 <空格> 678 <回车>

4.
```
#include <stdio.h>
main ( )
{
    int a = 12345;
    float b = - 198.345, c = 6.5;
    printf("a = %4d,b = % - 10.2e,c = %6.2f\n",a,b,c);
}
```

5.
```
#include <stdio.h>
main ( )
{
    int x = - 2345;
    float y = - 12.3;
    printf(" %6d,\t %6.2f\n",x,y);
}
```

四、编程题

1. 编写一个程序,从键盘接收 3 个整数,作为三角形的三边,求出三角形的面积。定义变量 a, b, c 为三条边长,则面积公式为: p=(a+b+c)/2; 面积=sqrt(p(p−a)(p−b)(p−c))。

2. 编写一个程序，输入华氏温度，输出相应的摄氏温度。转换公式是：

c=(f−32)/1.8，c 表示摄氏温度，f 表示华氏温度

3. 输入秒数，将它转换，用小时、分钟、秒来表示。例如，输入 7278 秒，则输出 2 小时 1 分 18 秒。

4. 输入人民币数值，分别输出对换后的美元(USD，汇率 0.1574)、英镑(GBP，汇率 0.102)、日元(JPY，汇率 18.97)、韩元(KRW，汇率 186.56)和港元(HKD，汇率 1.22)。