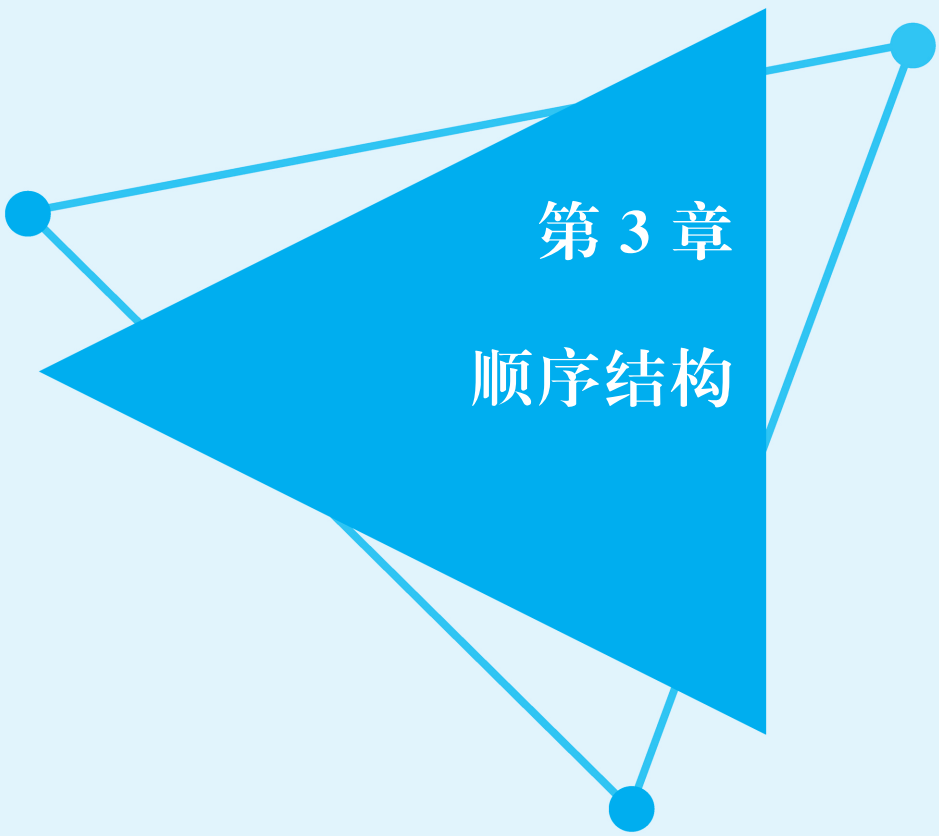

第 3 章

顺序结构

本章首先介绍 C 语言的语句，之后详细讨论 C 语言中输入输出函数的使用方法和规则，通过程序案例使读者理解什么是顺序结构程序。

本章学习目标

（1）了解输入输出函数的使用方法。

（2）掌握输入输出格式控制方法。

（3）掌握应用顺序结构程序解决问题的方法。

3.1 顺序结构简介

第 3 章案例代码

1. 语句

C 语言程序是以函数为基本单位的，函数是由一个一个的 C 语句构成。C 语句必须以分号结束。

C 语言的语句主要分为以下 6 类。

（1）说明语句。说明语句一般用来定义变量数据类型等。例如：

```
int a=5,b;
float f1,f2;
```

（2）表达式语句。表达式语句是指由一个 C 表达式加上分号构成的语句。例如：

```
a=b+1;                          /*赋值表达式 a=b+1 加上分号*/
i=1,j=2;                        /*逗号表达式 i=1,j=2 加上分号*/
i++;                            /*表达式 i++加上分号*/
```

（3）函数调用语句。函数调用语句是由一个函数调用加上分号。例如：

```
printf("\n");
srand(time(NULL));
```

这类语句也可以归属于表达式语句，因为函数调用本身也是一个表达式。

（4）空语句。空语句是仅由一个分号构成的语句，没有任何动作。例如：

```
;
```

（5）复合语句。复合语句是指将一组语句用大括号（{}）括起来，从而使整个大括号变成一个整体（复合语句）。整体上看，复合语句是一个语句。例如：

```
{
  a=5;
  b=6;
  c=7;
}
```

复合语句在 C 语言程序中的用处很大，在以后的学习中大家会逐渐体会到。有的书中也将复合语句称为分程序或语句块。

（6）控制语句。控制语句完成一定的控制功能，实现程序流程的跳转。C 语言提供的控制语句有：

```
goto                无条件转向语句
if( ) … else …      选择语句
switch( ){  }       多分支选择语句
while( )            循环语句
for( )              循环语句
do{  }while()       循环语句
break               循环控制语句
continue            循环控制语句
return              从函数返回语句
```

2. 顺序结构案例

顺序结构是最简单的程序结构，也是最常见的程序结构。顺序结构程序的执行顺序是自上而下，依次执行。之前章节中编写的所有程序都是顺序结构。

案例 03-01-01 考考你

案例代码 03-01-01.c

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main(){
    int a,b,c,d,r;
    srand((unsigned)time(NULL));
    printf("*********两位数四则运算测试*********\n");
    printf("请输入以下算式的结果:\n");
    a=rand()%89+11;
    b=rand()%89+11;
    c=rand()%4+1;
    (c==1)?(printf("%d+%d=",a,b),d=a+b):1;
    (c==2)?(printf("%d-%d=",a,b),d=a-b):1;
    (c==3)?(printf("%d*%d=",a,b),d=a*b):1;
    (c==4)?(printf("%d/%d=",a,b),d=a/b):1;
    scanf("%d",&r);
    printf(d==r?"答对了,你真棒!":"真对不起,你答错了!");
    return 0;
}
```

程序分析：

此例程序是顺序结构的程序，程序从第一条语句开始，依次向下一条一条语句执行，直到最后一个语句。请运行并分析此程序的功能。

案例拓展 考考计算机

请仿照以上案例编写顺序结构的程序，请设计由你输入一个加法或减法算式，然后由程序输出这个算式的结果。例如，你输入 1+2 时，程序输出 3；你输入 10−6 时，程序输出 4。

3.2 标准输入输出函数

C 语言数据的输入输出都是通过函数调用实现的。标准函数库 stdio.h 中包括标准输入输出函数 scanf()和 printf(),以及字符输入输出函数 getchar()和 putchar()等,使用这些函数的 C 程序中,应该在程序的开头处加上下面的编译预处理命令:

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

1. 标准格式输出函数 printf()

标准格式输出函数 printf()一般用于向标准输出设备(显示器)按规定的格式输出信息。调用 printf()函数的一般形式为:

```
printf(格式控制字符串,输出值参数列表);
```

功能说明:

(1) 格式控制字符串是一串用双引号括起来的字符,其中包括普通字符和格式转化说明符。格式转化说明符以%开头,后跟一个或几个规定字符,简称格式说明符。一个格式说明符用来代表一个输出的数据,并规定了该数据的输出格式。

例如,在语句:printf("a=%d,b=%d",a,b);中,字符串"a=%d,b=%d"是格式控制字符串。a,b 是输出值参数列表。在格式控制字符串"a=%d,b=%d"中,a=和 b=是普通字符,%d 是格式说明符。

(2) 格式控制字符串中的普通字符要按原样输出。

(3) 一个格式说明符用来代表一个输出的数据,所代表的数据在输出值参数列表中。输出值参数列表是一系列用逗号分开的表达式,表达式的个数和顺序与前面的格式说明符要一一对应。

例如,有 int a=3,b=8;,则执行以下语句:printf("a=%d,b=%d",a,b);输出结果是:a=3,b=8。

2. 格式说明符

不同类型的数据在输出时应该使用不同的格式说明符。C 语言提供的格式说明符及其含义见表 3-1。

表 3-1 C 语言提供的格式说明符及其含义

格式说明符	所代表的数据类型;输出形式
%d	int 型;十进制有符号整数,正数符号省略
%ld	long 型;十进制有符号长整数,正数符号省略

续表

格式说明符	所代表的数据类型;输出形式
%u	int 型;十进制无符号整数
%o	int 型;无符号八进制整数,不输出前导 0
%x,%X	int 型;无符号十六进制整数,不输出前导 0x
%#o、%#x、%#X	八进制和十六进制整型数据的输出须加上前导 0 或 0x 或 0X
%c	int 型(char 型);一个字符
%f	float 型;十进制小数,默认小数位数为 6 位
%lf	double 型;十进制小数,默认小数位数为 6 位
%e	double 型;指数形式,输出浮点数
%g	double 型;自动在%f 和%e 之间选择输出宽度小的表示法,且不输出小数末尾的 0
%s	字符串;顺序输出字符串的每个字符,不输出'\0'
%%	%本身

说明:

(1) 在"%"和字母之间加上一个整数表示最大场宽,即输出数据在输出设备(屏幕)所占据的最大宽度(字符个数)。例如:

%3d　表示输出场宽区为 3 的整数,若不够 3 位,则右对齐,左补空格。

%8s　表示输出 8 个字符的字符串,若不够 8 个字符,则右对齐,左补空格。

(2) 在"%"和字母之间加上一个由'.'分隔的两个整数(例如%9.2f),前一个整数表示最大场宽,后一个整数表示小数位数。例如:

%9.2f 表示输出场宽为 9 的浮点数,其中小数位为 2,小数点占 1 位,整数部分自然只余下 6 位,若整数部分不够 6 位,则右对齐,左补空格。

(3) 如果数据的实际值超过所给的场宽,则将按其实际长度输出。但是,对于浮点数,若整数部分位数超过给定的整数位宽度,则将按实际整数位输出,若小数部分位数超过给定的小数位宽度,则按给定的宽度以四舍五入输出。

(4) 如果想在输出数据前补 0 来补足场宽,就应在场宽前加 0。例如,%04d 表示在输出一个小于 4 位的整数时,将在前面补 0,使其总宽度为 4 位。

(5) 如果用类似 6.9 的形式表示字符串的输出格式(如%6.9s),那么小数点后的数字代表最大宽度,小数点前的数字代表最小宽度。

例如,%6.9s 表示输出一个字符串,其输出所占的宽度不小于 6 且不大于 9。若字符个数小于 6,则左补空格补至 6 个字符;若字符个数大于 9,则第 9 个字符以后的内容将不被显示。

(6) 控制输出的数据是左对齐或右对齐。方法是:在"%"和字母之间加入一个"-"号说明输出为左对齐,否则为右对齐。例如:

%-7d　表示输出整数占 7 位场宽,若不足 7 位,则左对齐,右补空格

%-10s 表示输出字符串占 10 位场宽,若不足 10 位,则左对齐,右补空格

案例 03-02-01 使用格式说明符输出数据

案例代码 03-02-01.c

```
#include <stdio.h>
int main(){
    int a=1234, i; char c;
    float f=3.141592653589;
    double x=0.12345678987654321;
    i=12;  c='\x41';
    printf("\n01.a=%d.", a);
    printf("\n02.a=%6d.", a);
    printf("\n03.a=%06d.", a);
    printf("\n04.a=%2d.", a);
    printf("\n05.a=%-6d.",a);
    printf("\n06.f=%f.", f);
    printf("\n07.f=%6.4f.", f);
    printf("\n08.x=%lf.",x);
    printf("\n09.x=%18.16lf.", x);
    printf("\n10.c=%c.", c);
    printf("\n11.c=%x.", c);
    printf("\n12.%s."   ,"ABCDEFGHIJK");
    printf("\n13.%4s."  ,"ABCDEFGHIJK");
    printf("\n14.%14s.","ABCDEFGHIJK");
    printf("\n15.%-14s.","ABCDEFGHIJK");
    printf("\n16.%4.6s.","ABCDEFGHIJK");
    return 0;
}
```

执行程序，输出：

```
01.a=1234.
02.a=␣␣1234.
03.a=001234.
04.a=1234.
05.a=1234␣␣.
06.f=3.141593.
07.f=3.1416.
08.x=0.123457.
09.x=0.1234567898765432.
10.c=A.
11.c=41.
12.ABCDEFGHIJK.
13.ABCDEFGHIJK.
14.␣␣␣ABCDEFGHIJK.
15.ABCDEFGHIJK␣␣␣.
16.ABCDEF.
```

案例拓展 格式说明代码分析

请分析以下程序的输出结果，并理解相关格式说明符的功能。

```
#include <stdio.h>
int main(){
    int a=1234;
    printf("\n01.a=%d.a=%#d", a,a);
    printf("\n02.a=%o.a=%#o", a,a);
    printf("\n03.a=%x.a=%#x", a,a);
    printf("\n05.a=%-8X.a=%#8X", a,a);
    printf("\n06.a=%08X.a=%#08X", a,a);
    return 0;
}
```

3. 标准格式输入函数 scanf()

C 语言中的标准格式输入函数是 scanf()，功能为：从标准输入设备(键盘)上读取用户输入的数据，并将输入的数据赋值给相应的变量。

调用输入函数 scanf()的一般形式为：

```
scanf(格式控制字符串,变量地址列表)
```

功能说明：

(1) 格式控制字符串是用来规定以何种形式从输入设备上接收数据，也就是规定用户以何种格式输入数据。格式控制字符串中包含以下三种字符。

① 格式说明符：这里的格式说明符与 printf()函数中的格式说明符基本相同，一个格式说明符代表一个输入的数据。

② 空白字符：空白字符会使 scanf()函数在读取数据时略去输入数据中的一个或多个空白字符。空白字符包括空格、回车和制表符(Tab 键)。

③ 普通字符：在输入数据时普通字符要原样输入。

(2) 变量地址列表是用逗号分隔的变量地址，与格式说明符一一对应。取变量地址的运算符为 &。

(3) 函数从前向后读取数据，如果突然读到非法数据，则函数结束，程序会往下运行并不报错。但从遇到非法数据的变量开始，以后的变量将得不到输入的值，通常它们的值是随机的(内存中原有的值)。

(4) 格式说明中也可以规定场宽，用来表示接收数据的最大位数。

(5) 读取某个具体整型或实型数据时，遇到第一个不是空白的字符便开始，遇到下列情况时认为该数据结束：空白字符、达到指定宽度、非法输入。

案例 03-02-02 使用格式说明符输入整数

案例代码 03-02-02.c

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

```
    scanf("%d,%d",&c,&d);
    printf("a=%d,b=%d,c=%d,d=%d\n",a,b,c,d);
    printf("a+b+c+d=%d",a+b+c+d);
    return 0;
}
```

执行程序输入：

```
12□34□56,78↙(□表示一个空格,↙表示一个回车,下同)
```

程序输出：

```
a=12,b=34,c=56,d=78
a+b+c+d=180
```

程序分析：

scanf("%d%d",&a,&b);语句中,"%d%d"的意义是：首先读取一个整数赋给变量 a，然后忽略若干空白字符(空格、回车、TAB),最后再读入另一个整数赋给变量 b。

scanf("%d,%d",&c,&d);语句中,"%d,%d"的意义是：读一个整数赋给 c,再读一个逗号,再读一个整数赋给变量 d,逗号作为普通字符要原样读入。

再次执行该程序,输入：

```
12□34□56□78↙
```

程序输出：

```
a=12,b=34,c=56,d=32767
a+b+c+d=32869
```

程序分析：

简单地说,当读完整数 56 并赋给变量 c 之后,程序要求读一个逗号,而此时输入数据(输入流)中剩下的第一个字符是空格,所以 scanf()函数因发生内部错误而结束,变量 d 并没有得到赋值,它的值是不可预期的(内存中原有的值)。

案例拓展 使用格式说明符输入实数

请编程输入 4 个实数,并输出它们的方差,结果保留 4 位小数。请注意格式说明符 %lf 的使用。(提示：方差是各个数据与平均数之差的平方的平均数)

4. 字符型数据输入输出

通过 scanf()函数输入数据时,用户从键盘上输入的一串字符,可以看作一个字符流,程序从该字符流中依次读取数据。

当用 scanf()函数输入字符数据时,如果格式说明符%c 前面没有空格,则输入流中的空白字符将会被当作一个字符读走;如果格式说明符%c 前面有空格,则输入流中的空白

字符将会被忽略，从而读取下一个非空白字符。

C 语言还提供了以下专门的字符输入输出函数。

字符输出函数 putchar(表达式 c)的功能是在标准输出设备(显示器)上输出一个字符，表达式 c 的值为将要输出的字符或其 ASCII 码。

字符输入函数 getchar()的功能是从标准输入设备(键盘)上读入一个字符，返回值为读入的字符(ASCII 码)。

案例 03-02-03 读取字符

案例代码 03-02-03.c

```
#include <stdio.h>
int main(){
  int a1,a2;
  char c1,c2;
  scanf("%d%c",&a1,&c1);
  scanf("%d %c",&a2,&c2);
  printf("a1=%d,c1=%c,c1=%d\n",a1,c1,c1);
  printf("a2=%d,c2=%c,c2=%d\n",a2,c2,c2);
  return 0;
}
```

执行程序，输入：

```
5□□□6□□□A□↙
```

输出：

```
a1=5,c1= ,c1=32
a2=6,c2=A,c2=65
```

程序分析：

语句 scanf("%d%c",&a1,&c1);在正确读入整数 5 给 a1 后，又读取到输入流中的字符空格给 c1。

语句 scanf("%d %c",&a2,&c2);在正确读入整数 6 给 a2 后，由于格式说明符%c 前有一个空格，所以忽略掉若干空白字符后，读取到输入流中的第一个非空白字符 A 给 c2。

案例拓展 字符输入分析

对于以上案例，请分析以下输入时的执行结果，积极思考并参与研讨。

输入样例 1：

```
5,6.
```

输入样例 2：

```
□□□5678□□□90
```

5. 从输入流中跳过某些数据

字符'*'用于格式说明中(例如%*d),表示在输入流中读入数据后不做赋值处理,直接忽略。

案例 03-02-04 某些输入被忽略

案例代码 03-02-04.c

```
#include<stdio.h>
int main(){
  int a,b;
  scanf("%d%*d%*2d%d",&a,&b);
  printf("a=%d,b=%d",a,b);
  return 0;
}
```

执行程序,输入:

```
123 4  56789↙
```

输出:

```
a=123,b=789
```

程序分析:

格式说明"%d%*d%*2d%d"的含义是:先读 1 个整数 123 给 a,再读 1 个整数 4 忽略,再读 1 个 2 位整数 56 忽略,再读整数 789 给 b。

对于上例程序,再次执行程序,输入:123 45 6 789↙,同样输出:a=123,b=789。请分析执行结果。

案例拓展 输入定制

输入数据是一大串数字,要求读取五个数,但要求只处理其中的第 1、3、5 个数,输出这三个数的和。第一个数只读 1 位数,第二个数只读 2 位数,第三个数只读 3 位数,第四个数只读 4 位数,第五个数只读 5 位数。

输入样例:

```
1234567890123456789
```

输出样例:

```
12802    (注:实际为 1+456+12345 的和)
```

6. scanf()函数的返回值

scanf()函数有返回值,具体为成功读入数据的个数。如果开始读入数据时就遇到

“文件结束”(后面没有数据了,在线评测时非常有用),则返回 EOF(－1)。

案例 03-02-05 scanf()函数的返回值

案例代码 03-02-05.c

```
#include<stdio.h>
int main(){
    int a,b,s;
    s=scanf("%d%d",&a,&b);
    printf("a=%d,b=%d,s=%d",a,b,s);
    return 0;
}
```

执行程序,几组输入输出结果如下:

输入样例 1: 23 54 89↙ 输出样例 1: a=23,b=54,s=2	输入样例 2: 23,54↙ 输出样例 2: a=23,b=5064,s=1	输入样例 3: X23A54↙ 输出样例 3: a=356,b=5064,s=0	输入样例 4: ^Z^Z↙ 输出样例 4: a=1,b=0,s=-1

程序分析:

第 1 组输入数据,程序成功读入两个数据,scanf()函数的返回值为 2;

第 2 组输入数据,程序成功读入一个数据,scanf()函数的返回值为 1。变量 b 没有读到数据,其值是不确定的,也是没意义的;

第 3 组数据没有读入成功,scanf()函数的返回值为 0。变量 a 和 b 没有读到数据,其值是不确定的,也是没意义的;

第 4 组数据中输入的是两次 CTRL＋Z,CTRL＋Z 在 Windows 操作系统中表示文件结束符(输入流结束,通常按两次才好使)。因为读第一个数据时就遇到了文件结束符,输入流中无数据可读,所以此时 scanf()函数的返回值为－1,变量 a 和 b 没有读到数据,其值是不确定的,也是没意义的。

案例拓展 输出几个数的和

编程输入最少 1 个最多不超过 4 个整数,输出它们的和。

```
输入样例 1:5 6 7 8          输出样例 1:26
输入样例 2:1 5              输出样例 2:6
输入样例 3:1 5 4            输出样例 3:10
输入样例 4:5                输出样例 4:5
```

温馨提示:输入数据时可以在最后一个数据后输入 CTRL＋Z 两次,再按回车键表示输入流结束。请使用条件运算符完成不同情况的判断。

3.3 顺序结构的应用

案例 03-03-01 统计三个数

编程输入三个整数，输出这三个整数的和及平均值。

这是一个比较简单的问题，但请读者朋友注意三个整数的平均值可能是一个实数，所以在定义变量时，要将代表平均值的变量定义成浮点数。程序如下：

案例代码 03-03-01.c

```
#include<stdio.h>
int main(){
    int num1,num2,num3,sum;
    double aver;
    printf("Please input three numbers:");
    scanf("%d%d%d",&num1,&num2,&num3);          /* 输入三个整数    */
    sum=num1+num2+num3;                         /* 求和            */
    aver=sum/3.0;                               /* 求平均值        */
    printf("num1=%d,num2=%d,num3=%d\n",num1,num2,num3);  /* 输出结果 */
    printf("sum=%d,aver=%lf\n",sum,aver);
    return 0;
}
```

执行程序，在提示信息后输入：

```
Please input three numbers:11 12 13↙
```

输出：

```
num1=11,num2=12,num3=13
sum=36,aver=12.000000
```

案例 03-03-02 三角形面积

输入三角形的三边长 a、b、c，输出其面积 s(假设用户输入的 a、b、c 可以构成三角形)。

已知三角形的三边长，可以利用海伦公式计算它的面积。设三角形的三边长分别为 a、b 和 c，$p=\dfrac{(a+b+c)}{2}$，则计算该三角形面积的海伦公式为 $s=\sqrt{p(p-a)(p-b)(p-c)}$。由此，就能得到以下程序。

案例代码 03-03-02.c

```
#include<stdio.h>
#include<math.h>
int main(){
```

```
    double a,b,c,p,s;
    scanf("%lf%lf%lf",&a,&b,&c);
    p=(a+b+c)/2.0;
    s=sqrt(p*(p-a)*(p-b)*(p-c));
    printf("s=%lf",s);
    return 0;
}
```

执行程序，输入：

```
3 4 5↙
```

输出：

```
s=6.000000
```

再次执行程序，输入：

```
6 8 10↙
```

输出：

```
s=24.000000
```

因为在程序中使用了开平方的函数 sqrt()，所以要在程序开始处加上预处理命令 #include<math.h>。

案例 03-03-03　一元二次方程的两个根

输入一元二次方程的三个系数 *a*、*b*、*c* 的值，输出其两个根（假设方程有实根）。

一元二次方程的求根公式大家一定不陌生，请先编写解决该问题的程序，通过上机运行检验程序的正确性，然后再和下面的程序对比，相信读者朋友们一定会很快地掌握和理解这一问题。

案例代码 03-03-03.c

```
#include<stdio.h>
#include<math.h>
int main(){
    double a,b,c,deta,x1,x2;
    scanf("%lf %lf %lf",&a,&b,&c);
    deta=b*b-4*a*c;
    x1=(-b+sqrt(deta))/(2*a);
    x2=(-b-sqrt(deta))/(2*a);
    printf("\nX1=%lf\nX2=%lf",x1,x2);
    return 0;
}
```

执行程序，输入：

```
1  4  3↙
```

输出：

```
X1=-1.000000
X2=-3.000000
```

执行程序，输入：

```
1 2 1↙
```

输出：

```
X1=-1.000000
X2=-1.000000
```

程序分析：

对于上面的程序，请大家一定要注意避免经常有人会犯的一个错误，那就是将求根的语句写成：

```
x1=-b+sqrt(deta)/2*a;
x2=-b-sqrt(deta)/2*a;
```

请大家分析一下，这样书写程序，会得到正确的结果吗？应该怎样避免出现此类错误呢？

案例 03-03-04 倒序 4 位数

输入一个 4 位的正整数，倒序输出。

关于这个问题，可以这样考虑：要想倒序输出一个整数，需要知道这个整数的每位数字是多少，然后将这些数字倒序输出就可以了。可以在输入语句中依次读入每一位数字，程序如下：

案例代码 03-03-04-A.c

```
#include<stdio.h>
int main(){
    int a,b,c,d;
    scanf("%1d%1d%1d%1d",&a,&b,&c,&d);
    printf("\n%d%d%d%d",d,c,b,a);
}
```

执行程序，输入：

```
1234↙
```

输出：

```
4321
```

再次执行程序，输入：

```
1   23    4567↙
```

输出：

```
4321
```

可以看出，通过输入 4 个 1 位正整数实现输入一个 4 位正整数，所以才会出现后一种输入输出结果。

也可以采取一次读取整个 4 位整数，然后通过运算处理得到它的各个数位上的数字。

案例代码 03-03-04-B.c

```
#include<stdio.h>
int main(){
    int n,a,b,c,d;
    scanf("%d",&n);                    /* 输入整数        */
    a=n/1000;                          /* 取得千位数字    */
    b=n%1000/100;                      /* 取得百位数字    */
    c=n%100/10;                        /* 取得十位数字    */
    d=n%10;                            /* 取得个位数字    */
    printf("%d%d%d%d",d,c,b,a);
    return 0;
}
```

执行程序，输入：

```
1234↙
```

输出：

```
4321
```

执行程序，输入：

```
2500↙
```

输出：

```
0052
```

执行程序，输入：

```
123   4↙
```

输出：

```
3210
```

此程序一次性读取一个完整的 4 位整数，通过取余数和除法操作的解析方法取得各位数字(1 位数)，最后反序一位一位输出。

也可以根据解析出的 4 位数字构造一个新的 4 位数，然后输出(见下例)。

案例代码 03-03-04-C.c

```
#include<stdio.h>
int main(){
    int n,a,b,c,d;
    scanf("%d",&n);                  /* 输入整数                          */
    a=n/1000;                        /* 取得千位数字                      */
    b=n%1000/100;                    /* 取得百位数字                      */
    c=n%100/10;                      /* 取得十位数字                      */
    d=n%10;                          /* 取得个位数字                      */
    n=d*1000+c*100+b*10+a;           /* 重新组合成一个新的 4 位数         */
    printf("\n%04d",n);              /* 输出结果，若不足 4 位，则前补 0   */
    return 0;
}
```

程序执行结果同上例，请读者自行试验其他输入输出案例。

关于如何取得一个整数的各位数字，本程序中已经给读者做出了示范，这一应用技巧在以后还会多次用到，请大家一定掌握。

案例 03-03-05 大小写转换

输入一字母(大写或小写)，输出其对应的另一字母(小写或大写)。

案例分析：

字符型数据和整型数据可以混合运算。一个字符其实就是一个整数，在数值上等于它的 ASCII 码。而一个大写字符的 ASCII 码和其对应的小写字符的 ASCII 码总是相差 32。输入一个字符后，在输出结果以前应该首先判断它是否为一个大写字符，如果是，则输出它加上 32 后的值；如果不是，则输出它减去 32 后的值(前提是用户输入的必须是字母)。具备判断功能的运算符只有条件运算符，所以得到如下程序：

案例代码 03-03-05.c

```
#include<stdio.h>
int main(){
    char n;
    n=getchar();
    putchar(n>=65&&n<=65+25?n+32:n-32);
    return 0;
}
```

执行程序，输入：

```
ABCD↙
```

输出：

```
a
```

执行程序，输入：

```
abcd↙
```

输出：

```
A
```

案例 03-03-06 辛巳蛇宝男

2001 年 01 月 24 日是农历辛巳蛇年的春节(大年初一)，2002 年 02 月 11 日是辛巳蛇年的除夕。赵中瑞同学的生日是 2002 年 01 月 07 日，所以称他为“辛巳蛇宝男”，赵中瑞想知道还有谁和他一样是“辛巳蛇宝男”，试帮他找出来。

输入格式：一行中给出一个中华人民共和国的二代身份证号和姓名，中间没有空格。注意：身份证号倒数第 2 位若为奇数，则为男生；若为偶数，则为女生。为保密，样例中的身份证号前 6 位统一设为 239999。

输出格式：若是“辛巳蛇宝男”，则输出 YES，否则输出 NO。

```
输入样例 1:239999200003132617 于龙        输出样例 1:NO
输入样例 2:239999200002210832 杨冰        输出样例 2:NO
输入样例 3:239999200201131429 张玮娜      输出样例 3:NO
输入样例 4:239999200201210017 刘哲        输出样例 4:YES
```

案例代码：03-03-06.c

```
#include<stdio.h>
int main(){
  int t,date,y;
  scanf("%*6d%8d%*2d%1d",&date,&y);
  printf( date>=20010124&&date<=20020211&&y%2==1?"YES":"NO" );
  return 0;
}
```

案例 03-03-07 X 在哪里

X 同学是好学生。他每天严格按作息时间过着“宿舍—食堂—教室”三点一线的生活。他早 6 点前，晚 6 点后在宿舍学习，早上 6 点至 7 点、中午 12 点至 1 点、下午 5 点至 6 点在食堂吃饭，其余时间在教室上课。你知道 X 现在在哪里吗？(不许用 if 语句和 switch 语句)

输入格式：一行中给出当天的一个时间点，形如：HH:MM:SS，HH 表示小时，MM 表示分，SS 表示秒，全天时间采用 24 小时制表示。

输出格式：根据不同情况，输出一行文本，若确定在宿舍，则输出 dormitory；若确定在食堂，则输出 canteen；若确定在教室，则输出 classroom；两段时间交接处不确定在哪里时，则输出 on the way。

```
输入样例 1:20:10:20                    输出样例 1:dormitory
输入样例 2:06:00:00                    输出样例 2:on the way
输入样例 3:08:00:00                    输出样例 3:classroom
输入样例 4:17:30:00                    输出样例 4:canteen
```

案例代码：03-03-07.c

```
#include<stdio.h>
int main(){
  int h,m,s,t;
  scanf("%d%*c%d%*c%d",&h,&m,&s);
  t=h*10000+m*100+s;
  t>60000&&t<70000 || t>120000&&t<130000 || t>170000&&t<180000 ?
      printf("canteen"):0;
  t>70000&&t<120000 || t>130000&&t<170000 ? printf("classroom"):0;
  t<60000 || t>180000 ? printf("dormitory"):0;
  t==60000||t==70000||t==120000||t==130000||t==170000||t==180000 ?
     printf("on the way"):0;
  return 0;
}
```

案例 03-03-08 日期格式化

世界上不同国家有不同的写日期的习惯。例如，美国习惯写成“月-日-年”，而中国习惯写成“年-月-日”。于龙同学在写一些日期时不小心把年份写错位置了，可能在最前，可能在中间，也可能在最后。请写一个程序，自动把读入的于龙日期改写成中国习惯的日期。

输入格式：在一行中按照“mm-dd-yyyy”或“mm-yyyy-dd”或“yyyy-mm-dd”的格式给出年、月、日。题目保证给出的日期是 1900 年元旦至今合法的日期，保证月在前日在后，保证年份是 4 位数。

输出格式：在一行输出正确的中国人习惯的日期格式，要求月份和日期都用 2 位输出，若不足 2 位，则前补 0。

```
输入样例:5-1-2019                      输出样例:2019-05-01
输入样例:11-2019-05                    输出样例:2019-11-05
```

案例代码 03-03-08.c

```
#include<stdio.h>
int main(){
  int a,b,c,y,m,d;
```

```
    scanf("%d%*c%d%*c%d",&a,&b,&c);
    a>=1000?(y=a,m=b,d=c):
    b>=1000?(y=b,m=a,d=c):
            (y=c,m=a,d=b);
    printf("%4d-%02d-%02d",y,m,d);
    return 0;
}
```

习题 3

一、单项选择题

1. C 语言的程序在编写源文件时，_____。

（A）一行只能写一个语句　　（B）一个语句只能写在一行

（C）一个语句可以写在多行　　（D）语句可在任意处断开写在多行

2. 执行下列程序片段时输出结果是_____。

```
float x=-1023.012;
printf("%8.3f,",x);
printf("%10.2f",x);
```

（A）1023.012，　－1023.012　　（B）－1023.012，　－1023.01

（C）1023.012,－1023.012　　（D）－1023.012，－1023.012

3. 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　20↙B↙　　（B）10　A　20　B↙

（C）10A20B↙　　（D）10A20　B↙

4. 执行下列程序片段时，输出结果是_____。

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

（A）3　　（B）2　　（C）1　　（D）0

5. 若定义 x 为 double 型变量，则能正确输入 x 值的语句是_____。

（A）scanf("%f",x);　　（B）scanf("%d",&x);

（C）scanf("%lf",&x);　　（D）scanf("%o",&x);

二、写出程序的运行结果

1. 写出下列程序的运行结果。

```
#include<stdio.h>
int main(){
    int x=3,y=5,z=7;
    printf("%d,%d\n",(x++,--y),++z);
    return 0;
}
```

2. 写出下列程序的运行结果。

```
#include<stdio.h>
int main(){
    int a=12345;
    double  b=-198.345, c=6.5;
    printf("a=%4d,b=%-10.2e,c=%6.2f\n",a,b,c);
    return 0;
}
```

三、编程题

1. 编程输入圆柱体的底半径 r，高 h，输出其体积。

2. 输入一个华氏温度 F，要求输出摄氏温度 c。转换公式为 $c=\frac{5}{9}(F-32)$。

3. 从键盘输入 5 个整数，求它们的和、平均值并输出。

4. 编写程序，从键盘上输入一个大的秒数，将其转换为几小时几分钟几秒的形式。例如输入 5000，得到的输出为 1 小时 23 分钟 20 秒。

5. 编程，让计算机生成一个 1～6 的随机整数来表示骰子的点数，由用户输入 1 个字符猜大小(规定 1、2、3 点小，4、5、6 点大)，输入字符'1'猜大，输入字符'2'猜小，输出用户猜的是否正确。