

# 第二章 顺序结构

- 第一节 赋值语句
- 第二节 运算符和表达式
- 第三节 常量和变量
- 第四节 标准数据类型
- 第五节 数据输入输出
- 第六节 顺序结构实例

**第一章的简单程序已体现出处理问题的步骤的顺序关系，每条语句按自上而下的顺序依次执行一次，这种自上而下依次执行的程序称为顺序结构程序。**

**在一个程序中，所有的操作都由执行部分来完成，而执行部分又都是由一条条语句组成的。因此，先要学习C++语言的基本语句，并且在学习过程中逐步学会程序设计的基本方法。我们还是先绕过那些繁琐的语法规则细节，先看一些例子，然后给出语法以供选手们参考。**

# 第一节 赋值语句

# 赋值语句

在C++语言中，“=”作为赋值运算符，而不表示“等于”判断。赋值语句是由赋值表达式再加上分号构成的表达式语句，它是程序中使用最多的语句之一。

变量=表达式;

在赋值语句的使用中，需要注意以下几点：

1) 由于赋值运算符“=”右边的表达式也可以是赋值表达式，因此，下述形式：

变量=（变量=表达式）；

是成立的，从而形成嵌套的情形。其展开之后的一般形式为：

变量=变量=...=表达式；

例如 “ a=b=c=d=e=5;” ， 它 实 际 上 等 价 于 ：  
e=5;d=e;c=d;b=c;a=b;

2) 在进行赋值运算时，如果赋值运算符两边的数据类型不同，系统将会自动进行类型转换，即将赋值运算符右边的数据类型转换成左边的变量类型。当左边是整型而右边是实型时，将去掉小数部分并截取该整型对应的有效位数。

**例2.1**输入两个正整数A和B，试交换A、B的值（使A的值等于B，B的值等于A）。

**【分析】** 交换两个变量的值方法很多，一般我们采用引入第三个变量的算法，二个变量交换，可以想像成一瓶酱油和一瓶醋进行交换，这时容易想到拿一个空瓶子过来：

① 将酱油倒到空瓶中；② 将醋倒到酱油瓶中；③ 将原空瓶中的酱油倒到醋瓶中。

程序如下：

```
#include<iostream>           //使用cin,cout, 须调用iostream库
using namespace std;
int main()
{ int a,b,c;                  //定义三个变量
  cout<<"Input a,b=";        //输入提示Input a,b=
  cin>>a>>b;                  //输入A、B的值
  c=a;  a=b;  b=c;            //交换A、B的值
  cout<<"a="<<a<<" b="<<b<<endl; //输出结果
}
```

输入底面半径r和高h，输出圆柱体的表面积，保留3位小数，格式见样例。

### 【分析】

```
#include<stdio>
#include<cmath>
using namespace std;
int main()
{ const double pi=3.1415926;
  double r,h,s1,s2,s;
  scanf("%lf%lf",&r,&h);
  s1=pi*r*r;
  s2=2*pi*r*h;
  s=2*s1+s2;
  printf("Area=%0.3lf\n",s);
  return 0;
}
```

## 例2.3 数学中经典的“鸡兔同笼”问题，已知头共30个，脚共90只，问笼中的鸡和兔各有多少只？

【分析】 设鸡为j只，兔为t只，头为h，脚为f，那么有：

$$j+t=30 \quad \text{①}$$

$$2*j+4*t=90 \quad \text{②}$$

假设笼中30个头全都是兔，那么都按每头4只脚计算，总脚数为(4\*h)，与实际脚数(f)之差为(4\*h-f)，如果这个差=0，则笼中全是兔（即鸡为0只）；如果这个差值 >0，说明多计算了脚数，凡是鸡都多计算了两只脚，用它除以2就能得到鸡的只数，算法为：①  $j=(4*h - f)/2$  //先用脚数差值除以2算出鸡的只数

②  $t=h - j$  //再用总头数减鸡数算出兔的只数

注意这两步运算的先后顺序。程序如下：

```
#include<cstdio>           //使用getchar()语句，须调用cstdio库
#include<iostream>          //使用cin,cout，须调用iostream库
using namespace std;
int main()
{   int h,f,j,t;           //定个变量
    h=30;f=90;             //赋初始值
    j=(4*h-f)/2;           //计算鸡的只数
    t=h-j;                 //计算兔的只数
    cout<<"j="<<j<<" t="<<t<<endl; //输出结果
    getchar();             //和 system("pause")作用相似，用于结果显示暂停
}
```

## 第二节 运算符和表达式

**C++语言中运算符和表达式数量之多，在高级语言中是少见的。正是丰富的运算符和表达式使C++语言功能十分完善。这也是C++语言的主要特点之一。**



# 运算符纵览

## 1. 算术运算符

用于各类数值运算。包括加(+)、减(-)、乘(\*)、除(/)、求余(或称模运算, %)、自增(++)、自减(-- )共七种。

## 2. 关系运算符

用于比较运算。包括大于(>)、小于(<)、等于(==)、大于等于(>=)、小于等于(<=)和不等不等于(!=)六种。

## 3. 逻辑运算符

用于逻辑运算。包括与(&&)、或(||)、非(!)三种。

## 4. 位操作运算符

参与运算的量, 按二进制位进行运算。包括位与(&)、位或(|)、位非(~)、位异或(^)、左移(<<)、右移(>>)六种。

## 5. 赋值运算符

用于赋值运算, 分为简单赋值(=)、复合算术赋值(+=, -=, \*=, /=, %=)和复合位运算赋值(&=, |=, ^=, >>=, <<=)三类共十一种。

## 6. 条件运算符

这是一个三目运算符, 用于条件求值(?:)。

## 7. 逗号运算符

用于把若干表达式组合成一个表达式(, )。

## 8. 指针运算符

用于取内容(\*)和取地址(&)二种运算。

## 9. 求字节数运算符

用于计算数据类型所占的字节数(sizeof)。

## 10. 特殊运算符

有括号(), 下标[], 成员(->, .)等几种。

# 一、算术运算符

## 1、模运算符

求余的运算符“%”也称为模运算符，是双目运算符，两个操作数都是整型数。 $a\%b$ 的值就是 $a$ 除以 $b$ 的余数， $5\%2$ 余数为1。其操作对象只能是整型数，而其他四种运算符对int,float,double,char都适用。

## 2、除法运算符

C++语言的除法运算符有一些特殊之处，即如果 $a$ 、 $b$ 是两个整数类型的变量或常量，那么 $a/b$ 的值是 $a$ 除以 $b$ 的商。例如， $5/2$ 的值是2，而不是2.5，而 $5.0/2$ 或 $5/2.0$ 的值是2.5。

## 3、自增自减运算符

自增、自减运算符用来对一个操作数进行加1或减1运算，其结果仍然赋予该操作数，而且参加运算的操作数必须是变量，而不能是常量或表达式。

1) 自增运算符。例如， $x++$ 表示在使用 $x$ 之后，使 $x$ 的值加1，即 $x=x+1$ ； $++x$ 表示使用 $x$ 之前，先使 $x$ 的值加1，即 $x=x+1$ 。

2) 自减运算符。例如， $x--$ 表示在使用 $x$ 之后，使 $x$ 的值减1，即 $x=x-1$ ； $--x$ 表示使用 $x$ 之前，先使 $x$ 的值减1，即 $x=x-1$ 。

4、复合算术赋值。例如， $a+=1$ ,相当于 $a=a+1$ ； $a+=b$ ,相当于 $a=a+b$ 。

## 例2.4 变量自加运算

- `#include<iostream>`
- `using namespace std;`
- `int main()`
- `{`
- `int x,y,z1,z2;`
- `x=7; y=8;`
- `z1=y-(x++); //计算z1=1, 计算后x=8`
- `z2=y-(++x); //计算前x的值自加1, x的值为9, 再与y求差`
- `cout<< "z1="<<z1<<endl<< "z2="<<z2;`
- `//分别输出z1和z2的值`
- `}`
- 运行结果:
- `z1=1`
- `z2=-1`

## 二、关系运算符

关系运算符用于数值的大小比较。包括大于(>)、小于(<)、等于(==)、大于等于(>=)、小于等于(<=)和不等于(!=)六种，它们都是双目运算符。

关系运算符运算的结果是整型，值只有两种：0或1，0代表关系不成立，1代表关系成立。

请看下面的例子：

```
int main()
{   int n1=4,n2=5,n3;
    n3=n1>n2;           //n3的值为0
    n3=n1<n2;           //n3的值变为1
    n3=n1==4;           //n3的值变为1
    n3=n1!=4;           //n3的值变为0
    n3=n1==1+3;         //n3的值变为1
}
```

### 三、逻辑运算符

C++语言中提供了三种逻辑运算符：与运算(&&)、或运算(||)、非运算(!)。与运算符(&&)和或运算符(||)均为双目运算符。具有左结合性。非运算符(!)为单目运算符，具有右结合性。逻辑运算符和其它运算符优先级的关系可表示如下：

按照运算符的优先顺序可以得出：

$a > b \ \&\& \ c > d$  等价于  $(a > b) \ \&\& \ (c > d)$

$!b == c || d < a$  等价于  $((!b) == c) || (d < a)$

$a + b > c \ \&\& \ x + y < b$  等价于  $((a + b) > c) \ \&\& \ ((x + y) < b)$

逻辑运算的值也为“真”和“假”两种，用“1”和“0”来表示。其求值规则如下：

1.与运算&&参与运算的两个量都为真时，结果才为真，否则为假。例如， $5 > 0 \ \&\& \ 4 > 2$ ，由于 $5 > 0$ 为真， $4 > 2$ 也为真，相与的结果也为真。

2.或运算||参与运算的两个量只要有一个为真，结果就为真。两个量都为假时，结果为假。例如： $5 > 0 || 5 > 8$ ，由于 $5 > 0$ 为真，相或的结果也就为真

3.非运算!参与运算量为真时，结果为假；参与运算量为假时，结果为真。例如： $!(5 > 0)$ 的结果为假。

虽然C++编译在给出逻辑运算值时，以“1”代表“真”，“0”代表“假”。但反过来在判断一个量是为“真”还是为“假”时，以“0”代表“假”，以非“0”的数值作为“真”。例如：由于5和3均为非“0”因此 $5 \ \&\& \ 3$ 的值为“真”，即为1。又如： $5 || 0$ 的值为“真”，即为1。

## 四、位运算符

运算符	含 义	说明	例子
&	按位与	把参与运算的两个数对应的二进制位相与，只有对应的二进制均为1时，结果的对应位才为1，否则为0。	9&5中9可以写成(00001001), 5可以写成(00000101)，那么9&5的运算结果为0000 0001，输出结果是1。
	按位或	把参与运算的两个数对应的二进制位相或，也就是只要对应的两个二进制位有一个为1时，其结果就为1。	9 5相当于00001001 00000101，运算结果是00001101，输出结果是13。
^	按位异或	把参与运算的两个数对应的二进制位相异或，当对应的二进制位上的数据字不相同，结果对应为1时，否则为0。	$1^1=0$ , $1^0=1$ , $0^0=0$ , $0^1=1$ 9^5相当于00001001^00000101，运算结果是00001100，输出结果是12。
~	取反	把运算数的各个二进制位按位求反。	~9相当于~(0000 1001)，运算结果为1111 0110。
<<	左移	把“<<”左边的运算数的各二进制位向左移若干位，“<<”右边的数是指定移动的位数，高位丢弃，低位补0。	a<<4指把a的各二进位向左移动4位，如a=00000011(十进制为3)，左移4位后为00110000(十进制48)。
>>	右移	把“>>”左边的运算数的各二进制位全部右移若干位，“>>”右边的数是指定移动的位数。	设a=15，a>>2表示把00001111右移为0000 0011(十进制为3)。

**需要说明的是，对于有符号数，在右移时，符号位将随同移动。当操作数为正数时，最高位为0，而为负数时，最高位为1。最高位是补0或补1取决于编译系统的规定。**

## 五、运算的简写

在C++语言中，有一些运算可以简写，如下表所示。

简 写	含 义		简 写	含 义
$a+=b$	$a=a+b$		$a\&=b$	$a=a\&b$
$a-=b$	$a=a-b$		$a =b$	$a=a b$
$a*=b$	$a=a*b$		$a\hat{=b}$	$a=a\hat{b}$
$a/=b$	$a=a/b$		$a\<\<=b$	$a=a\<\<b$
$a\%=b$	$a=a\%b$		$a\>\>=b$	$a=a\>\>b$

**算术运算符、关系运算符、逻辑运算符和赋值运算符的优先级如下：**

赋值运算符	逻辑运算符	关系运算符	算术运算符
低		高	

**关系运算符的结合性为：自左至右。**

根据以上优先级和结合性，计算出以下表达式的结果（假设a=3, b=2, c=1）

**a>b**      表达式为真，所以表达式的值为1

**(a>b)==c**      表达式为真，所以表达式的值为1

**b+c<a**      表达式为假，所以表达式的值为0

**d=a>b**      **a>b为真，所以d的值为1**

**f=a>b>c**     a>b为真, 结果为1, 1>c为假, 所以f的值为0



## 六、常用库函数

函数名	格式	功能说明	例子
绝对值函数	<code>abs(x)</code>	求一个数 $x$ 的绝对值	<code>abs(-5)=5</code>
自然数指数函数	<code>exp(x)</code>	求实数 $x$ 的自然指数 $e^x$	<code>exp(1)=2.718282</code>
向下取整	<code>floor(x)</code>	求不大于实数 $x$ 的最大整数	<code>floor(3.14)=3</code>
向上取整	<code>ceil(x)</code>	求不小于实数 $x$ 的最小整数	<code>ceil(3.14)=4</code>
自然对数函数	<code>log(x)</code>	求实数 $x$ 的自然数对数	<code>log(1)=0</code>
指数函数	<code>pow(x, y)</code>	计算 $x^y$ , 结果为双精度实数	<code>pow(2, 3)=8</code>
随机函数	<code>rand()</code>	产生0到RAND-MAX之间的随机整数	
平方根值函数	<code>sqrt(x)</code>	求实数 $x$ 的平方根	<code>sqrt(25)=5</code>

# 【上机练习】

## 1.A+B问题【1.3编程基础之算术表达式与顺序执行01】

大部分的在线题库，都会将A+B问题作为第一题，以帮助新手熟悉平台的使用方法。

A+B问题的题目描述如下：给定两个整数A和B，输出A+B的值。保证A、B及结果均在整型范围内。现在请你解决这一问题。

输入：

一行，包含两个整数A，B，中间用单个空格隔开。A和B均在整型范围内。

输出：

一个整数，即A+B的值。保证结果在整型范围内。

样例输入：

1 2

样例输出：

3

# 【上机练习】

## 2.计算 $(a+b)*c$ 的值【1.3编程基础之算术表达式与顺序执行02】

给定3个整数 $a$ 、 $b$ 、 $c$ ，计算表达式 $(a+b)*c$ 的值。

输入：

输入仅一行，包括三个整数 $a$ 、 $b$ 、 $c$ ，数与数之间以一个空格分开。

(  $-10,000 < a, b, c < 10,000$  )

输出：

输出一行，即表达式的值

样例输入：

2 3 5

样例输出：

25

# 【上机练习】

## 3.计算 $(a+b)/c$ 的值【1.3编程基础之算术表达式与顺序执行03】

给定3个整数 $a$ 、 $b$ 、 $c$ ，计算表达式 $(a+b)/c$ 的值， $/$ 是整除运算。

输入：

输入仅一行，包括三个整数 $a$ 、 $b$ 、 $c$ ，数与数之间以一个空格分开。（ $-10,000 < a,b,c < 10,000$ ， $c$ 不等于0）

输出：

输出一行，即表达式的值。

样例输入：

1 1 3

样例输出：

0

# 【上机练习】

## 4.带余除法【1.3编程基础之算术表达式与顺序执行04】

给定被除数和除数，求整数商及余数。此题中请使用默认的整除和取余运算，无需对结果进行任何特殊处理。

输入：

一行，包含两个整数，依次为被除数和除数（除数非零），中间用一个空格隔开。

输出：

一行，包含两个整数，依次为整数商和余数，中间用一个空格隔开。

样例输入：

10 3

样例输出：

3 1

# 【上机练习】

## 5.计算分数的浮点数值【1.3编程基础之算术表达式与顺序执行05】

两个整数a和b分别作为分子和分母，既分数 $a/b$ ，求它的浮点数值(双精度浮点数，保留小数点后9位)。

输入:

输入仅一行，包括两个整数a和b。

输出:

输出也仅一行，分数  $a/b$  的浮点数值（双精度浮点数，保留小数点后9位）。

样例输入:

5 7

样例输出:

0.714285714

## 第三节 常量和变量

# 一、常量

常量是指在程序中使用的一些具体的数、字符。在程序运行过程中，其值不能被更改。如123,145.88,'m',TRUE等。

1、整型常量：如3、-5、0等。

整型常量是表示整数的常量。有三种表示形式：

1) 十进制形式。如99、-1。

2) 八进制形式。以数字0打头，由0-7构成，如012，表示八进制整数12，即 $12_{(8)}$ 。

3) 十六进制形式。以0x打头，如0x12A，表示十六进制整数12A，即 $12A_{(16)}$ 。

2、实型常量：如3.1、-6.1E+2（科学记数法）。

3、字符常量：是用单引号括起来的字符，如 'k'、'5'、'%'。

注意：'a'与"a"表示的含义是不同的，'a'表示一个字符常量，"a"表示一个字符串。



# 常量的定义

一个常量可以直接调用（如124,'A'），也可以给常量取个名字用一个标识符代表它，这就是符号常量。其语法格式为：

`const 符号常量=常量字符串;`

例如：`const double PI=3.1415926;`

## 例2.5 输入半径r，求圆的周长及面积。

```
#include<cstdio> //调用iostream库，否则使用printf和scanf语句编译出错
using namespace std; //这个程序可省略这行
const double PI=3.1415926; //PI是符号常量。代表3.1415926
int main()
{
    float r,c,s; //定义实型变量
    printf("r="); //显示提示符r=
    scanf("%f",&r); //输入r的值，&符号不能漏掉
    c=2*PI*r; //计算圆的周长
    s=PI*r*r; //计算圆的面积
    printf("c=%.2f s=%.2f\n",c,s); //显示计算结果，结果保留2位小数
}
```

程序中定义的PI代表常量3.1415926，在编译源程序时，遇到PI就用常量3.1415926代替，PI可以和常量一样进行运算。C++语言规定，每个符号常量的定义占据一个书写行，而且符号常量不能被再赋值。如果在例2.5中使用以下赋值语句是错误的。

```
PI=3.1415926;
```

**习惯上，符号常量名用大写，而变量名用小写，以便于区别。**

**使用符号常量的好处：**

**1) 增加了程序的可读性。如看到例2.5程序中，见到PI就可知道它代表圆周率，定义符号常量名时应该尽量使用见名知意的常量名。**

**2) 增加了程序的易改性。如例2.5程序中，只需改动一处，程序中的所有PI都会自动全部代换，做到“一改全改”。**

# 变量定义

变量代表了一个存储单元，其中的值是可以改变的，因此称为变量。如游戏中玩家命的条数最初为3，当你死了一次，命减少一条，这里命的条数就是一个变量（或者说命的条数存储在一个存储单元中）。

一个程序中可能要使用到若干个变量，为了区别不同的变量，必须给每个变量（存储单元）取一个名（称为变量名），该变量（存储单元）存储的值称为变量的值，变量中能够存储值的类型为变量的类型。例如游戏中用于存储“命”的变量，在游戏程序中的存储命的变量名可取为life，它的类型为整型，游戏初始时这个变量的值为3。

# 变量名

- 用一个合法的标识符代表一个变量。如n, m, rot, total等都是合法变量名。在程序中用到的变量要“先定义后使用”，变量名应遵循自定义标识符的命名规则，并建议使用“见名知义”的原则，即用一些有意义的单词作为变量名。在C++语言，变量名大小写有区别。
- 用来标识变量名、符号常量名、函数名、数组名、类型名、文件名的有效字符序列称为标识符。C++语言规定，标识符只能由字母（包含下划线“\_”）开头，后面的字符可以是字母或数字。对于标识符的长度，不同的C++语言编译器有不同的规定，考虑到系统的可移植性，建议变量名的长度不要超过8个字符。例如：month、\_age、s2为合法的标识符；m.k.jack、a<=b、9y为不合法的标识符。

**定义变量的语法格式为：**

**数据类型    变量表**

**例如：**

**int i=5,j,k;                    //定义i,j,k为整型变量,  
                              //i赋初值为5,j,k的初值未知。**

**char a,b,c;                    //定义a,b,c为字符变量**

**float x,y,z;                    //定义x,y,z为实型变量**

**C++语言允许在定义变量的同时为变量赋初值。**

# 变量的类型

- 常量是有类型的数据，变量在某一固定时刻用来存储一个常量，因此也应有相应的类型。如整型变量用来存储整数，实型变量用来存储实数。变量的类型，可以是标准数据类型int、short、long、float、double和char等，也可以是用户自定义的各种类型。
- 变量一经定义系统就在计算机内存中为其分配一个存储空间。在程序中使用到变量时，就在相应的内存中存入数据或取出数据，这种操作称为变量的访问。

# 【上机练习】

## 1.甲流疫情死亡率【1.3编程基础之算术表达式与顺序执行06】

甲流并不可怕，在中国，它的死亡率并不是很高。请根据截止2009年12月22日各省报告的甲流确诊数和死亡数，计算甲流在各省的死亡率。

输入：

输入仅一行，有两个整数，第一个为确诊数，第二个为死亡数。

输出：

输出仅一行，甲流死亡率，以百分数形式输出，精确到小数点后3位。

样例输入：

10433 60

样例输出：

0.575%



# 【上机练习】

## 2.计算多项式的值【1.3编程基础之算术表达式与顺序执行07】

对于多项式 $f(x) = ax^3 + bx^2 + cx + d$ 和给定的 $a, b, c, d, x$ , 计算 $f(x)$ 的值, 保留到小数点后7位。

输入:

输入仅一行, 包含5个实数, 分别是 $x$ , 及参数 $a$ 、 $b$ 、 $c$ 、 $d$ 的值, 每个数都是绝对值不超过100的双精度浮点数。数与数之间以一个空格分开。

输出:

输出一个实数, 即 $f(x)$ 的值, 保留到小数点后7位。

样例输入:

2.31 1.2 2 2 3

样例输出:

33.0838692

# 【上机练习】

## 3.温度表达转化【1.3编程基础之算术表达式与顺序执行08】

利用公式 $C = 5 \times (F - 32) / 9$  (其中C表示摄氏温度, F表示华氏温度)进行计算转化,输入华氏温度f, 输出摄氏温度c, 要求精确到小数点后5位。

输入:

输入一行, 包含一个实数f, 表示华氏温度。 ( $f \geq -459.67$ )

输出:

输出一行, 包含一个实数, 表示对应的摄氏温度, 要求精确到小数点后5位。

样例输入:

41

样例输出:

5.00000

# 【上机练习】

## 4.与圆相关的计算【1.3编程基础之算术表达式与顺序执行09】

给出圆的半径，求圆的直径、周长和面积。输入圆的半径实数 $r$ ，输出圆的直径、周长、面积，每个数保留小数点后4位。

输入：

输入包含一个实数 $r$  ( $0 < r \leq 10,000$ )，表示圆的半径。

输出：

输出一行，包含三个数，分别表示圆的直径、周长、面积，数与数之间以一个空格分开，每个数保留小数点后4位。

样例输入：

3.0

样例输出：

6.0000 18.8495 28.2743

# 【上机练习】

## 5.计算并联电阻的阻值【1.3编程基础之算术表达式与顺序执行10】

对于阻值为 $r1$ 和 $r2$ 的电阻，其并联电阻阻值公式计算如下： $R = 1/(1/r1 + 1/r2)$ 。输入两个电阻阻抗大小，浮点型。输出并联之后的阻抗大小，结果保留小数点后2位。

输入：

两个电阻阻抗大小，浮点型，以一个空格分开。

输出：

并联之后的阻抗大小，结果保留小数点后2位

样例输入：

1 2

样例输出：

0.67

## 第四节 标准数据类型

C++语言提供了丰富的数据类型，本节介绍几种基本的数据类型：整型、实型、字符型。它们都是系统定义的简单数据类型，称为标准数据类型。

# 整型 (integer)

在C++语言中，整型类型标识符为int。根据整型变量的取值范围又可将整型变量定义为以下8种整型类型：

数据类型	定义标识符	占字节数	数值范围	数值范围
短整型	short [int]	2 (16位)	-32768~32767	$-2^{15} \sim 2^{15}-1$
整型	[long] int	4 (32位)	-2147483648~2147483647	$-2^{31} \sim 2^{31}-1$
长整型	long [int]	4 (32位)	-2147483648~2147483647	$-2^{31} \sim 2^{31}-1$
超长整型	long long [int]	8 (64位)	-9223372036854775808~ 9223372036854775807	$-2^{63} \sim 2^{63}-1$
无符号整型	unsigned [int]	2 (16位)	0~65535	$0 \sim 2^{16}-1$
无符号短整型	unsigned short [int]	2 (16位)	0~65535	$0 \sim 2^{16}-1$
无符号长整型	unsigned long [int]	4 (32位)	0~4294967295	$0 \sim 2^{32}-1$
无符号超长整型	unsigned long long	8 (64位)	0~18446744073709551615	$0 \sim 2^{64}-1$

# 表 2-2 实型数取值范围

数据类型	定义标识符	数值范围	占字节数	有效位数
单精度实型	float	$-3.4\text{E}-38 \sim 3.4\text{E}+38$	4 (32位)	6~7位
双精度实型	double	$-1.7\text{E}+308 \sim 1.7\text{E}+308$	8 (64位)	15~16位
长双精度实型	long double	$-3.4\text{E}+4932 \sim 1.1\text{E}+4932$	16 (128位)	18~19位
布尔变量	bool	真true或假false之一	1 (8位)	

# 三、字符型 (char)

字符常量有以下两种表示法:

## 1、普通表示形式

字符常量是由单个字符组成, 所有字符采用ASCII编码, ASCII编码共有128个字符(表2-4)。在程序中, 通常用一对单引号将单个字符括起来表示一个字符常量。如: 'a', 'A', '0'等。如字符A的序号是65, 字符a的序号是97, 字符0的序号的48。

## 2、转义字符表示形式。

转义字符有三种用法: 表示控制字符、表示特殊字符、表示所有字符。常用的转义字符如右表所示。

转义字符	含义
'\n'	换行
'\t'	水平制表
'\b'	退格
'\r'	回车 (不换行)
'\0'	空字符
'\"'	单引号
'\''	双引号
'\\'	一个反斜杠字符
'\ddd'	1~3位八进制数所代表的字符
'\xhh'	1~2位十六进制数所代表的字符



# ASCII编码表见下图:

序号	字符	序号	字符	序号	字符	序号	字符	序号	字符	序号	字符
32	空格	48	0	64	@	80	P	96	`	112	p
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	\$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	(	56	8	72	H	88	X	104	h	120	x
41	)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[	107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	
45	-	61	=	77	M	93	]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o	127	deL

## 例2.6 整型数据类型存储空间大小

分别定义int, short类型的变量各一个, 并依次输出它们的存储空间大小(单位: 字节)。

### 【参考程序】

```
#include<iostream>
using namespace std;
int main()
{
    int x;
    short y;
    cout<<sizeof(x)<<" "<<sizeof(y)<<endl;
        //sizeof返回一个对象或者类型所占的内存字节数
    return 0;
}
```

基于char数据的这个特点，它可以和int变量一样做加减运算，例2.7说明了这种用法。

## 例2.7 大小字母的转换

```
#include<iostream>
using namespace std;
int main()
{ char c1,c2;
  c1='a';
  c2='A';
  cout<<c1<<" " <<c2<<endl;
  c1=c1-32;           //小写字母转换大写字母
  c2=c2+32;           //大写字母转换小写字母
  cout<<c1<<" " <<c2<<endl;
}
```

运行结果：

a A

A a

因为所有小写字母的ASCII值要比对应大写字母的ASCII值大32，所以c1减去32后便得到原来字母的大写形式。反之，c2加上32后便得到原来字母的小写形式。

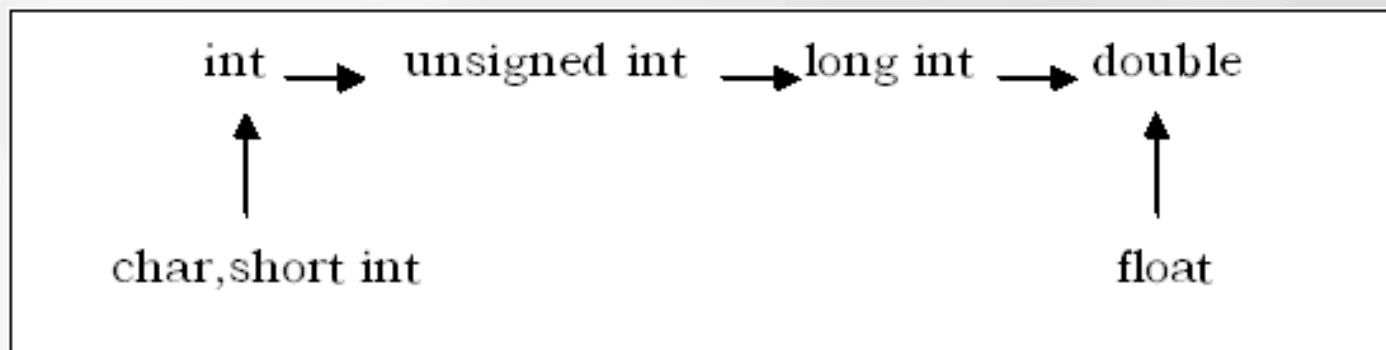
## 四、数据类型转换

**C++语言中，不同数据类型的运算对象进行混合运算，或者需要将一个表达式的结果转换成期望的类型时，就需要依据数据类型转换规则进行转换。**

# 1、混合运算时的类型转换规则

整型、实型、字符型数据间可以混合运算。在这种情况下，需要将不一致的数据类型转换成一致的数据类型，然后进行运算。为了保证运算精度，系统在运算时的转换规则是将存储长度较短的运算对象转成存储长度较长的类型，然后再进行处理。这种转换是系统自动进行的，具体见表2-5所示。

混合运算时的类型转换规则



## 【说明如下】

- 1) 纵向箭头表示必定会进行的转换，如float型数据必先转换为double型数据，然后与其他操作数进行运算。与此类似，char型或short型数据必先转换为int型数据，然后进行运算。
- 2) 横向箭头表示当运算对象为不同类型数据时的转换方向，如int型数据与unsigned型数据进行运算，int型转换为unsigned型后方可进行运算。int型与double型进行运算，int型直接转换为double型后进行运算，不能理解为先转换为unsigned int型，然后转换为long int型，最后再转换为double型。

## 2、赋值时的类型转换规则

当赋值运算符两侧的数据类型不同时，需进行类型转换，这种转换是系统自动进行的，转换规则如下：

1) float、double型赋值给int型：直接截断小数。

例如，“int i=f+0.6;”，f的值为4.0，右边算术表达式运算后的结果为4.6的double型数据，根据上述转换原则，直接舍弃小数，所以i的值为4。

2) int、char型赋值给float、double型：补足有效位以进行数据类型转换。

例如：“float f=4;”，float为7位有效数字，所以f的值为4.0000000。

3) char型(1字节)赋给int型(4字节)：数值赋给int型的低8位，其它位补0。

4) long int型赋值给int型：long int型截断低字节给int型。

5) int型赋值给long int型：赋给long int型的低16位，如果int型的最高位是0，则long int的高16位全为0；如果int型的最高位是1，则long int型的高16位全为1（称为“符号扩展”）。

6) unsigned int型赋值给int型：直接传送数值。

7) 非unsigned int型赋值给位数相同的unsigned int型：直接传送数值。

### 3、强制类型转换

在C++语言中，还允许强制类型转换，即将某一数据的数据类型转换为指定的另一种数据类型，强制类型转换只是临时转换。强制转换运算符组成的运算表达式的一般形式为：（类型名）（表达式）

例如：已知有变量定义“`int b=7;float a=2.5,c=4.7;`”，求下面算术表达式的值。

`a+(int)(b/3*(int)(a+c)/2.0)%4`

根据运算符的结合性规则，表达式要自左至右执行，`b/3`为2，`2*(int)(a+c)`为14，`14/2.0`为7.0，强制类型转换后为7，`7%4`为3，`a`的值2.5与3相加，最终结果为5.5。



# 【上机练习】

## 1.整型数据类型存储空间大小【1.2编程基础之变量定义、赋值及转换01】

分别定义int, short类型的变量各一个, 并依次输出它们的存储空间大小(单位: 字节)。

输入:

无。

输出:

一行, 两个整数, 分别是两个变量的存储空间大小, 用一个空格隔开。

## 2.浮点型数据类型存储空间大小【1.2编程基础之变量定义、赋值及转换02】

分别定义float, double类型的变量各一个, 并依次输出它们的存储空间大小(单位: 字节)。

输入:

无。

输出:

一行, 两个整数, 分别是两个变量的存储空间大小, 用一个空格隔开。

# 【上机练习】

## 3.其他数据类型存储空间大小【1.2编程基础之变量定义、赋值及转换03】

分别定义bool, char类型的变量各一个, 并依次输出它们的存储空间大小(单位: 字节)。

输入:

无。

输出:

一行, 两个整数, 分别是两个变量的存储空间大小, 用一个空格隔开。

## 4.浮点数向零舍入【1.2编程基础之变量定义、赋值及转换06】

输入一个单精度浮点数, 将其向零舍入到整数。说明: 向零舍入的含义是, 正数向下舍入, 负数向上舍入。提示: 可以使用强制类型转换来实现。

输入:

一个单精度浮点数。

输出:

一个整数, 即向零舍入到整数的

结果。

样例输入:

2.3

样例输出:

2

# 【上机练习】

## 5.打印ASCII码【1.2编程基础之变量定义、赋值及转换07】

输入一个除空格以外的可见字符（保证在函数scanf中可使用格式说明符%c读入），输出其ASCII码。

输入：

一个除空格以外的可见字符。

输出：

一个十进制整数，即该字符的ASCII码。

样例输入：

A

样例输出：

65

# 【上机练习】

## 6.打印字符【1.2编程基础之变量定义、赋值及转换08】

输入一个整数，即字符的ASCII码，保证存在对应的可见字符。输出相对应的字符。

输入：

一个整数，即字符的ASCII码，保证存在对应的可见字符。

输出：

一行，包含相应的字符。

样例输入：

65

样例输出：

A

# 【上机练习】

## 7.整型与布尔型的转换【1.2编程基础之变量定义、赋值及转换09】

将一个整型变量的值赋给一个布尔型变量，再将这个布尔型变量的值赋给一个整型变量，得到的值是多少？

输入：

一个整型范围内的整数，即初始时整型变量的值。

输出：

一个整数，经过上述过程后得到的结果。

样例输入：

3

样例输出：

1

# 【上机练习】

## 7.整型与布尔型的转换【1.2编程基础之变量定义、赋值及转换09】

将一个整型变量的值赋给一个布尔型变量，再将这个布尔型变量的值赋给一个整型变量，得到的值是多少？

输入：

一个整型范围内的整数，即初始时整型变量的值。

输出：

一个整数，经过上述过程后得到的结果。

样例输入：

3

样例输出：

1

# 【上机练习】

## 8. Hello, World! 的大小【1.2编程基础之变量定义、赋值及转换10】

在上一章里，我们曾经输出过的“Hello, World!”吗？它虽然不是本章所涉及的基本数据类型的数据，但我们同样可以用sizeof函数获得它所占用的空间大小。

输入：

无。

输出：

一个整数，即“Hello, World!”的大小。

## 第五节 数据输入输出

**C++语言中没有提供专门的输入输出语句，所有的输入输出都是调用标准库函数中的输入输出函数来实现的。在使用时，应在源程序的开头使用如下语句：**

```
#include<iostream>  
#include<cstdio>  
using namespace std;
```

**C++语言标准函数库提供了许多标准输入、输出函数，本节将介绍6个最基本的输入、输出函数：字符输入getchar，字符输出putchar，格式化输入scanf，格式化输出printf，流输入cin，流输出cout。**



# 一、字符输入函数getchar

**getchar函数是接收从键盘输入的单个字符数据。它是一个无参函数，其语法格式为：**

```
getchar();
```

## **【说明】**

**1)通常把输入的字符赋予一个字符变量，构成赋值语句。例如：**

```
char ch;
```

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

**2) getchar函数只能接受单个字符，输入数字也按字符处理。**

**3)输入多于一个字符时，只接收第一个字符。**

**4) getchar函数等待用户输入，直到按回车键才结束，可用于暂停程序的运行，直到输入一个回车键。**

**5)如果在程序中连续有两个以上getchar()函数，应该一次性输入所需字符，最后再按回车键，否则会把回车作为一个字符传给后面的getchar()函数。**

## 例2.8 利用getchar函数接收键盘输入。

```
#include<cstdio>
#include<iostream>
using namespace std;
int main()
{
    char ch;
    ch=getchar();           //读入字符
    cout<<"input="<<ch<<endl;
}
```

## 二、字符输出函数putchar

putchar函数是字符输出函数，功能是向标准输出设备(如显示器)输出单个字符数据，其语法格式为：

putchar(ch);                      //其中，ch为一个字符变量或常量。

例2.9 利用putchar 函数输出字符。

```
#include <cstdio>
#include <iostream>
using namespace std;
int main()
{
    char c='B';                      //定义字符变量c并赋值'B'
    putchar(c);                      //输出该字符
    putchar('\x42');                //用转义字符输出字母'B'
    putchar(0x42);                  //用16进制ASCII码值输出字母'B'
    putchar(66);                    //用10进制ASCII码值输出字母'B'
}
```

运行结果：BBBB

### 三、通过cout流输出数据

流插入运算符<<和cout结合在一起使用，可向显示器屏幕输出数据。

#### 格式1:

**功能：**它把表达式的值输出到屏幕上，该表达式可以是各种基本类型的常量、变量或者由它们组成的表达式。输出时，程序根据表达式的类型和数值大小，采用不同的默认格式输出，大多数情况下可满足要求。

**若要输出多个数据，可以连续使用流插入运算符**

#### 格式2:

**功能：**将表达式的内容一项接一项的输出到屏幕上。

# 1. 输出字符串和输出变量的区别

每当我们输出字符串常量的时候，必须用双引号把字符串引起来，以便将它和变量名明显的区分开来。

## 例2.10

下面两个语句是不同的：

<code>cout &lt;&lt; "Hello" ;</code>	<code>//打印字符串Hello到屏幕上</code>	<code>cout &lt;&lt; Hello ;</code>
<code>cout &lt;&lt; Hello;</code>	<code>//把变量Hello存储的内容打印到屏幕上</code>	

## 2. 如何增强信息的可读性

为了增强输出信息的可读性，在输出多个数据时可以通过插入空格符，换行符或其他提示信息将数据进行组织，以获得更好的效果。

### 例2.11

```
x=12;
```

```
cout << "tom is my friend,he is";
```

```
cout << x;
```

```
cout << "years old";
```

输出结果为tom is my friend,he is12year old

# 3. 换行符的使用

必须注意，除非我们明确指定，`cout`并不会自动在其输出内容的末尾加换行符，因此下面的语句：

例2.12

```
cout << "This is a sentence." ;  
cout << "This is another sentence." ;
```

将会有如下内容输出到屏幕：

This is a sentence.This is another sentence.

虽然我们分别调用了两次`cout`，两个句子还是被输出在同一行。所以，为了在输出中换行，我们必须插入一个换行符来明确表达这一要求，在C++中换行符可以写作`\n`。

```
cout << "First sentence.\n";  
cout << "Second sentence.\nThird sentence.";
```

将会产生如下输出：

First sentence.  
Second sentence.  
Third sentence.

另外，你也可以用操作符`endl`来换行，例如：

```
cout << "First sentence." <<  
endl;  
cout << "Second sentence." <<  
endl;
```

将会输出：

First sentence.  
Second sentence.

## 例2.13

在屏幕上输出

2 3

4

```
cout<<2<<" "<<3<<endl;
```

```
cout<<4;
```

```
或 cout<<"2 3\n4";
```

除了以上两种写法外，还可以有其它的的写法，请试试看。你可以使用  
`\n`或`endl`来指定`cout`输出换行，注意两者的不同用法。



## 四、通过cin流读入数据

流读取运算符>>和cin结合在一起使用，可从键盘输入数据。

格式1:

功能：是从键盘读取一个数据并将其赋给“变量”。

说明：在使用cin输入的时候必须考虑后面的变量类型。如果你要求输入一个整数，在 >>后面必须跟一个整型变量，如果要求一个字符，后面必须跟一个字符型变量。

例2.14声明一个整型变量age然后等待用户从键盘输入到cin并将输入值存储在这个变量中。

```
int age;
```

```
cin >> age;
```

也可以连续使用>>，实现从键盘对多个变量输入数据。

格式2:

这要求从键盘输入的数据的个数、类型与变量相一致。从键盘读取数据时，各数据之间要有分隔符，分隔符可以是一个或多个空格键、回车键等。

## 例2.15用cin 让用户输入多个数据 。

```
cin >> a >> b;
```

等同于:

```
cin >> a;
```

```
cin >> b;
```

## 例2.16流读取运算符>>和cin的使用

```
#include<iostream>
using namespace std;
int main( )
{ char c;
  int i;
  float x,y;
  cout<<"enter:\n";
  cin >>i>>x>>y;
  c=i;
  cout<<"c="<<c<<"\ti="<<i<<"\n";
  cout<<"x="<<x<<"\ty="<<y<<"\n";
  return 0;
}
```

**程序运行时屏幕先显示：**

**enter:**

**这时从键盘输入一个整数和两个实数，中间用一个或多个空格键作分隔符。 如输入**

**65 2.3 3.5**

**最后屏幕显示：**

**c=A    i=65**

**x=2.3   y=3.5**

**程序中"\n"和"\t"都是转义符。 '\n '和"\n"效果相同，都表示换行符。**

**'\t'是制表符，可以理解为连续输出几个空格。字符变量和整型变量i的值都是65，但输出的形式不同。**

# 五、格式化输入函数scanf

scanf函数的功能是格式化输入任意数据列表，其一般调用格式为：

scanf(格式控制符，地址列表)

**【说明】**

- 1) 地址列表中给出各变量的地址，可以为变量的地址，也可以为字符串的首地址。
- 2) 格式控制符由%和格式符组成，作用是将要输入的字符按指定的格式输入，如%d，%c等。

表2-6 scanf函数的格式符

格式符	说 明
d, i	用于输入十进制整数
u	以无符号十进制形式输入十进制整数
o(字母)	用于输入八进制整数
x	用于输入十六进制整数
c	用于输入单个字符
s	用于输入字符串（非空格开始，空格结束，字符串变量以'\0'结尾）
f	用于输入实数（小数或指数均可）
e	与f相同（可与f互换）

**表2-7    scanf函数的附加格式说明符**

附加格式	说 明
l (字母)	用于长整型数 (%ld、%lo、%lx) 或double型实数 (%lf、%le)
h	用于短整型数 (%hd、%ho、%hx)
域宽 (一个整数)	指定输入所占列宽
*	表示对应输入量不赋给一个变量

### 例.1 \*格式符的使用

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

输入: 1 2 3 回车

输出: a=1,b=3

### 例.3 %s 格式符的使用

```
#include<stdio>
int main()
{
    char st[40];
    scanf("%s",st);
    printf("Your input is:%s\n",st);
}
```

输入: abc hello

输出: abc

### 例.2 域宽格式符的使用

```
#include<stdio>
int main()
{
    int a,b;
    scanf("%4d%4d",&a,&b);
    printf("a=%d,b=%d\n",a,b);
}
```

输入: 1234567 回车

输出: a=1234,b=567

### 例.4 非格式符的使用

```
#include<stdio>
int main()
{
    int a,b,c;
    scanf("%d,%d,%d",&a,&b,&c);
    printf("a=%d,b=%d,c=%d\n",a,b,c);
} //例中 scanf 用非格式字符',' 作间隔符
```

故输入应为: 5,6,7 (本来输入:5 6 7)

输出: a=5,b=6,c=7

## 六、格式化输出函数printf

printf函数的功能是格式化输出任意数据列表，其一般调用格式为：

**printf(格式控制符, 输出列表)**

### 【说明】

1) 格式控制由输入格式说明和普通字符组成，必须用双引号括起来。

- 格式说明由%和格式字符组成，作用是将要输出的字符转换为指定的格式，如%d, %c等。

- 普通字符是在输出时原样输出的字符，一般在显示时起提示作用。

2) 输出列表是需要输出的一组数据（可以为表达式和变量），各参数之间用“,”分开。要求格式说明和各输出项在数量和类型上要一一对应，否则将会出现意想不到的错误。



## 表2-9 d格式符

参 数	说 明
%d	输出数字长为变量数值的实际长度
%md	输出m位(不足补空格, 大于m位时按实际长度输出)
%-md	m含义同上。左对齐输出
%ld	l(小写字母)表示输出“长整型”数据
%mld	指定长整型输出宽度m位, 左边补空格; 否则, 按实际位数输出
%0md, %0mld	0(数字0)表示位数不足m时补0

例如, 对如下程序段

```
int i=1;
```

```
long j=123;
```

```
printf("%d,%2d,%03d,%1d,%-4ld,%05ld",i,i,i,j,j,j);
```

输出: 1, 1,001,123,123 ,00123

**表2-10 f格式符**

参 数	说 明
%f	按实数格式输出，整数部分按实际位数输出，6位小数
%m. nf	总位数m（含小数点），其中有n位小数
%-m. nf	同上，左对齐

**表2-11 s格式符**

参 数	说 明
%s	按实际宽度输出一个字符串
%ms	m指定宽度（不足时左补空格，大于时按实际宽度输出）
%-ms	左对齐，不足时右补空格
%m. ns	输出占m个字符位置，其中字符数最多n个，左补空格
%-m. ns	同上，右补空格

**例2.17** 某幼儿园里，有5个小朋友编号为1, 2, 3, 4, 5，他们按自己的编号顺序围坐在一张圆桌旁。他们身上都有若干个糖果(键盘输入)，现在他们做一个分糖果游戏。从1号小朋友开始，将自己的糖果均分三份（如果有多余的糖果，则立即吃掉），自己留一份，其余两份分给他的相邻的两个小朋友。接着2号、3号、4号、5号小朋友同样这么做。问一轮后，每个小朋友手上分别有多少糖果。

**【分析】**题目中有5位小朋友，他们初始时糖果的数目不确定，用a,b,c,d,e分别存储5个小朋友的糖果数，初始值由键盘输入。

程序如下：

```
#include <stdio>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int a,b,c,d,e;
```

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

```
    a=a/3;b=b+a;e=e+a;
```

//1号小朋友分糖

```
    b=b/3;c=c+b;a=a+b;
```

//2号小朋友分糖

```
    c=c/3;d=d+c;b=b+c;
```

//3号小朋友分糖

```
    d=d/3;e=e+d;c=c+d;
```

//4号小朋友分糖

```
    e=e/3;a=a+e;d=d+e;
```

//5号小朋友分糖

```
    printf("%5d%5d%5d%5d%5d\n",a,b,c,d,e); // %5d按5位宽度输出
```

```
    return 0;
```

```
}
```

运行结果：

输入：8 9 10 11 12

输出： 11 7 9 11 6

## 七、几种输入输出格式的几点说明

1、cin和cout在Dev C++中只能调用<iostream>库，而其他输入输出格式要调用<stdio.h>库或<cstdio>库。

2、cin和cout属于C++的概念，调用时涉及输入输出流，而scanf和printf属于C的概念，是C语言的标准输入/输出库中的函数，所以在时效上，scanf和printf优于cin、cout，对于大数据的输入输出，通常情况下应该用scanf、printf。

3、对于普通数据的输入输出，cin和cout比较方便，而在格式化方面，scanf和printf比较容易。cin效率比scanf低很多，尤其输入数据达到20万以上时非常明显，在100万时cin读入就会超时。

4、scanf和printf也缺点，cin和cout能够自动识别变量的数据类型，因此，在进行输入输出时，不需要指定数据类型，printf和scanf函数在输入输出时需指定数据类型。

# 【上机练习】

## 1.输出保留3位小数的浮点数【1.1编程基础之输入输出04】

读入一个单精度浮点数，保留3位小数输出这个浮点数。

输入：

只有一行，一个单精度浮点数。

输出：

也只有一行，读入的单精度浮点数。

样例输入：

12.34521

样例输出：

12.345

# 【上机练习】

## 2.输出保留12位小数的浮点数【1.1编程基础之输入输出05】

读入一个双精度浮点数，保留12位小数，输出这个浮点数。

输入：

只有一行，一个双精度浮点数。

输出：

也只有一行，保留12位小数的浮点数。

样例输入：

3.1415926535798932

样例输出：

3.141592653580

# 【上机练习】

## 3.空格分隔输出【1.1编程基础之输入输出06】

读入一个字符，一个整数，一个单精度浮点数，一个双精度浮点数，然后按顺序输出它们，并且要求在他们之间用一个空格分隔。输出浮点数时保留6位小数。

输入：

第一行是一个字符；

第二行是一个整数；

第三行是一个单精度浮点数；

第四行是一个双精度浮点数。

输出：

输出字符、整数、单精度浮点数和双精度浮点数，之间用空格分隔。

样例输入：

a  
12  
2.3  
3.2

样例输出：

a 12 2.300000 3.200000

# 【上机练习】

## 4.输出浮点数【1.1编程基础之输入输出07】

读入一个双精度浮点数，分别按输出格式“%f”，“%f”保留5位小数，“%e”和“%g”的形式输出这个整数，每次在单独一行上输出。

输入：

一个双精度浮点数。

输出：

第一行是按“%f”输出的双精度浮点数；

第二行是按“%f”保留5位小数输出的双精度浮点数；

第三行是按“%e”输出的双精度浮点数；

第四行是按“%g”输出的双精度浮点数。

样例输入：

12.3456789

样例输出：

12.345679

12.34568

1.234568e+001

12.3457



# 【上机练习】

## 5.字符菱形【1.1编程基础之输入输出09】

给定一个字符，用它构造一个对角线长5个字符，倾斜放置的菱形。

输入：

输入只有一行， 包含一个字符。

输出：

该字符构成的菱形。

样例输入：

\*

样例输出：

\*

\*\*\*

\*\*\*\*\*

\*\*\*

\*

## 第六节 顺序结构实例

**我们已经学习了数据输入输出、赋值语句以及基本的数据类型。下面举一些实例，通过阅读和模仿这些程序，让选手逐步熟悉程序的编写和巩固知识点，为以后各章的学习打好基础。**

**例2.18** 输入一个三位数，要求把这个数的百位数与个位数对调，输出对调后的数。

**【分析】**先求出自然数的个位、十位、百位，然后个位与百位对调。  
程序如下：

```
#include<iostream>
using namespace std;
int main()
{ int a,b,c,m,n;
  cin>>m;           //输入一个三位数
  a=m/100;           //百位数
  b=(m/10)%10;       //十位数
  c=m%10;             //个位数
  n=c*100+b*10+a;    //重新组合对调后的数
  cout<<"n="<<n<<endl; //输出结果
}
```

**运行结果：**

**输入：234**

**输出：n=432**

**例2.19** 已知某班有男同学 $x$ 位，女同学 $y$ 位， $x$ 位男生平均分是87分， $y$ 位女生的平均分是85，问全体同学平均分是多少分？

**【分析】** 男女生的人数需要用户输入，然后根据题意 $(x*87+y*85)/(x+y)$ 求出全体同学的平均分。

程序如下：

```
#include<iostream>
using namespace std;
int main()
{
    int x,y;
    cin>>x>>y;                //输入男女人数
    cout<<float(x*87+y*85)/(x+y)<<endl;
    //数据类型强制转换，按实数格式输出，4位小数
}
```

**例2.20** 歌手大奖赛上6名评委给一位参赛者打分，6个人打分的平均分为9.6分；如果去掉一个最高分，这名参赛者的平均分为9.4分；如果去掉一个最低分，这名参赛者的平均分为9.8分；如果去掉一个最高分和一个最低分，这名参赛者的平均是多少？

**【分析】** 首先求出6名评委的总分，然后根据去掉最高分的总分和最低分的总分，求出最高分的分值和最低分的分值，最后总分减去最高分和最低分除以4即是答案。

程序如下：

```
#include<stdio>

int main()
{ float high,low,sc_all,sc_high,sc_low,ans;
  sc_all=6*9.6;           //求6名评委的总分
  sc_high=5*9.4;          //求去掉最高分后的总分
  sc_low=5*9.8;           //求去掉最低分后的总分
  high=sc_all-sc_high;    //求最高分
  low=sc_all-sc_low;      //求最低分
  ans=(sc_all-high-low)/4; //求平均分
  printf("%5.2f\n",ans);  //按实数格式输出，保留2位小数
}
```

**//运行结果： 9.60**

**例2.21** 传说古代的叙拉古国王海伦二世发现的公式，利用三角形的三条边长来求取三角形面积。已知 $\triangle ABC$ 中的三边长分别为 $a, b, c$ ，求 $\triangle ABC$ 的面积。(提示：海伦公式

$$s = \sqrt{p(p-a)(p-b)(p-c)}$$

，其中 $p=(a+b+c)/2$  )

**【分析】** 公式中 $p$ 是三角形周长的一半，求出 $p$ 后直接代入海伦公式中求得面积。

```
#include <cstdio>                                //Dev C++可调用<iostream>和<stdio.h>库
#include <math.h>                                //在Dev C++中可调用数学函数库cmath
int main()
{ float a,b,c,p,s;
  scanf("%f%f%f",&a,&b,&c); //输入三角形的三边
  p=(a+b+c)/2;           //求出p的值
  s=sqrt(p*(p-a)*(p-b)*(p-c)); //根据p求面面积，sqrt是开方函数
  printf("%0.3f\n",s);      //输出面积，0.3f按实际位数输出，保留3位小数
}
```

**运行结果：**

输入：3 4 5    输出：6.000

**例2.22 分钱游戏。**甲、乙、丙三人共有24元钱，先由甲分钱给乙、丙两人，所分给的数与各人已有数相同；接着由乙分给甲、丙，分法同前；再由丙分钱给甲、乙，分法亦同前。经上述三次分钱之后，每个人的钱数恰好一样多。求原先各人的钱数分别是多少？

**【分析】** 设甲、乙、丙三人的钱数分别为A，B，C。用倒推(逆序)算法，从最后结果入手，按反相顺序，分步骤推算出每次各人当时的钱数：（在每个步骤中，各人钱数分别存在A、B、C中）

```
#include<cstdio>      //Dev C++可调用<iostream>和<stdio.h>库
int main()             //调用iostream库，使用printf语句编译通不过
{ int a,b,c;
  a=8; b=8; c=8;        //对应于步骤①
  a=a/2;b=b/2;c=a+b+c;  //对应于步骤②
  a=a/2;c=c/2;b=a+b+c;  //对应于步骤③
  b=b/2;c=c/2;a=a+b+c;  //对应于步骤④
  printf("a=%-5db=%-5dc=%-5d\n",a,b,c);
                        //%-5d按5位宽度输出，左对齐。
}
```

**运行结果：**

**a=13   b=7   c=4**

## 例2.23 求一元二次方程 $x^2+3x+2=0$ 的两个实数根。

【分析】方程的系数是常量，分别用A, B, C表示，运用数学求方程的根，采取如下方法：

- ① 先求出 $d=B^2-4AC$ ；（求根公式中需用开方运算的那部分）
- ② 再用求根公式算出 $x_1, x_2$ 的值。（ $x_1, x_2 = ?$ ）
- ③ 输出 $x_1, x_2$ 。

```
#include <stdio.h>           //Dev C++ 可调用 <stdio.h> 库
#include <math.h>             //在Dev C++ 中可调用数学函数库cmath
#define A 1
#define B 3                   //常量说明，A, B, C表示方程系数
#define C 2
int main()
{
    int d;                    //d为整型变量
    float x1,x2;              //x1, x2为实型变量
    d=B*B-4*A*C;
    x1=(-B+sqrt(d))/(2*A);    //求方程的根
    x2=(-B-sqrt(d))/(2*A);
    printf("x1=%-8.3fx2=%-8.3f\n",x1,x2); //%-8.3f按8位宽度，3位小数，左对齐
} //运行结果：
// x1=-1.000 x2=-2.000
```



# 【上机练习】

## 1.计算浮点数相除的余【1.3编程基础之算术表达式与顺序执行11】

计算两个双精度浮点数a和b的相除的余数，a和b都是双精度浮点数。这里余数（r）的定义是： $a=k*b+r$ ，其中k是整数， $0 \leq r < b$ 。

输入：

输入仅一行，包括两个双精度浮点数a和b。

输出：

输出也仅一行， $a \div b$ 的余数

样例输入：

73.263 0.9973

样例输出：

0.4601

# 【上机练习】

## 2.计算球的体积【1.3编程基础之算术表达式与顺序执行12】

对于半径为 $r$ 的球，其体积的计算公式为 $V=4/3*\pi r^3$ ，这里取 $\pi=3.14$ 。现给定 $r$ ，即球半径，类型为double，求球的体积 $V$ ，保留到小数点后2位。

输入：

输入为一个不超过100的非负实数，即球半径，类型为double。

输出：

输出一个实数，即球的体积，保留到小数点后2位。

样例输入：

4

样例输出：

267.95

# 【上机练习】

## 3.反向输出一个三位数【1.3编程基础之算术表达式与顺序执行13】

将一个三位数反向输出，例如输入358，反向输出853。

输入：

一个三位数n。

输出：

反向输出n。

样例输入：

100

样例输出：

001

# 【上机练习】

## 4.大象喝水查【1.3编程基础之算术表达式与顺序执行14】

一只大象口渴了，要喝20升水才能解渴，但现在只有一个深 $h$ 厘米，底面半径为 $r$ 厘米的小圆桶( $h$ 和 $r$ 都是整数)。问大象至少要喝多少桶水才会解渴。

输入：

输入有一行：包行两个整数，以一个空格分开，分别表示小圆桶的深 $h$ 和底面半径 $r$ ，单位都是厘米。

输出：

输出一行，包含一个整数，表示大象至少要喝水的桶数。

样例输入：

23 11

样例输出：

3

# 【上机练习】

## 5.计算线段长度【1.3编程基础之算术表达式与顺序执行16】

已知线段的两个端点的坐标 $A(X_a, Y_a)$ ， $B(X_b, Y_b)$ ，求线段AB的长度，保留到小数点后3位。

输入：

第一行是两个实数 $X_a$ ， $Y_a$ ，即A的坐标。

第二行是两个实数 $X_b$ ， $Y_b$ ，即B的坐标。

输入中所有实数的绝对值均不超过10000。

输出：

一个实数，即线段AB的长度，保留到小数点后3位。

样例输入：

1 1

2 2

样例输出：

1.414

# 【上机练习】

## 6.计算三角形面积【1.3编程基础之算术表达式与顺序执行18】

平面上有一个三角形，它的三个顶点坐标分别为 $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$ ，那么请问这个三角形的面积是多少，精确到小数点后两位。

输入：

输入仅一行，包括6个单精度浮点数，分别对应 $x_1, y_1, x_2, y_2, x_3, y_3$ 。

输出：

输出也是一行，输出三角形的面积，精确到小数点后两位。

样例输入：

0 0 4 0 0 3

样例输出：

6.00

# 【上机练习】

## 7.A\*B问题【1.3编程基础之算术表达式与顺序执行19】

输入两个正整数A和B，求A\*B的值。注意乘积的范围和数据类型的选择。

输入：

一行，包含两个正整数A和B，中间用单个空格隔开。 $1 \leq A, B \leq 50000$ 。

输出：

一个整数，即A\*B的值。

样例输入：

3 4

样例输出：

12

# 【上机练习】

## 8.计算2的幂【1.3编程基础之算术表达式与顺序执行20】

给定非负整数 $n$ ，求 $2^n$ 的值，即2的 $n$ 次方。

输入：

一个整数 $n$ 。  $0 \leq n < 31$ 。

输出：

一个整数，即2的 $n$ 次方。

样例输入：

3

样例输出：

8



# 【上机练习】

## 9.苹果和虫子【1.3编程基础之算术表达式与顺序执行15】

你买了一箱 $n$ 个苹果，很不幸的是买完时箱子里混进了一条虫子。虫子每 $x$ 小时能吃掉一个苹果，假设虫子在吃完一个苹果之前不会吃另一个，那么经过 $y$ 小时你还有多少个完整的苹果？

输入：

输入仅一行，包括 $n$ ， $x$ 和 $y$ （均为整数）。

输出：

输出也仅一行，剩下的苹果个数

样例输入：

10 4 9

样例输出：

7

# 【上机练习】

## 10.求三角形面积【1.3编程基础之算术表达式与顺序执行17】

给定三条线段的长度，判断这三条线段是否能够构成三角形。如果能够构成，则计算其面积。

输入：

输入只有一行，包含三个浮点数，分别表示三角形三边的边长，数与数之间以一个空格分开。

输出：

对于可以构成三角形的情况，输出三角形的面积，保留4位小数。

对于不能构成三角形的情况，输出 “Data Error” 。

样例输入：

3 4 5

样例输出：

6.0000