第三章 程序的控制结构

第一节 概述

第二节 if选择结构

第三节 switch语句

第一节 概述

- 程序由若干条语句组成，各语句按照顺序一条一条地执行，这种顺序结构是简洁的。但在现实世界中，在解决问题的过程中，不可避免地遇到需要进行选择、或需要循环工作的情况。这时，程序执行的顺序需要发生变化，而非从前向后逐一执行。因此，程序中除了顺序结构以外，通常还有选择结构、循环结构以及转移机制。
- C++为了支持这些控制结构，提供了丰富、灵活的控制语句。从结构化程序设计的观点看，所有程序都可用3种控制结构即顺序结构、选择结构、和循环结构实现。C++在默认的情况下采取顺序结构，除非特别指明，计算机总是按语句顺序一条一条地执行。为使程序更清晰、更易调试与修改，并且不容易出错，结构化编程要尽量少用或不用goto等跳转语句。

- 选择类语句包括if语句和switch语句，用它们来解决实际应用中按不同的情况进行不同处理的问题。如根据学生的成绩，对学生做出不同的等第评价。if选择结构称为单分支选择结构，选择或忽略一个分支的操作。if/else选择结构称为双分支选择结构，在两个不同分支中选择。switch选择结构称为多分支（或多项）选择结构，以多种不同的情况选择多个不同的操作。
- 循环类语句包括for循环语句、while循环语句和do循环语句三种，用它们来解决实际应用中需要重复处理的问题。如当统计全班同学总分时，就需要重复地做加法，依次把每个人的分数累加起来。

- if、else、switch、while、do和for等都是C++关键字。这些关键字是该语言保留的，用于实现C++控制结构的不同特性。关键字不能作为变量名等一些标识符。注意，将关键字while的拼写变为“While”是个语法错误，因为C++是区分大小写的语言。while、if和else等所有C++保留关键字只能包含小写字母。

第二节 if选择结构

C++提供三种选择结构，即if选择结构、if-else选择结构和switch选择结构。

一、if语句（单分支结构）

格式1：

```
if（条件表达式）
  语句1;
```

功能：如果条件表达式的值为真，即条件成立，语句1将被执行。否则，语句1将被忽略（不被执行），程序将按顺序从整个选择结构之后的下一条语句继续执行。执行流程如图3-1所示

说明：格式中的“条件表达式”必须用圆括号括起来。

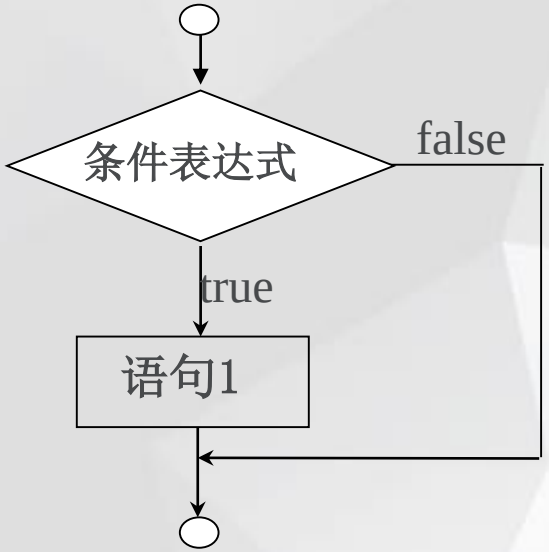

图3-1

程序设计风格提示：从语法上来讲，整个if语句可以写在一行。如果条件表达式和语句都非常简单，那么整个语句可以写在一行；否则，最好在条件表达式后换行，而且语句部分要相对if缩进两格。

例3.1 读入一个整数a，如果a为偶数在屏幕上输出yes

```
#include<iostream>
using namespace std;
int main( )
{
   int a;
   cin>>a;
   if (a%2==0)
     cout<<"yes";
   return 0;
}
```

注意：关系运算符==用来表达该符号的左右两边是否相等，不要写成赋值号=。

试一试

若题目改为“读入一个整数a，如果a为奇数在屏幕上输出no”该如何修改程序？

例3.2 读入一个数，若这个数大于1并且小于100，则输出yes

```
# include<iostream>
using namespace std;
int main ( )
{
   int a;
   cin>>a;
   if ((a>1)&&(a<100))
      cout<<"yes";
   return 0;
}
```

注意：此程序中的条件表达式为(a>1)&&(a<100)，根据要求“条件表达式”必须用圆括号括起来，否则编译会出错。

格式2：

```
if (条件表达式)
{
  语句1;
  语句2;
  ……
 }
```

若条件成立时，要执行的操作由多个句子构成，我们必须把这些句子括在一对花括号{ }内，我们称这种形式为语句块或复合语句。

程序设计风格提示：书写语句块（也称为复合语句）时，左右花括号要对齐，组成语句块的各语句要相对花括号缩进一层并对齐。

例3.3 读入a，b，若a>b则交换a，b的值

```
#include<iostream>
using namespace std;
int main ( )
{
   float a,b,c;
   cin>>a>>b;
   if (a>b)
     {
      c=a; a=b; b=c;
      }
   cout<<"a="<<a<<" b="<<b;
   return 0;
}
```

二、if-else语句（双分支结构）

- if单分支选择结构只在条件为true时采取操作，条件为false时则忽略这个操作。利用if-else双分支选择结构则可以在条件为true时和条件为false时采取不同操作。
- 格式1：
- 　　功能：如果（条件表达式）的值为“真”，即条件成立，则执行语句1，执行完“语句1”后继续执行整个if - else语句的后继语句；如果（条件表达式）的值为“假”，即条件不成立，那么跳过语句1选择执行“语句2”，执行完语句2后继续执行整个if - else语句的后继语句；也就是说if - else语句总是根据（条件表达式）的结果，选择“语句1”和“语句2”中的一个执行，执行完以后，整个if - else就算执行完了。执行流程如图3-2所示

```
if（条件表达式)
    语句1;
else
    语句2;
```

程序设计风格提示：书写if－else语句时，if和else要对齐，而分支的语句部分要缩进两格。

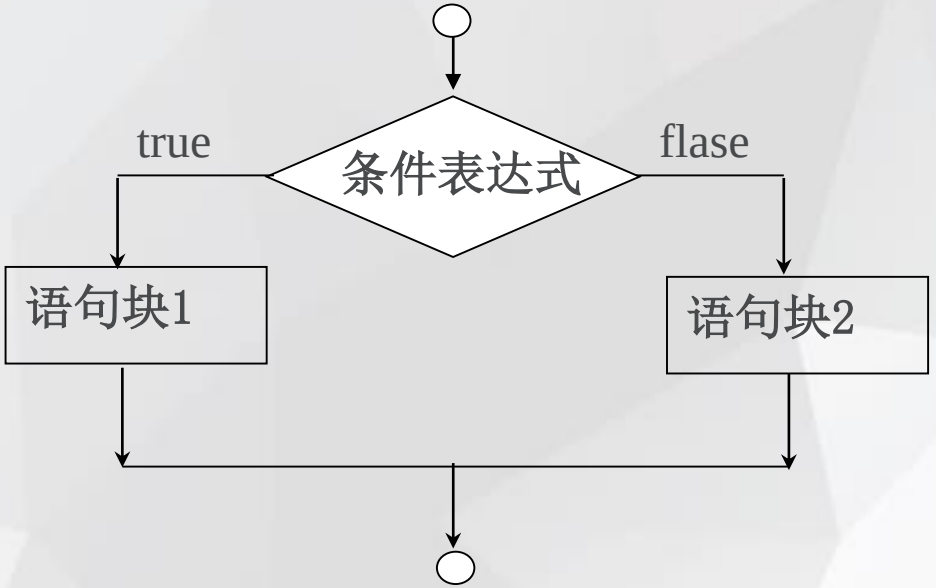

图3-2

例3.4 根据用户输入的成绩，判断是否通过了考试。

```
#include<iostream >
using namespace std;
int main()
{
    int c;
    cin >> c;
    if (c>=60)
       cout<<"pass! \n";
    else
       cout<<"sorry!\n";
    return 0;
 }
```

例3.5　输入温度t的值，判断是否适合晨练。(25<=t<=30，则适合晨练ok，否则不适合no)

```
#include<iostream>
using namespace std;
int main()
 {
      int t;
      cin >> t;
      if  ((t>=25) &&(t<=30))
         cout<<"ok!\n";
      else
         cout<<"no!\n";
      return 0;
  }
```

格式2：

```
if (条件表达式)
{ 语句1;
  语句2;
  ……
 }
else
{ 语句1;
语句2;
……
   }
```

若分支语句由多个句子构成，我们必须把这些句子括在一对花括号{ }内。

例3.6 乘坐飞机时，当乘客行李小于等于20公斤时，按每公斤1.68元收费，大于20公斤时，按每公斤1.98元收费，编程计算收费(保留2位小数)。

```
#include<cstdio>
using namespace std;
int main( )
{
   float  w, p;
   scanf("%f",&w);
   if (w<=20)
     printf("%.2f",w*1.68);
   else
     printf("%.2f",w*1.98);
   return 0;
}
```

if语句允许嵌套，即语句1和语句2还可以是if语句，当if语句嵌套时，约定else总是和最近的一个if语句配对。

例3.7 if (a>b)

if (b>c) y=a;

else y=c;

else部分否定的是条件b>c，即它与第二个if语句配对；若想让else部分与第一个if语句配对，则要引入一个复合语句，将上述语句写成如下形式：

```
if (a>b)
  {
    if (b>c) y=a;
  }
  else  y=c;
```

【上机练习】

1.判断数正负【1.4编程基础之逻辑表达式与条件分支01】

给定一个整数N，判断其正负。如果N>0，输出positive；如果N=0，输出zero；如果N<0，输出negative。

输入:

一个整数N(-10^9 <= N <= 10^9)

输出:

如果N > 0, 输出positive;

如果N = 0, 输出zero;

如果N < 0, 输出negative

样例输入:

1

样例输出:

positive

【上机练习】

2.输出绝对值【1.4编程基础之逻辑表达式与条件分支02】

输入一个浮点数，输出这个浮点数的绝对值，保留到小数点后两位。

输入:

输入一个浮点数，其绝对值不超过10000。

输出:

输出这个浮点数的绝对值，保留到小数点后两位。

样例输入:

-3.14

样例输出:

3.14

【上机练习】

3.奇偶数判断【1.4编程基础之逻辑表达式与条件分支03】

给定一个整数，判断该数是奇数还是偶数。如果n是奇数，输出odd；如果n是偶数，输出even。

输入:

输入仅一行，一个大于零的正整数n。

输出:

输出仅一行，如果n是奇数，输出odd；如果n是偶数，输出even。

样例输入:

5

样例输出:

odd

【上机练习】

4.奇偶ASCII值判断【1.4编程基础之逻辑表达式与条件分支04】

任意输入一个字符，判断其ASCII是否是奇数，若是，输出YES，否则，输出NO。例如，字符A的ASCII值是65，则输出YES，若输入字符B(ASCII值是66)，则输出NO。

输入:

输入一个字符。

输出:

如果其ASCII值为奇数，则输出YES，否则，输出NO。

样例输入:

A

样例输出:

YES

【上机练习】

5.整数大小比较【1.4编程基础之逻辑表达式与条件分支05】

输入两个整数，比较它们的大小。若x>y，输出>；若x=y，输出=；若x<y，输出<。

输入:

一行，包含两个整数x和y，中间用单个空格隔开。0<=x<2^32，-2^31<=y<2^31。

输出:

一个字符。若x > y，输出 > ；若x = y，输出 = ；若x < y，输出 < ；

样例输入:

1000 100

样例输出:

>

【上机练习】

6.判断是否为两位数【1.4编程基础之逻辑表达式与条件分支06】

判断一个正整数是否是两位数(即大于等于10且小于等于99)。若该正整数是两位数，输出1，否则输出0。

输入:

一个正整数，不超过1000。

输出:

一行。若该正整数是两位数，输出1，否则输出0。

样例输入:

54

样例输出:

1

【上机练习】

7.收集瓶盖赢大奖【1.4编程基础之逻辑表达式与条件分支07】

某饮料公司最近推出了一个“收集瓶盖赢大奖”的活动：如果你拥有10个印有“幸运”、或20个印有“鼓励”的瓶盖，就可以兑换一个神秘大奖。现分别给出你拥有的印有“幸运”和“鼓励”的瓶盖数，判断是否可以去兑换大奖。若可以兑换大奖，输出1，否则输出0。

输入:

一行，包含两个整数，分别是印有“幸运”和“鼓励”的瓶盖数，用一个空格隔开。

输出:

一行。若可以兑换大奖，输出1，否则输出0。

样例输入:

11 19

样例输出:

1

【上机练习】

8.判断一个数能否同时被3和5整除【1.4编程基础之逻辑表达式与条件分支08】

判断一个数n 能否同时被3和5整除，如果能同时被3和5整除输出YES，否则输出NO。

输入:

输入一行，包含一个整数n。（ -1,000,000 < n < 1,000,000)

输出:

输出一行，如果能同时被3和5整除输出YES，否则输出NO。

样例输入:

15

样例输出:

YES

【上机练习】

9.判断能否被3，5，7整除【1.4编程基础之逻辑表达式与条件分支09】

给定一个整数，判断它能否被3，5，7整除，并输出以下信息：

1、能同时被3，5，7整除（直接输出3 5 7，每个数中间一个空格）；

2、只能被其中两个数整除（输出两个数，小的在前，大的在后。例如：3 5或者 3 7或者 5 7，中间用空格分隔）；

3、只能被其中一个数整除（输出这个除数）；

4、不能被任何数整除，输出小写字符‘n’，不包括单引号。

输入:

输入一行，包括一个整数。

输出:

输出一行，按照描述要求给出整数被3，5，7整除的情况。

样例输入:

105

样例输出:

3 5 7

【上机练习】

10．有一门课不及格的学生【1.4编程基础之逻辑表达式与条件分支10】

给出一名学生的语文和数学成绩，判断他是否恰好有一门课不及格(成绩小于60分)。若该生恰好有一门课不及格，输出1；否则输出0。

输入:

一行，包含两个在0到100之间的整数，分别是该生的语文成绩和数学成绩。

输出:

若该生恰好有一门课不及格，输出1；否则输出0。

样例输入:

50 80

样例输出:

1

第三节　switch语句

应用条件语句可以很方便地使程序实现分支，但是出现分支比较多的时候，虽然可以用嵌套的if语句来解决，但是程序结构会显得复杂，其至凌乱。为方便实现多情况选择，C++提供了一种switch开关语句。

1.语句格式：

```
switch（表达式）
    {
     case 常量表达式1：
          语句序列1；
          break;
     case 常量表达式2：
          语句序列2；
          break;
     ……
     case 常量表达式n：
          语句序列n；
          break;
     default:
          语句序列n+1；
    }
```

该语句中可以使用一次或多次case标号，但只能使用一次default标号，或者省略整个default部分；多个case标号也允许使用在同一个语句序列的前面；每个语句标号有保留字case和后面的常量表达式及冒号组成，每个常量表达式通常为字面常量，如常数或字符。

2.语句执行过程

switch语句执行过程分为以下3步描述。

(1)计算出switch后面圆括号内表达式的值，假定为M，若它不是整型，系统将自动舍去其小数部分，只取其整数部分作为结果值。

(2)依次计算出每个case后常量表达式的值，假定它们为M1、M2、…，同样若它们的值不是整型，则自动转换为整型。

(3)让M依次同M1、M2、…进行比较，一旦遇到M与某个值相等，则就从对应标号的语句开始执行；在碰不到相等的情况下，若存在default子句，则就执行其冒号后面的语句序列，否则不执行任何操作；当执行到复合语句最后的右花括号时就结束整个switch语句的执行。

在实际使用switch语句时，通常要求当执行完某个case后的一组语句序列后，就结束整个语句的执行，而不让它继续执行下一个case语句后面的语句序列，为此，可通过使用break语句来实现。该语句只有保留字break，而没有其它任何成分。它是一条跳转语句，在switch中执行到它时，将结束该switch语句，系统接着向下执行其它语句。

在使用switch语句时，还应注意以下几点：

1. case语句后的各常量表达式的值不能相同，否则会出现错误码。

2.每个case或default后，可以包含多条语句，不需要使用“{”和“}”括起来。

3.各case和default子句的先后顺序可以变动，这不会影响程序执行结果。

4. default子句可以省略，default后面的语句末尾可以不必写break。

程序设计风格提示：写switch语句时，switch（表达式）单独一行，各case分支和default分支要缩进两格并对齐，分支处理语句要相对再缩进两格，以体现不同层次的结构。

3.语句格式举例

(1)左右两边的书写格式是等价的

```
switch(a)
{
    case 1:x++;break;
    case 2:y++;break;
    case 3:z++;break;
    default:cout<<"error";
  }
```

```
switch(a)
{
   case 1:
          x++;
          break;
   case 2:
          y++;
          break;
   case 3:
          z++;
          break;
   default:
          cout<<"error";
 }
```

(2) switch(ch)

```
{
case ‘a’:
case ‘A‘:
        d1=(x+y)/2;
        d2=x*y-2;
        break;
case ‘b’:
case ‘B’:
        d1=(a+b)/2;
        d2=a*b-2;
        break;
default:
        cout<<"input error!";
}
```

说明： 1.每个case后面的语句可以写在冒号后的同一行或换到新行写。

2.<语句序列1>…<语句序列n+1>都是一组语句，有时可为空，如(2)。

例3.8 根据从键盘上输入的表示星期几的数字，对应输出它的英文名称。

```
#include<iostream>
using namespace std;
int main()
{
    int weekday;
    cin>>weekday;
    switch(weekday)
    {
        case 1:cout<<"Monday"<<endl; break;
        case 2: cout<<"Tuesday"<<endl; break;
        case 3: cout<<"Wednesday"<<endl; break;
        case 4: cout<<"Thursday"<<endl; break;
        case 5: cout<<"Friday"<<endl; break;
        case 6: cout<<"Saturday"<<endl; break;
        case 7: cout<<"Sunday"<<endl; break;
        default:cout<<"input  error!";
    }
   return 0;
}
```

例3.9 判断2006年每个月份的天数。

【分析】 程序分为：输入月份,计算该月的天数,输出天数。

程序如下：

```
#include<iostream>
using namespace std;
int main()
{
    int month,day;
    cin>>month;
    switch(month)
    {
       case 2:day=28;break;
       case 4:day=30;break;
       case 6:day=30;break;
       case 9:day=30;break;
       case 11:day=30;break;
       default:day=31;
    }
    cout<<day<<endl;
    return 0;
}
```

例3.10　期末来临了，班长小Q决定将剩余班费X元钱，用于购买若干支钢笔奖励给一些学习好、表现好的同学。已知商店里有三种钢笔，它们的单价为6元、5元和4元。小Q想买尽量多的笔（鼓励尽量多的同学），同时他又不想有剩余钱。请您编一程序，帮小Q制订出一种买笔的方案。

【分析】对于以上的实际问题，要买尽量多的笔，易知都买4元的笔肯定可以买最多支笔。因此最多可买的笔为x div 4支。由于小q要把钱用完，故我们可以按以下方法将钱用完：

若买完x div 4支4元钱的笔，还剩1元，则4元钱的笔少买1支，换成一支5元笔即可；若买完x div 4支4元钱的笔，还剩2元，则4元钱的笔少买1支，换成一支6元笔即可；若买完x div 4支4元钱的笔，还剩3元，则4元钱的笔少买2支，换成一支5元笔和一支6元笔即可。

从以上对买笔方案的调整，可以看出笔的数目都是x div 4，因此该方案的确为最优方案。

程序如下：

```
#include<iostream>
using namespace std;
int main()
{
    int a,b,c,x,y;
                    //a,b,c分别表示在买笔方案中，6元、5元和4元钱笔的数目
                        //x，y分别表示剩余班费和买完最多的4元笔后剩的钱
    cin>>x;             //输入x
    c=x/4;              //4元笔最多买的数目
    y=x%4;              //求买完c支4元笔后剩余的钱数y
    switch (y)          //判断购买方案
    {
      case 0: a=0; b=0;       break;
      case 1: a=0; b=1; c--;  break;
      case 2: a=1; b=0; c--;  break;
      case 3: a=1; b=1; c-=2; break;
    }
    cout<<a<<' '<<b<<' '<<c<<endl;     //三个数间以空格隔开
    return 0;
}
```

【上机练习】

1.晶晶赴约会【1.4编程基础之逻辑表达式与条件分支11】

晶晶的朋友贝贝约晶晶下周一起去看展览，但晶晶每周的1、3、5有课必须上课，请帮晶晶判断她能否接受贝贝的邀请，如果能输出YES；如果不能则输出NO。注意YES和NO都是大写字母！

输入:

输入有一行，贝贝邀请晶晶去看展览的日期，用数字1到7表示从星期一到星期日。

输出:

输出有一行，如果晶晶可以接受贝贝的邀请，输出YES，否则，输出NO。注意YES和NO都是大写字母！

样例输入:

2

样例输出:

YES

【上机练习】

2.骑车与走路【1.4编程基础之逻辑表达式与条件分支12】

在清华校园里，没有自行车，上课办事会很不方便。但实际上。并非去办任何事情都是骑车快,因为骑车总要找车、开锁、停车、锁车等,这要耽误一些时间。假设找到自行车，开锁并车上自行车的时间为27秒；停车锁车的时间为23秒；步行每秒行走1.2米，骑车每秒行走3.0米。请判断走不同的距离去办事,是骑车快还是走路快。如果骑车快，输出一行"Bike"；如果走路快，输出一行"Walk"；如果一样快，输出一行"All"。

输入:

输入一行，包含一个整数，表示一次办事要行走的距离,单位为米。

输出:

输出一行,如果骑车快,输出一行"Bike";如果走路快,输出一行"Walk";如果一样快,输出一行"All"。

样例输入:

120

样例输出:

Bike

【上机练习】

3.分段函数【1.4编程基础之逻辑表达式与条件分支13】

编写程序，计算下列分段函数y=f(x)的值。结果保留到小数点后三位。

y=-x+2.5;　　0<=x<5

y=2-1.5(x-3)(x-3);　5<=x<10

y=x/2-1.5;　　10<=x<20

输入:

一个浮点数N，0 <= N < 20。

输出:

输出N对应的分段函数值：f(N)。结果保留到小数点后三位。

样例输入:

1.0

样例输出:

1.500

【上机练习】

4.计算邮资【1.4编程基础之逻辑表达式与条件分支14】

根据邮件的重量和用户是否选择加急计算邮费。计算规则：重量在1000克以内(包括1000克)，基本费8元。超过1000克的部分，每500克加收超重费4元，不足500克部分按500克计算；如果用户选择加急，多收5元。

输入:

输入一行，包含整数和一个字符，以一个空格分开，分别表示重量（单位为克）和是否加急。如果字符是y，说明选择加急；如果字符是n，说明不加急。

输出:

输出一行，包含一个整数，表示邮费。

样例输入:

1200 y

样例输出:

17

【上机练习】

5.最大数输出【1.4编程基础之逻辑表达式与条件分支15】

输入三个整数，数与数之间以一个空格分开。 输出一个整数，即最大的整数。

输入:

输入为一行，包含三个整数，数与数之间以一个空格分开。

输出:

输出一行，包含一个整数，即最大的整数。

样例输入:

10 20 56

样例输出:

56

【上机练习】

6.三角形判断【1.4编程基础之逻辑表达式与条件分支16】

给定三个正整数，分别表示三条线段的长度，判断这三条线段能否构成一个三角形。如果能构成三角形，则输出“yes”，否则输出“no”。

输入:

输入共一行，包含三个正整数，分别表示三条线段的长度，数与数之间以一个空格分开。

输出:

如果能构成三角形，则输出“yes”，否则输出“no”。

样例输入:

3 4 5

样例输出:

yes

【上机练习】

7.判断闰年【1.4编程基础之逻辑表达式与条件分支17】

判断某年是否是闰年。如果公元a年是闰年输出Y，否则输出N。

输入:

输入只有一行，包含一个整数a(0 < a < 3000)。

输出:

一行，如果公元a年是闰年输出Y，否则输出N。

样例输入:

2006

样例输出:

N

【上机练习】

8.点和正方形的关系【1.4编程基础之逻辑表达式与条件分支18】

有一个正方形，四个角的坐标（x,y)分别是(1，-1)，(1，1)，(-1，-1)，(-1，1)，x是横轴，y是纵轴。写一个程序，判断一个给定的点是否在这个正方形内(包括正方形边界)。如果点在正方形内，则输出yes，否则输出no。

输入:

输入一行，包括两个整数x、y，以一个空格分开，表示坐标(x,y)。

输出:

输出一行，如果点在正方形内，则输出yes，否则输出no。

样例输入:

1 1

样例输出:

yes

【上机练习】

9.简单计算器【1.4编程基础之逻辑表达式与条件分支19】

一个最简单的计算器，支持+，-，*，/ 四种运算。仅需考虑输入输出为整数的情况，数据和运算结果不会超过int表示的范围。然而：

1. 如果出现除数为0的情况，则输出：Divided by zero!

2．如果出现无效的操作符(即不为 +，-，*，/ 之一），则输出：Invalid operator!

输入:

输入只有一行，共有三个参数，其中第1、2个参数为整数，第3个参数为操作符（+,-,*,/）。

输出:

输出只有一行，一个整数，为运算结果。然而：

1.如果出现除数为0的情况，则输出：Divided by zero!

2.如果出现无效的操作符(即不为 +，-，*，/ 之一），则输出：Invalid operator!

样例输入:

1 2 +

样例输出

3

【上机练习】

10.求一元二次方程【1.4编程基础之逻辑表达式与条件分支20】

利用公式x1 = (-b + sqrt(b*b-4*a*c))/(2*a)，x2 = (-b - sqrt(b*b-4*a*c))/(2*a)，求一元二次方程ax2+ bx + c =0的根，其中a不等于0。结果要求精确到小数点后5位。

输入:

输入一行，包含三个浮点数a，b，c（它们之间以一个空格分开），分别表示方程ax2 + bx + c =0的系数。

输出:

输出一行，表示方程的解。

若两个实根相等，则输出形式为：x1=x2=...。

若两个实根不等，则输出形式为：x1=...;x2 = ...，其中x1若是两个虚根，则输出：x1=实部+虚部i; x2=实部-虚部i，其中x1,x2满足以下两个条件中的一个：

1.x1的实部大于x2的实部

2.x1的实部等于x2的实部且x1的虚部大于等于x2的虚部

所有实数部分要求精确到小数点后5位，数字、符号之间没有空格。

样例输入:

1.0 2.0 8.0

样例输出:

x1=-1.00000+2.64575i;x2=-1.00000-2.64575i