//import java.util.Scanner;

public class A {

//main方法，表示程序的入口

public static void main(String[] args) {

System.out.println("Hello World!");

/\*制表符的使用\t 在打印的时候把前面的字符串长度补齐到8，或者8的整数倍 。最少补一个空格，最多8个

System.out.println("name" + '\t' + "age");

System.out.println("tom" + '\t' + "age");

\*/

/\*

变量

定义格式： 数据类型 变量名=数据值；

数据类型：限定了变量能存储数据的类型

=：表示赋值右边赋给左边

int a=10;

System.out.println(a);

//在一条语句中可以定义多个变量

int d=100, e=200, f=300

\*/

/\*

练习：一开始车上没有乘客。int count=0;

第一站上去一位乘客。count=count+1;

第二站上去两位乘客，下去一位乘客。count=count+2-1;

第三站上去两位乘客，下来一位乘客。count=count+2-1;

第四站下来一位。count=count-1;

第五站上去一位。count=count+1;

请问到了终点，车上一共有多少人。

System.out.printlm(count);

\*/

/\*

数据类型

基本数据类型：整数类型byte(-128,+127) short int long、浮点数类型float double、字符类型char、布尔类型boolean

如果定义long类型变量，在数据值后面加L

如果定义float类型变量，在数据值后面加F

取值范围double>float>long>int>short>byte

\*/

/\*

Java帮我们写好一个类叫Scanner,这个类就可以接收键盘输入的数字

import java.util.Scanner;//写在类的上面 （导包）

Scanner sc=new Scanner(System.in); （创建对象）

int i=sc.nextInt();//接收数据，赋值 （接收数据）

System.out.println(i);

\*/

/\*

练习：键盘录入两个整数，求出他们的和并打印出来

import iava.util.Scanner;

public static ScannerTest(String[] args)

{

Scanner sc=new Scanner(System.in);

int number1=sc.nextInt();

int number2=sc.nextInt();

System.out.println(number1+number2);

\*/

/\*

除法 System.out.println(10/2);

System.out.println(10/3);

整数参与计算，结果只能是整数

取模，取余，实际上也是做除法运算，只不过得到余数而已%

System.out.println(10%3);

问题：键盘录入一个三位数，将其拆分为个位，十位，百位后，打印出来

公式：个位 数值%10

十位 数值/10%10

百位 数值/100%10

千位 数值/1000%10...

Scanner sc=new Scanner(System.in);

int number=sc.nextInt();

int ge=number%10;

int shi=number/10%10;

int bai=number/100%10;

System.out.println(ge);

System.out.println(shi);

System.out.println(bai);

\*/

/\*

数字进行运算时，数据类型不一样不能运算，需要转成一样的，才能运算

取值范围：byte（取值范围最小）<short<int<long<float<double(取值范围最大）

类型转换的分类：（数据类型不一样，不能进行计算，需要转成一样的才可以进行计算

1.隐式转换（自动类型提升）：取值范围小的数值转换为 取值范围大的数值

取值范围小的，和取值范围大的进行运算，小的会先提升大的，在进行运算

byte short char 三种类型的数据在运算的时候，都会直接先提升为int，然后再进行运算

int a=10; byte a=10;

double b=12.3; byte b=20;

c=a+b; c为double类型 c=a+b; c为int类型

2.强制转换：取值范围大的数值转换为取值范围小的数值

格式：目标数据类型 变量名=（目标数据类型）被强转的数据；

byte b1=10; 若要使result转为byte类型 byte b1=10;

byte b2=20; byte b2=20;

int result=b1+b2; byte result=(byte)(b1+b2);

System.out.println(result); System.out.println(result);

\*/

/\*

字符串的+操作

当+操作中出现字符串时，这个+是字符串的连接符，而不是运算符，会在前后的数据进行拼接，并产生一个新的字符串

连续进行+操作时，从左到右逐个进行

"123"+123 246

1+99+"年黑马” 100年黑马

3.7+"abc" 3.7abc

"abc"+true abctrue

'中'+"abc"+true 中abctrue

1+2+"abc"+2+1 3abc21

\*/

/\*

字符+操作

字符+字符 字符+数字时，会把字符通过ASCII码表查询到对应的数字在进行运算

1+'a' 98

'a'+"abc" "aabc"

\*/

/\*

自增自减运算符

++ 加 变量的值加1

-- 减 变量的值减1

++和--无论是放在变量的前面还是后面，答案读写一行结果是一样的

int a=10;

a++;

System.out.println(a); 11

++a;

System.out.println(a); 12

a--;

System.out.println(a); 11

--a;

System.out.println(a); 10

int a=10; int a=10;

int b=a++; 先用后加b为10 a为11 int b=++a;先加后用b为11 a为10

练习： int x=10; x:10

int y=x++; y:10 x:11

int z=++x; z:11+1 x:11+1

System.out.println("x:" +x); 12

System.out.println("y:" +y); 10

System.out.println("z:" +z); 12

\*/

/\*

赋值运算符

+= -= /= %= \*= 底层都隐藏了一个强制类型转换

= 赋值 int a=10,将10赋值给变量a

+= 加后赋值 a+=b,将a+b的值给a a=a+b;

-= 减后附值 a-=b,将a-b的值给a a=a-b;

\*= 乘后赋值 a\*=b,将a乘b的值给a a=a\*b;

/= 除后附值 a/=b,将a除b的商给a a=a/b;

%= 取余后赋值 a%=b,将a除b的余数给a a=a%b;

\*/

/\*

关系运算符

== a==b,判断a和b是否相等，成立为true,不成立为false

！= a！=b,判断a和b是否不相等，成立为true,不成立为false

> a>b,判断a是否大于b,成立为true,不成立为false

>= a>=b,判断a是否大于等于b,成立为true,不成立为false

< a<b,判断a是否大于b,成立为true,不成立为false

<= a<=b,判断a是否小于等于b,成立为true,不成立为false

关系运算符的结果都是boolean类型，要么是true,要么是false

练习：您和您的约会对象在餐厅里面正在约会。手动录入两个证书，表示你和你的约会对象衣服的时髦度。

如果你的时髦度大于你对象的时髦度，相亲就成功，输出为true。否则为false。

Scanner sc=new Scanner(System.in);

System.out.println("请输入我们自己衣服的时髦度");

int number1=sc.nextInt();

System.out.println("请输入相亲对象衣服的时髦度");

int number2=sc.nextInt();

boolean result=number1>number2;

System.out.println("result");

\*/

/\*

逻辑运算符

& 逻辑与（且） 并且，两边都是真，结果才为真

| 逻辑或 或者，两边都为假，结果才是假

^ 逻辑异或 相同为false,不同为true

! 逻辑非 取反

System.out.println(true&true); true

System.out.println(false&false); true

System.out.println(true&false); false

System.out.println(false&true); false

System.out.println(true|true); true

System.out.println(false|false); false

System.out.println(true|false); true

System.out.println(false|true); true

System.out.println(true^true); false

System.out.println(false^false); false

System.out.println(true^false); true

System.out.println(false^true); true

System.out.println(!true); false

System.out.println(!false); true

\*/

/\*

短路逻辑运算符

&& 短路与 结果和&相同，但是有短路效果

|| 短路或 结果和|相同，但是有短路效果

System.out.println(true&&true); true

System.out.println(false&&false); false

System.out.println(true&&false); false

System.out.println(false&&true); false

System.out.println(true&&true); true

System.out.println(false&&false); false

System.out.println(true&&false); true

System.out.println(false&&true); true

例子; int a=10;

int b=10;

boolean result=++a<5&&++b<5; ++a<5 false -> false&&++b<5 ->false(++b<5直接不判断了）

System.out.println(result); false

System.out.println(a); 11

System.out.println(b); 10

& | 无论左边是false还是true，右边都要执行

&& || 如果坐便能确定整个表达式的结果，右边不执行

练习：数字6是一个伟大的数字，键盘录入两个整数。如果其中一个为6，最终结果输出true。

如果他们的和为6的倍数。最终结果输出true.其他情况都要输出false.

分析 ：a==6 b==6 (a+b)%6==0

Scanner sc=new Scanner(System.in);

System.out.println("请输入第一个整数");

int a=sc.nextInt();

System.out.println("请输入第一个整数");

int b=sc.nextInt();

boolean result= a==6||b==6||(a+b)%6==0

System.out.println(result);

\*/

/\*

三元运算符（三元表达式）

格式： 关系表达式？表达式1：表达式2；

关系表达式为true，则表达式1是三元运算符的结果。

关系表达式为false，则表达式2是三元运算符的结果

使用时不能直接写，需将结果赋值给一个变量或者结果直接打印

例子：使用三元运算符，过去两个数的较大值

int number1=10;

int number2=20;

int max=number1>number2?number1:number2;

System.out.println(max);

练习：动物园里面有两只老虎，体重分别为通过键盘录入获得。请用程序实现判断两只老虎的体重是否相同。

Scanner sc=new Scanner(System.in);

int weight1=sc.nextInt();

int weight2=sc.nextInt();

System.out.println(weight1>weight2?"相同":"不同");

练习：一个寺庙里住着三个和尚，一支他们的身高分别为150cm,210cm,165cm,请用程序获取这三个和尚的最高身高

分析：先比较前两个得出的结果与第三个进行比较

int height1=150;

int height2=210;

int height3=165;

int max1=height1>height2?height1:height2;

int max2=max1>height3?max1:height3;

System.out.println(max2);

\*/

/\*

if语句

if(关系表达式){

语句体

}

如果对一个boolean的变量进行判断，不要用==号，直接把变量写在小括号中

boolean flag=false;

if(flag){

System.out.println("flag值为true");

if(关系表达式){

语句体1

}else{

语句体2

}

练习：假设某应援卖了100张票，票的需要为1~100.其中技术票号坐左边，偶数票号坐右边。键盘录入一个整数表示电影票的票号

根据不同情况给出不同提示。

Scanner sc=new Scanner(System.in);

System.out.println("请输入票号");

int ticket=sc.nextInt();

if(ticket>0&&ticket<=100){

if(ticket%2==1){

System.out.println("左边");

}else{

System.out.println("右边");

}

}

if(关系表达式1){

语句体1；

}else if(关系表达式2){

语句体2；

}

......

else{

语句体；

}

\*/

/\*

首先计算表达式的值。一次和case 后面的值进行比较，如果有对应的值，就会执行相应语句，在执行过程中，遇到break就会结束。

如果所有的case后面的值 和表达式的值都不匹配，就会执行default里面的语句体，然后结束整个switch语句。

表达式：（将要匹配的值）取值为byte,short,int char.JDK5以后可以是枚举，JDK7以后可以是String.

case:后面跟的是要和表达式进行比较的值（被匹配的值）。

break：表示中断，结束的意思，用来结束switch语句。

default:表示所有情况都不匹配的时候，就执行该处的内容，和if语句的else相似。

case后面的值只能是自变量，不能是变量

case给出的值不允许重复

switch(){

case 值1：

语句体1

break；

case 值2：

语句体2；

break;

...

default:

语句体n+1;

break;

}

\*/

/\*

String noodles="兰州拉面“；

switch(noodles){

case"兰州拉面"：

System.out.println("吃兰州拉面");

break；

case"武汉热干面"：

System.out.println("吃武汉热干面");

break；

case"北京炸酱面"：

System.out.println("吃北京炸酱面");

break；

case"陕西油泼面"：

System.out.println("吃陕西油泼面");

break；

default:

System.out.println("吃方便面");

break；

}

\*/

/\*

int number=1;

switch(number){

case 1->System.out.println("一");

case 2->System.out.println("二");

case 3->System.out.println("三");

default->System.out.println("没有这种选项");

}

switch：把有限个数据一一列举出来，让我们任选其一

if的第三种格式：一般用于对范围的判断

\*/

/\*

练习：键盘录入星期数，输出工作日，休息日

一。 Scanner sc=new Scanner(System.in);

int week=sc.nextInt();

switch(week){

case 1,2,3,4,5:

System.out.println("工作日");

break；

case 6,7:

System.out.println("休息日");

break；

default：

System.out.println("没有这个星期");

break;

}

二。Scanner sc=new Scanner(System.in);

int week=sc.nextInt();

switch(week){

case 1,2,3,4,5->System.out.println("工作日");

case 6,7->System.out.println("休息日");

default->System.out.println("没有这个星期");

}

三。Scanner sc=new Scanner(System.in);

int week=sc.nextInt();

switch(week){

case 1:

case 2:

case 3:

case 4:

case 5:

System.out.println("工作日");

break；

case 6:

case 7:

System.out.println("休息日");

break；

default：

System.out.println("没有这个星期");

break;

}

\*/

/\*

for循环

for(初始化语句：条件判断语句：条件控制语句){

循环体语句；

}

例如： for(int i=0;i<5;i++)

{

System.out.println("HelloWorld");

}

\*/

/\*

打印1-5

for(int i=1;i<=5;i++)

{

System.out.pringln(i);

}

打印5-1

for(int i=5;i>=1;i--)

{

System.out.pringln(i);

}

求1-5之间的和

int sum=0;

for(int i=1;i<=5;i++){

sum=sum+i;

}

System.out.println(sum);

求1-100之间的偶数和

int sum=0;

for(int i=1;i<=100;i++){

if(i%2==0)

{

sum=sum+i;

}

}

System.out.println(sum);

键盘录入两个数字，表示一个范围。统计这个范围中，既能被3整除，又能被5整除的数字有多少个

Scanner sc=new Scanner(System.in);

System.out.println("请录入一个数字表示范围开始：");

int start=sc.nextInt();

System.out.println("请录入一个数字表示范围结束：");

int end=sc.nextInt();

int count=0;

for(int i=start;i<=end;i++)

{

if(i%3==0&&i%5==0){

count++;

}

}

System.out.println(count);

\*/

/\*

while循环的执行流程

初始化语句

while(条件判断语句){

循环体语句；

条件控制语句；

}

循环下面其他语句

练习：打印1-100

int i=1;

while(i<=100)

{

System.out.println(i);

i++;

}

for和while的区别：for循环中，控制循环的变量，因为归属于for循环的语法结构中，在for循环结束后，

就不能再次被访问到了。while循环中，控制循环的变量，对于while循环来说不归属其语法结构中，在while

循环结束后，改变了还可以继续使用。

for循环中：直到循环次数或者循环的范围

while循环中：不知道循环的次数和范围，只知道循环的结束条件。

\*/

/\*

练习：世界最高山峰是珠穆朗玛峰8844.43米=8844430毫米，假如我有一张足够大的纸，他的厚度是0.1毫米，

请问，折叠多少次，可以折成珠穆朗玛峰的高度？

分析：每一次折叠纸张的厚度都是原来的两倍

double height=8844430;

double paper=0.1;

int count=0;

while(paper<height){

paper=paper\*2;

count++;

}

System.out.println(count);

\*/

/\*

练习：给你一个整数x。如果x是一个回文整数，打印true,否则打印false.

解释：回文数是指正序(从左向右）和倒序（从右向左）读都是一样的整数。

例如121是回文，而123不是

int x=1234;

int temp=1234; 定义一个临时变量用来记录x的变量 用作比较

int num=0;

while(x!=0){

int ge=x%10;

x=x/10;

num=num\*10+ge; 把当前获取到的数字拼接到最右边

}

System.out.println(num==temp);

\*/

/\*

练习：给定两个整数，被除数和除数（都是正数，且不超过int的范围）。将两数相除，

要求不使用乘法、除法和%运算符。得到商和余数。

int dividend=10;

int divisor=3;

int count=0;

while(dividend>=divisor){

dividend=dividend-divisor;

count++;

}

System.out.println(count); 商

System.out.println(dividend); 余数

\*/

/\*

do...while先执行后判断

do{

循环体语句；

条件控制语句；

}while(条件判断语句);

\*/

/\*

无限循环：循环一直停不下来

for(;;){

}

while(true){

}

do(

}while(true)

\*/

/\*

跳转控制语句

一、for(int i=1;i<=5;i++){

if(i==3){

continue;

}

System.out.println("小老虎在吃第"+i+"个包子");

}

二、for(int i=1;i<=5;i++){

System.out.println("小老虎在吃第"+i+"个包子");

if(i==3){

break;

}

}

\*/

/\*

练习：从任意一个数字开始报数，当你要报的数字是包含7或者是7的倍数时都要说：过。

使用程序在控制台打印1-100之间的满足逢七必过的数据.

分析：各位7、十位7、7的倍数

for(int i=1;i<=100;i++){

if(i%10==7||i/10%10==7||i%7==0){

System.out.println("过");

continue;

}

System.out.println(i);

}

\*/

/\*

练习：键盘录入一个大于等于2的整数x，计算并返回x的平方根。结果只保留整数部分，小数部分将被舍去。

分析：在代码当中从1开式循环，拿着数字的平方根原来的数字进行比较。如果小于的，那么继续往后判断

如果想等，那么大昂前数字就是平方根，如果大于的那么前一个数字九十平方根的整数部分

Scanner sc = new Scanner(System.in);

System.out.println("请输入一个整数");

int number =sc.nextInt();

for(int i=1;i<=number;i++){

if(i\*i==number){

System.out.println(i+"就是"+number+"的平方根");

}if(i\*i>number){

System.out.println((i-1)+"就是"+number+"的平方根");

break;

}

}

\*/

/\*

练习：键盘录入一个正整数x，判断该整数是否为一个质数。

质数：如果一个整数只能被1和本身整除，那么这个数就是质数，否则这个数叫做合数

//7=1\*7 质数

//8=1\*8 8=2\*4 合数

一、Scanner sc=new Scanner(System.in);

System.out.println("请输入一个正整数");

int number=sc.nextInt();

//写一个循环，从2开始判断，一直判断到number-1为止 看这个范围之内，有没有数字可以被number整除

for(int i=2;i<number;i++){

//i表示这个范围内的每一个数字

if(number%i==0){

System.out.println(number+"不是一个质数");

break;

}

System.out.println(number+"是质数");

}

二、boolean flag=true;

Scanner sc=new Scanner(System.in);

System.out.println("请输入一个正整数");

int number=sc.nextInt();

for(int i=2;i<number;i++){

if(number%i==0){

flag=false;

break;

}

}

if(flag){

System.out.println(number+"是一个质数");

}else{

System.out.println(number+"不是一个质数");

}

\*/

/\*

随机数

import java.util.Random;导包

Random r=new Random();创建对象

int number=r.nextInt(随机数的范围);生成随机数 10->0~9 100->0~99

练习：生成一个随机数，1~100

int number=r.nextInt(100)+1;

7~15

int number=r.nextInt(9)+7;

练习：Random r=new Random();

int number=r.nextInt(100)+1;

Scanner sc=new Scanner(System.in);

while(true){

System.out.println("请输入一个数字：");

int a=sc.nextInt();

if(number==a){

System.out.println("猜对啦");

}else if(a>number){

System.out.println("猜大啦");

}else{

System.out.println("猜小啦");}

}

注意： 生成随机数的代码不能写在循环里面

\*/

/\*

数组指的是一种容器，可以用来存储同种数据类型的多个值

数组容器在存储数据的时候，需要结合隐式转换考虑。建议容器类型和存储数据类型保存一致

一、 数据类型 [] 数组名

int [] array

二、 数据类型 数组名[]

int array[]

数组的静态初始化

初始化：就是内存中，为数组容器开辟空间，并将数据存入容器中的过程。

格式： 数据类型[] 数组名=new 数据类型[]{元素1，元素2，元素3...};

int[] array=new int[]{11,22,33};

int[] array={11,22,33};

索引：也叫做下标，角标。从0开始，逐个+1增长，连续不间断。

int [] arr={1,2,3,4,5};

int number=arr[0];

System.out.println(number);

\*/

}

}