System.out.println(“Hello”) :

System is package

out is instance of class/ref variable

println() is method

public class Main {

  public static void main(String[] args) {

    System.out.print("Hello World");

    System.out.println();

    System.out.println(10);

    System.out.println("5 \* 10"); // whatever write in double quote will be printed as it is

    System.out.println(5 \* 10);// whatever not in double quote compiler try to compute

    //System.out.println(Hello World);// If anything write without double quote compiler try to compute , if the staement don't have meaning in terms of compiler it will throw error

    /\*

    [CompilationError] Your code was terminated due to compilation error

    Main.java:11: error: ')' expected

    System.out.println(Hello World);// If anything write without double quote compiler try to compute , if the staement don't have meaning in terms of compiler it will throw error

    Main.java:11: error: illegal start of expression

    System.out.println(Hello World);// If anything write without double quote compiler try to compute , if the staement don't have meaning in terms of compiler it will throw error

    \*/

    // this single line comment

    /\*

    This

    is

    multiline

    comment

    \*/

    // Select all the line and do ctrl + / for comment

    //System.out.println("Ram "+" Shyam") // error as semicolon missing

    // [CompilationError] Your code was terminated due to compilation error

    // Main.java:31: error: ';' expected

    // System.out.println("Ram "+" Shyam") // error as semicolon missing

    System.out.println("Ram "+" Shyam");

    System.out.println("My age is "+30);

  }

}

OUTPUT:  
Hello World

10

5 \* 10

50

Ram Shyam

My age is 30

VARIABLES:

It’s a container to store a data.

DATATYPE:

1)Text

- “I am Awesome Shubham”

- “My age is 30”

Anything write inside doble quote is string

2)Numbers

-int (4 Bytes size, 32 bits, -231 <= int <= 231-1)= approx. -109 to 109

-long (8 Bytes size, 64 bits, -263 <= long <= 263-1) = approx. -1018 to 1018

Integer and long variable store integer type data {-∞,…,-3,-2,-1,0,1,2,3,…,∞}

-float (32 bits) 7 decimal digits

-double (64 bits) 15 decimal digits

Stores floating point numbers like 3.14

TYPE CASTING:

Converting value from one datatype to another data type.

public class Main {

  public static void main(String[] args) {

   int i=5;

    System.out.println(i);

   //long l = 10000000000;// this is wrong assigning, java first read number as integer then assign to variable

   /\*

   [CompilationError] Your code was terminated due to compilation error

   Main.java:6: error: integer number too large: 10000000000

   long l = 10000000000;// this is wrong assigning, java first read number as integer then assign to variable

   \*/

   long l = 10000000000l;// when we give l at the end then compiler read it as long value then assign to variable

   System.out.println(l);

    // similarly by default java read decimal number as double

    float f=3.14f;

    System.out.println(f);

    double d =1.1456789;

    System.out.println(d);

    //Type casting

    //low -> high , implicit typecasting

    int i1=5;

    long l1=i1;

    System.out.println(i1+"\n"+l1);

    // high -> low

    long l2=100000000000l;

   //int i2=l2;

    /\*

    [CompilationError] Your code was terminated due to compilation error

    Main.java:33: error: incompatible types: possible lossy conversion from long to int

    int i2=l2;

    \*/

  // forcefully converting high to low , explicit typecasting

    int i2=(int)l2;

    System.out.println("i2="+i2+"\nl2="+l2);// i2 store some garbage value

    long l3=1000l;

    int i3=(int)l3;

    System.out.println("i3="+i3+"\nl3="+l3);//both value print correctly as 1000 is within 32 bit range

  }

}

import java.util.Scanner;

public class Main {

  public static void main(String[] args) {

    //Taking inputs

    Scanner ip=new Scanner(System.in);

    int i=ip.nextInt();

    long l=ip.nextLong();

    float f=ip.nextFloat();

    double d=ip.nextDouble();

    System.out.println("i="+i+"\nl="+l+"\nf="+f+"\nd="+d);

    String s=ip.next();// Read one word at a time, it read input till 1st space occured

    System.out.println(s);

    String s1=ip.nextLine();// it will read till new line character

    System.out.println(s1);

    String s2=ip.nextLine();

    System.out.println(s2);

    // print name1 loves name2

    String name1=ip.next();

    String name2=ip.next();

    System.out.println(name1+" loves "+name2);

    // take name and age and print x age is y

    int age=ip.nextInt();

    String name3=ip.next();

    System.out.println(name3+" age is "+age);

  }

}

CUSTOM INPUT:

5

1000000000000000000

3.14

8.8888888888888

Shubham Lekule

Awesome Shubham

Shubham

Pikachu

30

Shubham

OUTPUT:  
i=5

l=1000000000000000000

f=3.14

d=8.8888888888888

Shubham

Lekule

Awesome Shubham

Shubham loves Pikachu

Shubham age is 30

**Problem Description**  
Print "Hello World !" in the output.  
  
**Output Format**

Print in a single line "Hello World !"  
  
**Example Output**

Hello World !  
  
Note - You don't have to print "" in the output

CODE:

import java.lang.\*;

import java.util.\*;

public class Main {

    public static void main(String[] args) {

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

    }

}

**Problem Description**  
Print the following text:

Hello 40

50 World !  
**Output Format**

Print the output specifed in the problem description.  
  
**Example Output**

Hello 40  
50 World !

CODE:

import java.lang.\*;

import java.util.\*;

public class Main {

    public static void main(String[] args) {

        System.out.println("Hello 40");

        System.out.println("50 World !");

    }}

**Problem Description**  
Print the following text in the output:

Hello

World !

Note: There is a empty line you need print between Hello and World !

**Output Format**

Print the required text in the output.

**Example Output**

Hello

World !

CODE:

import java.lang.\*;

import java.util.\*;

public class Main {

    public static void main(String[] args) {

        System.out.println("Hello");

        System.out.println();

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

    }

}

**Problem Description**  
Print the text

Hello

World !

**Output Format**

Hello

World !

CODE:

import java.lang.\*;

import java.util.\*;

public class Main {

    public static void main(String[] args) {

        System.out.println("Hello");

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

    }

}

**Problem Description**

Print the following pattern in output

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

**Output Format**

Print the following pattern in 5 lines with 5, 4, 3, 2, 1 stars in each line

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

**Example Output**

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

CODE:

import java.lang.\*;

import java.util.\*;

public class Main {

    public static void main(String[] args) {

        System.out.println("\*\*\*\*\*");

        System.out.println("\*\*\*\*");

        System.out.println("\*\*\*");

        System.out.println("\*\*");

        System.out.println("\*");

    }

}

**Problem Description**

Print the following pattern in output

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

**Output Format**

Print the following pattern in 5 lines with 4 spaces and 1 star, 3 spaces and 2 stars, 2 spaces and 3 stars, 1 spaces and 4 stars, 0 spaces and 5 stars in respectively lines

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

**Example Output**

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

CODE:  
import java.lang.\*;

import java.util.\*;

public class Main {

    public static void main(String[] args) {

       System.out.println("    \*");

       System.out.println("   \*\*");

       System.out.println("  \*\*\*");

       System.out.println(" \*\*\*\*");

       System.out.println("\*\*\*\*\*");

    }

}

**Q7. Simple MathsSolved**

**Problem Description**  
Perform the following operations on 6 and 3 and print their respective outputs in different lines.

1. Addition  
2. Subtraction  
3. Multiplication  
4. Division  
  
**Output Format**

Print the output of 6+3, 6-3, 6\*3, 6/3 in separate lines.  
  
**Example Output**

9  
3  
18  
2  
**Example Explanation**

6 + 3 = 9  
6 - 3 = 3  
6 \* 3 = 18  
6 / 3 = 2

CODE:

import java.lang.\*;

import java.util.\*;

public class Main {

    public static void main(String[] args) {

        System.out.println(6+3);

        System.out.println(6-3);

        System.out.println(6\*3);

        System.out.println(6/3);

    }

}

**Q8. A says Hi to B**

**Problem Description**  
Take two names **A** and **B** as input from the user, print "A says Hi to B" (Without quotations), where **A** and **B** are the names in input.  
  
**Problem Constraints**

1 <= len(A), len(B) <= 15  
Characters in A and B are in lowercase English Alphabets.

**Input Format**

There are two input lines  
The first line has a string A.  
The second line has a string B.

**Output Format**

Print in a single line A says Hi to B.

**Example Input**

Input:-

Ram

Shyam

**Example Output**

Output:-

Ram says Hi to Shyam

CODE:

import java.lang.\*;

import java.util.\*;

public class Main {

    public static void main(String[] args) {

        Scanner ip=new Scanner(System.in);

        String name1=ip.next();

        String name2=ip.next();

        System.out.println(name1+" says Hi to "+name2);

    }

}

**Q9. Print A B C D EProblem Description**

**Print the first five letters of the English alphabet i.e. A, B, C, D and E.**

**Output Format**

**Print the characters in separate lines.**

**Example Output**

**A  
B  
C  
D  
E**

CODE:

public class Main {

    public static void main(String[] args) {

        System.out.println("A\nB\nC\nD\nE");

    }

}

**Q10. A Simple Calculator**

**Problem Description**

Given two numbers **A** and **B**. Print A+B, A\*B, A-B, A/B in the same order.  
  
**Problem Constraints**

1 <= A, B <= 104  
  
**Input Format**

There are two input lines  
The first line has a single integer **A**.  
The second line has a single integer **B**.  
**Output Format**

Print in a single line separated by space A+B, A\*B, A-B, A/B in the same order.  
  
**Example Input**

Input 1:-

4

5

Input 2:-

16

2

**Example Output**

Output 1:-

9 20 -1 0

Output 2:-

18 32 14 8  
  
**Example Explanation**

Explanation 1:-

4 + 5 = 9, 4 \* 5 = 20, 4 - 5 = -1, 4 / 5 = 0

Explanation 2:-

16 + 2 = 18, 16 \* 2 = 32, 16 - 2 = 14, 16 / 2 = 8

CODE:

public class Main {

    public static void main(String[] args) {

       Scanner ip=new Scanner(System.in);

       int x=ip.nextInt(),y=ip.nextInt();

       System.out.println((x+y)+" "+(x\*y)+" "+(x-y)+" "+(x/y));

    }

}

**Q11. Concatenate Two Numbers**

**Problem Description**

Given two numbers **A** and **B**. Concatenate the two numbers and print it.  
  
**Problem Constraints**

1 <= A, B <= 104  
**Input Format**

There are two input lines  
The first line has a single integer **A**.  
The second line has a single integer **B**.  
  
**Output Format**

Print in a single line the concatenated number.  
**Example Input**

Input 1:-

4

5

Input 2:-

16

2  
**Example Output**

Output 1:-

45

Output 2:-

162  
**Example Explanation**

Explanation 1:-

Concatenation of 4 and 5 is 45.

Explanation 2:-

Concatenation of 16 and 2 is 162.

CODE 1:

public class Main {

    public static void main(String[] args) {

        Scanner ip=new Scanner(System.in);

        String num1=ip.next(),num2=ip.next();

        System.out.println(num1+num2);

    }

}

CODE 2:

public class Main {

    public static void main(String[] args) {

        Scanner ip=new Scanner(System.in);

        int num1=ip.nextInt(),num2=ip.nextInt();

        System.out.println(num1+""+num2);

    }

}

**Q1. Add 10**

Problem Description

You will be given an integer in the input. You need to add 10 to it and print the result to the output.

Input Format

One integer value in the input.

Output Format

Print a single integer the sum.

Problem Constraints

1 <= N <= 1000

Example Input

Input 1:-

3

Input 2:-

11

Example Output

Output 1:-

13

Output 2:-

21

CODE:

public class Main {

    public static void main(String[] args) {

        System.out.println(10+new Scanner(System.in).nextInt());

    }

}

**Q2. Print sentence**

**Problem Description**

Given an english sentence, take it in the input and print it in the output.

**Input Format**

Single line containing an english sentence.

**Output Format**

Print in a single line the sentence in the input.

**Example Input**

**Input 1:**

Harry loves Hagrid !

**Input 2:**

I am a Scaler

**Example Output**

**Output 1:**

Harry loves Hagrid !

**Output 2:**

I am a Scaler

CODE:

public class Main {

    public static void main(String[] args) {

       System.out.println(new Scanner(System.in).nextLine());

    }

}

**Q3. Hello Name**

**Problem Description**

Take a name **A** as input from the user and print "Hello A", where A is the name in input.

**Problem Constraints**

1 <= len(A) <= 15 Characters in A are in lowercase English Alphabets.

**Input Format**

There is a single input line, which is the string \*\*A\*\*.

**Output Format**

Print in a single line "Hello A" without quotes.

**Example Input**

Input 1:-

Ram

Input 2:-

Shyam

**Example Output**

Output 1:-

Hello Ram

Output 2:-

Hello Shyam

CODE:

public class Main {

    public static void main(String[] args) {

        System.out.println("Hello "+new Scanner(System.in).next());

    }

}

**Q4. Number of bills**

**Problem Description**

Given total bills amount and amount of a single bill. Print number of bills.  
  
**Note :** The total amount is equally splitted in all bills. The number of bills should be an integer value.

**Input Format**

The first line contains a real number N denoting the total budget.

The second line contains an integer M denoting the value of a single bill.

**Output Format**

Print in a single line denoting the total number of bills that can fit in the total budget.

**Problem Constraints**

1 <= N <= 10000  
1 <= M <= 100

**Example Input**

Input:-

126.3

5

**Example Output**

Output:-

25

**Note:** The **problem constraints** mean that when we test your code, the test cases used in the backend can have input values only within those constraints. You need not implement them in your code. You need to make sure your code will work for all such input values!

CODE:

public class Main {

    public static void main(String[] args) {

        Scanner ip=new Scanner(System.in);

        System.out.println((int)(ip.nextDouble()/ip.nextInt()));

    }

}

**Q5. Total Bills Value**

**Problem Description**

Given the value of a single bill and the number of bills you received, print the total value of the bills.  
  
**Note:** The value of all the bills are same  
  
**Problem Constraints**

1 <= N <= 100  
1 <= M <= 100  
**Input Format**

The first line of the input is an integer N denoting the value of a single bill.  
The second line of the input is an integer M denoting the number of bills.  
**Output Format**

Print in a single line denoting the total value of bills.  
**Example Input**

Input:-  
12  
10  
**Example Output**

Output:-  
120

CODE:

public class Main {

    public static void main(String[] args) {

        Scanner ip=new Scanner(System.in);

        System.out.println(ip.nextInt()\*ip.nextInt());

    }

}

**Q6. Remaining Bake Time**

**Problem Description**

You're going to write some code to help you cook a gorgeous lasagna from your favorite cookbook. According to your cookbook, the Lasagna should be in the oven for 40 minutes. Given the time (in minutes), the lasagna has been in the oven, find how many more minutes the lasagna still needs to bake for.  
  
**Problem Constraints**

0 <= N <= 40

**Input Format**

The only first line contains the integer N, denoting the actual time (in minutes) the lasagna has been in the oven for.

**Output Format**

Print in a single line how many minutes the lasagna needs to bake.  
  
**Example Input**

Input:   
30  
  
**Example Output**

Output:   
10

CODE:

public class Main {

    public static void main(String[] args) {

        System.out.println(40-new Scanner(System.in).nextInt());

    }

}

**Q7. Preparation Time**

**Problem Description**

You'll write some code to help you cook a gorgeous lasagna from your favorite cookbook. Now, you also want to add a few layers to the lasagna. Assume \*\*each layer takes 2 minutes\*\* to prepare. Given the number of layers you want to add to the lasagna, find how many minutes you would spend making them.  
**Input Format**

The only first line contains the integer N denoting the number of layers.  
**Output Format**

Print in a single line how many minutes are required to prepare N layers.  
**Example Input**

Input:-  
2  
**Example Output**

Output:-  
4

CODE:

public class Main {

    public static void main(String[] args) {

        System.out.println(2\*new Scanner(System.in).nextInt());

    }

}

**Q8. Total Elapsed Cooking Time**

**Problem Description**

You wrote some code to help you cook a gorgeous lasagna from your favorite cookbook. Now, you want to find the total number of minutes you've been cooking for the sum of your preparation time and the time the lasagna has already spent baking in the oven. The preparation time of one layer is 2 minutes. Given the number of layers added to the lasagna and the number of minutes the lasagna has been baking in the oven, find the total elapsed cooking time (prep + bake) in minutes.  
**Problem Constraints**

1 <= N <= 20  
0 <= M <= 40  
**Input Format**

There are 2 lines in the input.  
The first line contains the integer N denoting the number of layers.  
The second line contains the integer M denoting the time the lasagna has already spent baking in the oven.  
**Output Format**

Print in a single line the total elapsed cooking time.  
**Example Input**

Input 1:-  
3  
20  
Input 2:-  
1  
29  
**Example Output**

Output 1:-  
26  
  
Output 2:-  
31

CODE:

public class Main {

    public static void main(String[] args) {

        Scanner ip=new Scanner(System.in);

        System.out.println(ip.nextInt()\*2+ip.nextInt());

    }

}

IF ELSE:

import java.util.Scanner;

public class Main {

  public static void main(String[] args) {

    Scanner ip=new Scanner(System.in);

    // eligible for vote or not

    int age=ip.nextInt();

    if(age<18){

      System.out.println("Not Eligible");

    }

    else{

      System.out.println("Eligible");

    }

    age=ip.nextInt();

    if(age<18){

      System.out.println("Not Eligible");

    }

    else{

      System.out.println("Eligible");

    }

    //Max of two number

    int num1=ip.nextInt();

    int num2=ip.nextInt();

    if(num1>num2){

      System.out.println(num1+" is greater");

    }else if(num2>num1){

      System.out.println(num2+" is greater");

    }else{

      System.out.println(num1+" and "+num2+" both are same");

    }

    num1=ip.nextInt();

    num2=ip.nextInt();

    if(num1>num2){

      System.out.println(num1+" is greater");

    }else if(num2>num1){

      System.out.println(num2+" is greater");

    }else{

      System.out.println(num1+" and "+num2+" both are same");

    }

    num1=ip.nextInt();

    num2=ip.nextInt();

    if(num1>num2){

      System.out.println(num1+" is greater");

    }else if(num2>num1){

      System.out.println(num2+" is greater");

    }else{

      System.out.println("both are same");

    }

    // temp low high or normal, normal = 98.2 to 98.8

    float temp=ip.nextFloat();

    System.out.println(temp>98.8?"High":(temp<98.2?"low":"Normal"));

    temp=ip.nextFloat();

    System.out.println(temp>98.8?"High":(temp<98.2?"low":"Normal"));

    temp=ip.nextFloat();

    System.out.println(temp>98.8?"High":(temp<98.2?"low":"Normal"));

  }

}

CUSTOM INPUT:  
30

15

10

20

20

10

10

10

98.1

98.5

99.1  
OUTPUT:  
Eligible

Not Eligible

20 is greater

20 is greater

both are same

low

Normal

High

import java.util.Scanner;

public class Main {

  public static void main(String[] args) {

    //OPERATORS

    // Division (/)

    /\*

      int/int -> int

      float/int -> float

      int/float -> float

      float/float -> float

      long/int -> long

      double/float -> double

    \*/

    System.out.println(9/3);

    System.out.println(11/3);

    System.out.println(11f/3);

    System.out.println(11d/3);

    //Multiplication (\*)

    /\*

      int \* int -> int

      int \* float -> float

      float \* int -> float

      float \* float -> float

      long \* int -> long

      float \* double -> double

    \*/

    int x= 100000;//10^5

    int y= 100000;//10^5

    System.out.println(x\*y);// it will print garbage value, as int \* int -> int

     System.out.println((long)(x\*y));// it will print garbage value, 1st int \* int -> int then int type cast to long

    long z=x\*y;

    System.out.println(z);// it will print garbage value, 1st int \* int -> int then int assign to long

    z=(long)x\*y;

    System.out.println(z);// it will print correctly

    System.out.println(11.0/3);

    //Modulo (%)

    // odd or even

    Scanner ip=new Scanner(System.in);

    int num=ip.nextInt();

    if(num%2==0){

      System.out.println("even");

    }else{

      System.out.println("odd");

    }

    num=ip.nextInt();

    if(num%2==0){

      System.out.println("even");

    }else{

      System.out.println("odd");

    }

    // given a integer print its last digit

    int num1=ip.nextInt();

    System.out.println(num1%10);

    //OPERATORS

    //Relational OPERATORS

    /\*

      a>b

      a<b

      a>=b

      a<=b

      a==b

      a!=b

    \*/

    //Logical OPERATORS

    // AND OPERATOR (&&)

    /\*

    T && T -> T

    T && F -> F

    F && T -> F

    F && F -> F

    \*/

    // OR OPERATOR (||)

    /\*

    T || T -> T

    T || F -> T

    F || T -> T

    F || F -> F

    \*/

    // NOT OPERATOR (!)

    /\*

    !T -> F

    !F -> T

    \*/

    /\*

      Q. Given units of electricity as an integer input A, print the bill amount

      [1 - 50] -> 1

      [51 - 100] -> 2

      [101 and beyond] -> 4

    \*/

    int unit=ip.nextInt();

    if(unit>100){

      System.out.println(150+(unit-100)\*4);

    }else if(unit>50){

      System.out.println(50+(unit-50)\*2);

    }else{

      System.out.println(unit);

    }

    unit=ip.nextInt();

    if(unit>100){

      System.out.println(150+(unit-100)\*4);

    }else if(unit>50){

      System.out.println(50+(unit-50)\*2);

    }else{

      System.out.println(unit);

    }

    unit=ip.nextInt();

    if(unit>100){

      System.out.println(150+(unit-100)\*4);

    }else if(unit>50){

      System.out.println(50+(unit-50)\*2);

    }else{

      System.out.println(unit);

    }

    //Q if given number multiple of 3 print Fizz , if multiple of 5 then Buzz, if both then FizzBuzz

    int fizzBuzzNum=ip.nextInt();

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

      System.out.println("FizzBuzz");

    } else if(fizzBuzzNum%3==0){

      System.out.println("Fizz");

    } else if(fizzBuzzNum%5==0){

      System.out.println("Buzz");

    }

    fizzBuzzNum=ip.nextInt();

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

      System.out.println("FizzBuzz");

    } else if(fizzBuzzNum%3==0){

      System.out.println("Fizz");

    } else if(fizzBuzzNum%5==0){

      System.out.println("Buzz");

    }

    fizzBuzzNum=ip.nextInt();

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

      System.out.println("FizzBuzz");

    } else if(fizzBuzzNum%3==0){

      System.out.println("Fizz");

    } else if(fizzBuzzNum%5==0){

      System.out.println("Buzz");

    }

  }

}

CUSTOM INPUT:

3

6

73

20

80

120

6

5

30  
OUTPUT:

3

3

3.6666667

3.6666666666666665

1410065408

1410065408

1410065408

10000000000

3.6666666666666665

odd

even

3

20

110

230

Fizz

Buzz

FizzBuzz

**Q1. Categorise the number - Nested if-else**

**Problem Description**

Given the number **N,**Categorise the number according to following condition :  
1. Odd-Positive  
2. Odd-Negative  
3. Even-Positive  
4. Even-Negative  
**Note** : Intention of problem is to teach you **Nested If-Else**, so try to solve this problem using nested if-else  
**Problem Constraints**

-10000 <=**N** <= 10000 **except 0**  
**Input Format**

Take Number in single line.  
**Output Format**

Print the statement, according to number N in single line.  
**Example Input**

**Input 1** :  
15  
**Input 2** :  
-38  
**Example Output**

**Output 1** :  
Odd-Positive  
**Output 2** :  
Even-Negative  
**Example Explanation**

**Example Explanation 1 :**  
N is 15, which is Odd and Positive.  
**Example Explanation 2 :**N is -38, which is Even and Negative.

CODE:

public class Main {

    public static void main(String[] args) {

        Scanner ip=new Scanner(System.in);

        int num=ip.nextInt();

        if(num<0 && num%2==0){

            System.out.println("Even-Negative");

        }else if(num<0 && num%2!=0){

            System.out.println("Odd-Negative");

        }else if(num>0 && num%2==0){

            System.out.println("Even-Positive");

        }else{

            System.out.println("Odd-Positive");

        }

    }

}

**Q2. Check even/odd**

**Problem Description**

Write a program to input an integer from user and print **1** if it is odd otherwise print **0**.

**Problem Constraints**

1 <= N <= 1000000  
**Input Format**

One line containing an integer **N**.  
**Output Format**

Print either 1 or 0 as per the question.  
**Example Input**

Input 1:

5

Input 2:

1000  
**Example Output**

Output 1:

1

Output 2:

0  
**Example Explanation**

Explanation 1:

Clearly, 5 is odd.

Explanation 2:

Clearly, 1000 is even.

CODE:

public class Main {

    public static void main(String[] args) {

        System.out.print(new Scanner(System.in).nextInt()%2==0?0:1);

    }

}

**Q3. Which Month?**

**Problem Description**

Write a program to input an integer(**A**) from user and print the Ath month of the year.

Months list: **{January, February, March, April, May, June, July, August, September, October, November, December}**  
**Problem Constraints**

1 <= A <= 12  
**Input Format**

One line containing an integer integer **A**.  
**Output Format**

One line containing the **Ath** month of the year.  
  
**Example Input**

Input 1:

1

Input 2:

6  
  
**Example Output**

Output 1:

January

Output 1:

June  
  
**Example Explanation**

Explanation 1:

Clearly, January is the 1st month.

Explanation 2:

Clearly, June is the 6th month.

CODE:

public class Main {

    public static void main(String[] args) {

        int month=new Scanner(System.in).nextInt();

        if(month==1) System.out.println("January");

        if(month==2) System.out.println("February");

        if(month==3) System.out.println("March");

        if(month==4) System.out.println("April");

        if(month==5) System.out.println("May");

        if(month==6) System.out.println("June");

        if(month==7) System.out.println("July");

        if(month==8) System.out.println("August");

        if(month==9) System.out.println("September");

        if(month==10) System.out.println("October");

        if(month==11) System.out.println("November");

        if(month==12) System.out.println("December");

    }

}

**Q4. Max of three**

**Problem Description**

Write a program to input three numbers(**A, B & C**) from user and print the maximum element among A, B & C in each line.

**Problem Constraints**

1 <= A <= 1000000

1 <= B <= 1000000

1 <= C <= 1000000  
**Input Format**

First line is a single integer **A**.  
Second line is a single integer **B**.  
Third line is a single integer **C.**  
**Output Format**

One line containing an integer as per the question.  
**Example Input**

Input 1:

5   
6   
7

Input 2:

1000   
10000   
100000  
**Example Output**

Output 1:

7

Output 2:

100000  
  
**Example Explanation**

Explanation 1:

Clearly, among 5, 6 and 7, 7 is maximum.

Explanation 2:

Clearly, among 1000, 10000 and 100000, 100000 is maximum.

CODE:

public class Main {

    public static void main(String[] args) {

        Scanner ip=new Scanner(System.in);

        int a=ip.nextInt(),b=ip.nextInt(),c=ip.nextInt();

        System.out.println(a>b?(a>c?a:c):(b>c?b:c));

    }

}

**Q5. Percentage and Grade**

**Problem Description**

Write a program to calculate the percentage (according to marks of a student) and grade (according to the percentage of a student). Five numbers(A, B, C, D & E) represent the marks of a student in 5 subjects which are out of 100. Print the percentage and the grade of the student.  
If percentage >= 90% : Grade A  
If percentage >= 80% but <90 : Grade B  
If percentage >= 70% but <80: Grade C  
If percentage >= 60% but <70: Grade D  
If percentage >= 40% but <60: Grade E  
If percentage < 40%: Grade F  
NOTE: You have to take the lowest integer of the percentage. E.g. 90.8% will be treated as 90%.  
**Input Format**

There will be five lines of inputs as following:  
The five lines contain the 5 subject marks of the student in numerical format.  
**Output Format**

The first line indicates the percentage in integer format.  
The next line displays the grade in string format.  
**Example Input**

50  
60  
70  
80  
90

**Example Output**

70  
C

CODE:

public class Main {

    public static void main(String[] args) {

        Scanner ip=new Scanner(System.in);

        int mark=(ip.nextInt()+ip.nextInt()+ip.nextInt()+ip.nextInt()+ip.nextInt())/5;

        System.out.println(mark);

        if(mark>=90) System.out.println("A");

        else if(mark>=80) System.out.println("B");

        else if(mark>=70) System.out.println("C");

        if(mark>=60) System.out.println("D");

        if(mark>=40) System.out.println("E");

        else System.out.println("F");

    }

}

**Q6. Which triangle?**

**Problem Description**

Write a program to input from user three numbers(**A, B & C**) representing side lengths of a triangle.

You have to print if the traingle is "equilateral", "scalene" or "isosceles".  
**Problem Constraints**

1 <= A <= 100000

1 <= B <= 100000

1 <= C <= 100000  
**Input Format**

One line containing three space separated integers **A, B & C**.  
**Output Format**

One string either "equilateral", "scalene" or "isosceles".  
**Example Input**

Input 1:

5 6 7

Input 2:

30 30 30  
**Example Output**

Output 1:

scalene

Output 2:

equilateral  
**Example Explanation**

Explanation 1:

Since all sides are different, hence it's a scalene triangle.

Explanation 2:

Since all sides are same, hence it's a equilateral triangle.

CODE:

public class Main {

    public static void main(String[] args) {

        Scanner ip=new Scanner(System.in);

        int A=ip.nextInt(),B=ip.nextInt(),C=ip.nextInt();

        if(A==B && B==C && A==C)System.out.print("equilateral");

        else if(A==B || B==C || A==C)System.out.print("isosceles");

        else System.out.print("scalene");

    }

}

**Q7. Min of two**

**Problem Description**

Write a program to input two numbers(**A & B**) from user and print the minimum element among A & B in each line.  
**Problem Constraints**

1 <= A <= 1000000

1 <= B <= 1000000  
**Input Format**

First line is a single integer **A**.  
Second line is a single integer **B**.  
**Output Format**

One line containing an integer as per the question.  
**Example Input**

Input 1:

5   
6

Input 2:

1000   
10000  
**Example Output**

Output 1:

5

Output 2:

1000  
  
**Example Explanation**

Explanation 1:

Clearly, among 5 and 6, 5 is minimum.

Explanation 2:

Clearly, among 1000 and 10000, 1000 is minimum.

CODE:

public class Main {

    public static void main(String[] args) {

        Scanner ip=new Scanner(System.in);

        int num1=ip.nextInt();

        int num2=ip.nextInt();

        System.out.println(num1<num2?num1:num2);

    }

}

**Q8. Confusion In Electricity Bill**

**Problem Description**

Mr. T got the Electricity bill of his house. He felt that the bill amount was too much. He come to you to understand the relation between amount and number of units with Electricity bill.  
Instructions are give on Electricity biil that :  
1. For **first 50** units **Rs. 0.50/unit**.  
2. For **next 100** units **Rs. 0.75/unit**.  
3. For **next 100** units **Rs. 1.20/unit**.  
4. For **above 250** units **Rs. 1.50/unit**.  
5. An additional **surcharge of 20%** is added to the bill.

**NOTE:** As the electricity bill can have any real value (floating point), you have to tell the Integral value of the bill. For eg. Integral value of 2.91 is 2.  
To avoid manual calculation again and again, You have to write a code which take number of Unit (**suppose N**) from user and print the amount.  
**Problem Constraints**

0 < **N** <= 100000  
**Input Format**

Take an integer **N** from user  
**Output Format**

Print the amount for that much unit  
**Example Input**

Input 1 :

N = 150

Input 2 :

N = 4

Input 3 :

N = 300  
**Example Output**

Output 1 :

120

Output 2 :

2

Output 3 :

354  
**Example Explanation**

Explanation 1:

For first 50 units, the bill is (Rs 0.5/unit) \* (50 units) = Rs 25

For next 100 units, the bill is (Rs 0.75/unit) \* (100 units) = Rs 75

Bill amount without additional surcharge (25 + 75) = Rs 100

Total Bill amount with additional surcharge = Rs (100 + 0.20 \* 100) = Rs 120

Explanation 2:

Bill without additional surcharge (Rs 0.5/unit) \* (4 units) = Rs 2

Total Bill amount with additional surcharge = Rs (2 + 0.20 \* 2) = Rs 2.4

Integral value of Bill amount = Rs 2

Explanation 3:

For first 50 units, the bill is (Rs 0.5/unit) \* (50 units) = Rs 25

For next 100 units, the bill is (Rs 0.75/unit) \* (100 units) = Rs 75

For next 100 units, the bill is (Rs 1.20/unit) \* (100 units) = Rs 120

For the remaining 50 units, the bill is (Rs. 1.50/unit) \* (50 units) = Rs 75

Bill amount without additional surcharge (25 + 75 + 120 + 75) = Rs 295

Total Bill amount with additional surcharge = Rs (295 + 0.20 \* 295) = Rs 354

CODE:

public class Main {

public static void main(String[] args) {

/\*

1 to 50 -> 0.50

51 to 150 -> 0.75

151 to 250 -> 1.20

251 > -> 1.50

\*/

Scanner ip = new Scanner(System.in);

int units=ip.nextInt();

double cal;

if(units>250){

cal=(units-250)\*1.50+100\*1.20+100\*0.75+50\*0.50;

}else if(units>150){

cal=(units-150)\*1.20+100\*0.75+50\*0.50;

}else if(units>51){

cal=(units-50)\*0.75+50\*0.50;

}else{

cal=units\*0.50;

}

int result=(int)(cal+(cal\*20/100));

System.out.println(result);

}

}

**Q9. Print last digit**

**Problem Description**

Given an integer in the input print its last digit.

**Input Format**

Single line containing an integer.

**Output Format**

Print in a single line the last digit of input integer.

**Problem Constraints**

1 <= N <= 1000

**Example Input**

Input 1:-

3

Input 2:-

29

**Example Output**

Output 1:-

3

Output 2:-

9

CODE:

public class Main {

    public static void main(String[] args) {

        System.out.println(new Scanner(System.in).nextInt()%10);

    }

}

**Q10. Fizz Buzz**

**Problem Description**

Write a program that takes in a number **N**as input and does the following:

* if **N**is a multiple of 3, print **Fizz**
* if **N**is a multiple of 5, print **Buzz**
* if **N**is a multiple of both 3 and 5, print **FizzBuzz**

**Problem Constraints:**

1 <= N <= 1000  
**Input Format**

There is only 1 single line in the input, which is the integer **N.**  
**Output Format**

Print **Fizz / Buzz / FizzBuzz**depending on the value **N.**  
**Example Input**

Input 1:-

9

Input 2:-

15  
**Example Output**

Output 1:-

Fizz

Output 2:-

FizzBuzz

CODE:

public class Main {

    public static void main(String[] args) {

        int num=new Scanner(System.in).nextInt();

        if(num%3==0 && num%5==0){

            System.out.println("FizzBuzz");

        }else if(num%5==0){

            System.out.println("Buzz");

        }else if(num%3==0){

            System.out.println("Fizz");

        }

    }

}

**Q1. Angles Of Valid Triangle?**

**Problem Description**

You are given 3 integer angles(in degrees) **A, B** and **C** of a triangle. You have to tell whether the triangle is valid or not.

A triangle is valid if sum of its angles equals to 180.

**NOTE:** You have to take the **input** of 3 sides of triangle from the user.  
**Problem Constraints**

1 <= **A, B, C** <= 180  
**Input Format**

First line of the input contains an integer **A**.

Second line of the input contains an integer **B**.

Third line of the input contains an integer **C**.  
**Output Format**

Print 1 if the triangle having given sides is valid, else print 0.  
**Example Input**

Input 1:

60

60

60

Input 2:

30

40

50  
**Example Output**

Output 1:

1

Output 2:

0   
**Example Explanation**

Explanation 1:

Sum of angles = A + B + C = 60 + 60 + 60 = 180

Hence, the given triangle is valid.

Explanation 2:

Sum of angles = A + B + C = 30 + 40 + 50 = 120

As sum of angles is not equal to 180, the given triangle is not valid.

CODE:

public class Main {

    public static void main(String[] args) {

        Scanner ip=new Scanner(System.in);

        System.out.println((ip.nextInt()+ip.nextInt()+ip.nextInt())==180?1:0);

    }

}

**Q2. Profit Or Loss**

**Problem Description**

You are given the Cost Price **C** and Selling Price **S** of a Product. You have to tell whether there is a Profit or Loss. Also, calculate total profit or loss.

**NOTE:** It is guaranteed that Cost Price and Selling Price are not equal.

**NOTE:** You have to take input of the **Cost Price(C)** and **Selling Price(S)** from the user.  
**Problem Constraints**

1 <= **C, S** <= 109

**C ≠ S**  
**Input Format**

First line of the input contains a single integer **C**.

Second line of the input contains a single integer **S**.  
**Output Format**

Print two integers in separate lines.

First integer denotes whether there is a profit or loss. If there is a profit, print 1, else print -1.

Second integer is a non-negative integer denoting the absolute value of total profit or loss.  
**Example Input**

Input 1:

2

4

Input 2:

4

1  
**Example Output**

Output 1:

1

2

Output 2:

-1

3  
**Example Explanation**

Explanation 1:

Cost Price = 2

Selling Price = 4

As Cost Price < Selling Price, there is a profit.

Total Profit = Selling Price - Cost Price = 4 - 2 = 2

Explanation 2:

Cost Price = 4

Selling Price = 1

As Cost Price > Selling Price, there is a loss.

Total Loss = Cost Price - Selling Price = 4 - 1 = 3

CODE:

public class Main {

    public static void main(String[] args) {

        Scanner ip=new Scanner(System.in);

        int C=ip.nextInt(),S=ip.nextInt();

        System.out.println(C<S?1+"\n"+(S-C):-1+"\n"+(C-S));

    }

}

**Q3. Coding Rating**

**Problem Description**

Write a program to input from user an integer(**n**) representing the rating of a person on a platform.

You have to print the category of that person.

* If the rating is greater than or equal to 2100 then that person is "grand master".
* If the rating is greater than or equal to 1900 then that person is "candidate master".
* If the rating is greater than or equal to 1600 then that person is "expert".
* If the rating is greater than or equal to 1400 then that person is "pupil".
* If the rating is smaller than 1400 then that person is "newbie".

**NOTE:** Print all the chars of the category of the person in lowercase if rating is odd otherwise print in UPPERCASE  
**Problem Constraints**

1000 <= n <= 2500  
**Input Format**

One line containing an integer**n**.  
**Output Format**

A string representing the category of the person.  
**Example Input**

Input 1:

1659

Input 2:

2100  
**Example Output**

Output 1:

expert

Output 2:

GRAND MASTER  
**Example Explanation**

Explanation 1:

Clearly, 1659 is odd and is in the range of expert.

Explanation 2:

Clearly, 2100 is even and is in the range of grand master.

CODE:

public class Main {

    public static void main(String[] args) {

        int rating=new Scanner(System.in).nextInt();

        if(rating>=2100 && rating%2!=0)System.out.println("grand master");

        else if(rating>=2100 && rating%2==0)System.out.println("GRAND MASTER");

        else if(rating>=1900 && rating%2!=0)System.out.println("candidate master");

        else if(rating>=1900 && rating%2==0)System.out.println("CANDIDATE MASTER");

        else if(rating>=1600 && rating%2!=0)System.out.println("expert");

        else if(rating>=1600 && rating%2==0)System.out.println("EXPERT");

        else if(rating>=1400 && rating%2!=0)System.out.println("pupil");

        else if(rating>=1400 && rating%2==0)System.out.println("PUPIL");

        else if(rating%2!=0)System.out.println("newbie");

        else if(rating%2==0)System.out.println("NEWBIE");

    }

}

**Q4. Days In Month**

**Problem Description**

You are given an integer **A**.

You have to tell how many days are there in the month denoted by **A** in a non-leap year.

Months are denoted as follows:

* January : 1
* February : 2
* March : 3
* April : 4
* May : 5
* June : 6
* July : 7
* August : 8
* September : 9
* October : 10
* November : 11
* December : 12  
  **Problem Constraints**

1 <= **A** <= 12  
**Input Format**

The input contains a single integer **A**.  
**Output Format**

Print a single integer denoting the number of days on a single line.  
**Example Input**

Input 1:

1

Input 2:

11   
**Example Output**

Output 1:

31

Output 2:

30   
**Example Explanation**

Explanation 1:

Number of days in January(1) in a non-leap year = 31.

Explanation 2:

Number of days in November(11) in a non-leap year = 30.

CODE:

public class Main {

    public static void main(String[] args) {

        int month = new Scanner(System.in).nextInt();

        if(month == 1) System.out.println(31);

        if(month == 2) System.out.println(28);

        if(month == 3) System.out.println(31);

        if(month == 4) System.out.println(30);

        if(month == 5) System.out.println(31);

        if(month == 6) System.out.println(30);

        if(month == 7) System.out.println(31);

        if(month == 8) System.out.println(31);

        if(month == 9) System.out.println(30);

        if(month == 10) System.out.println(31);

        if(month == 11) System.out.println(30);

        if(month == 12) System.out.println(31);

    }

}

**Q5. Min of three**

**Problem Description**

Write a program to input three numbers(**A, B & C**) from user and print the minimum element among A, B & C.

**Problem Constraints**

1 <= A <= 1000000

1 <= B <= 1000000

1 <= C <= 1000000  
**Input Format**

First line is a single integer **A**.  
Second line is a single integer **B**.  
Third line is a single integer **C**.  
**Output Format**

One line containing an integer as per the question.  
**Example Input**

Input 1:

5   
6   
7

Input 2:

1000   
10000   
100000  
**Example Output**

Output 1:-

5

Output 2:-

1000  
**Example Explanation**

Explanation 1:

Clearly, among 5, 6 and 7, 5 is minimum.

Explanation 2:

Clearly, among 1000, 10000 and 100000, 1000 is minimum.

CODE:

public class Main {

    public static void main(String[] args) {

        Scanner ip=new Scanner(System.in);

        int A=ip.nextInt(),B=ip.nextInt(),C=ip.nextInt();

        if(A<B){

            if(A<C){

                System.out.println(A);

            }else{

                System.out.println(C);

            }

        }else{

            if(B<C){

                System.out.println(B);

            }else{

                System.out.println(C);

            }

        }

    }

}

**Q6. Bank Account**

**Problem Description**

You are given a Bank account having N amount and you are asked to perform **ADD**(credit) or **SUBTRACT**(debit) operation of an amount X.

After the operation **print the amount left** in the Bank account. If the debit amount is greater than current balance print **"Insufficient Funds"**(without quotes) and the operation is skipped.  
**Problem Constraints**

1 <= N, X <= 105  
**Input Format**

First line contains a single integer N denoting the balance in bank account.

Next line contains two space separated integers **Type and Amount(X)**.

* If Type == 1, Perform ADD operation.
* If Type == 2, Perform SUBTRACT operation.

**Output Format**

Print Amount in the bank balance after the operation.  
**Example Input**

Example Input 1 :  
1000

1 500  
Example Input 2 :  
1000  
2 200  
Example Input 3 :  
1000  
2 1500  
**Example Output**

Example Output 1 :  
1500  
Example Output 2 :  
800  
Example Output 3 :  
Insufficient Funds  
**Example Explanation**

Example 1 :  
Initially bank balance is 1000.

ADD 500, bank balance becomes 1500, print it.  
Example 2 :  
Initially bank balance is 1000.

SUBTRACT 200, bank balance becomes 800, print it.  
Example 3 :  
Initially bank balance is 1000.

SUBTRACT 1500, can't subtract since balance is only 1000, print "Insufficient Finds".

CODE:

public class Main {

    public static void main(String[] args) {

        Scanner ip=new Scanner(System.in);

        int N=ip.nextInt(),operation=ip.nextInt(),X=ip.nextInt();

        if(operation==1){

            System.out.println(X+N);

        }else if(operation==2){

            if(N<X){

                System.out.println("Insufficient Funds");

            }else{

                System.out.println(N-X);

            }

        }

    }

}

**Q7. Divisible by 2 numbers**

**Problem Description**

Take an integer **A** as input. You have to tell whether **A** is divible by **both 5 and 11** or not.  
**Problem Constraints**

1 <= **A** <= 109  
**Input Format**

The input contains a single integer **A**.  
**Output Format**

Print 1 if **A** is divisible by both 5 and 11, else print 0.  
**Example Input**

Input 1:

55

Input 2:

22   
**Example Output**

Output 1:

1

Output 2:

0   
**Example Explanation**

Explanation 1:

55 is divisible by both 5 (5 \* 11 = 55) and 11 (11 \* 5 = 55).

Explanation 2:

22 is divisble by 11 (11 \* 2 = 22),but it is not divisible by 5.

CODE:

public class Main {

    public static void main(String[] args) {

        int num=new Scanner(System.in).nextInt();

        System.out.println(num%5==0 && num%11==0 ? 1 : 0);

    }

}

**Q8. Floor of A/B**

**Problem Description**  
Given two numbers **A** and **B**. Print the floor of **A/B**.  
**Problem Constraints**

1 <= A, B <= 104  
**Input Format**

There are two input lines  
The first line has a single integer **A**.  
The second line has a single integer **B**.  
**Output Format**

Print the floor of A/B in a single line.  
**Example Input**

Input 1:-

4

5

Input 2:-

16

2  
**Example Output**

Output 1:-

0

Output 2:-

8  
**Example Explanation**

Explanation 1:-

floor(4/5) = 0

Explanation 2:-

floor(16/2) = 8

CODE:

public class Main {

    public static void main(String[] args) {

        Scanner ip=new Scanner(System.in);

        System.out.println((int)(ip.nextInt()/ip.nextInt()));

    }

}