WTN METTL SOLUTIONS

Question 1: Is Even?

Test link: https://tests.mettl.com/authenticateKey/2bd025dc

```
import java.io.*;
import  java.util.*;
// Read only region start
class UserMainCode
{
	public int isEven(int input1){
		// Read only region end
		// Write code here...
		if(input1%2==0) return 2;
		else return 1;
	}
}
```

Question 2: Is odd?

Test link: https://tests.mettl.com/authenticateKey/dbdac2a9

```
import java.io.*;
import  java.util.*;
// Read only region start
class UserMainCode
{
	public int isOdd(int input1){
```

```
            if(input1%2!=0) return 2;
            else
                return 1;
        }
}
```

Test Cases Results

Default 2 — Pass: True

Inputs	Expected	Actual	Cpu (ms)	Processing (ms)	Memory (KB)
-4	1	1	0	607	67740

Default 1 — Pass: True

Inputs	Expected	Actual	Cpu (ms)	Processing (ms)	Memory (KB)
3	2	2	0	354	125080

Question 3: Return last digit of the given number

Test link: https://tests.mettl.com/authenticateKey/454f012b

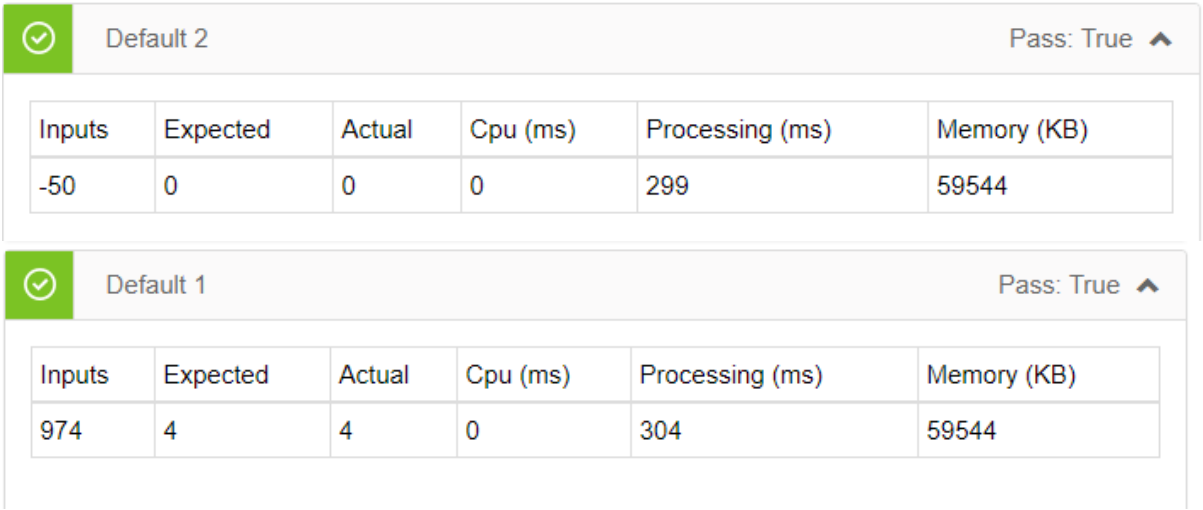
```
import java.io.*;
import  java.util.*;

// Read only region start
class UserMainCode
{

        public int lastDigitOf(int input1){
            // Read only region end
```

```
            if(input1<0)
                input1=(-1)*input1;

            return input1%10;
        }
}
```



Default 2 — Pass: True

Inputs	Expected	Actual	Cpu (ms)	Processing (ms)	Memory (KB)
-50	0	0	0	299	59544

Default 1 — Pass: True

Inputs	Expected	Actual	Cpu (ms)	Processing (ms)	Memory (KB)
974	4	4	0	304	59544

Question 4: Return second last digit of given numbers

Test link: https://tests.mettl.com/authenticateKey/9f87004e

```
import java.io.*;
import  java.util.*;

// Read only region start
class UserMainCode
{       public int secondLastDigitOf(int input1){
        if(input1<0)
            input1=(-1)*input1;
        int c=0;
        int l=Integer.toString(input1).length();
```

```
        int r=0;
                if(l==1)
                return -1;
        else
        {
        while(input1>0)
        {
          r=input1%10;
         c++;
        input1/=10;
        if(c==2)
                break;
        }
                return r;
        }
}
}
```

==========

Question 5: Sum of last digits of two given numbers.

Test link: https://tests.mettl.com/authenticateKey/783a1fcf

```
import java.io.*;
import  java.util.*;
class UserMainCode
{
        public int addLastDigits(int input1,int input2){
        if(input1<0)
                input1=(-1)*input1;
        if(input2<0)
                input2=(-1)*input2;
```

```
        return (input1%10)+(input2%10);
        }
}
```

==========

Question 6 : Is N an exact multiple of M?

Test link: https://tests.mettl.com/authenticateKey/36c4ef58

```
import java.io.*;
import  java.util.*;

// Read only region start
class UserMainCode
{

        public int isMultiple(int input1,int input2){
                // Read only region end
                // Write code here...
                int val=0;
                if(input1==0 || input2==0) val=3;
                else if((input1%input2)!=0) val=1;
                else val=2;
                return val;
        }
}
```

Default 1 Pass: True

Inputs	Expected	Actual	Cpu (ms)	Processing (ms)	Memory (KB)
5,2	1	1	0	280	59544

Default 2 Pass: True

Inputs	Expected	Actual	Cpu (ms)	Processing (ms)	Memory (KB)
8,4	2	2	0	798	125080

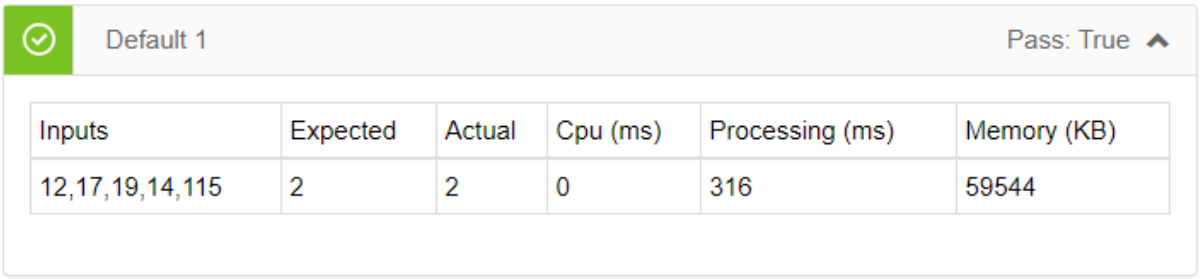
Question 7: Of given 5 numbers, how many are even?

Test link: https://tests.mettl.com/authenticateKey/8edbe922

```
#include<stdio.h>
#include<string.h>
int countEvens(int input1,int input2,int input3,int input4,int input5)
{
   // Read only region end
        // Write code here
        int cnt=0;
   if(input1<0) input1=(-1)*input1;
        if(input2<0) input2=(-1)*input2;
        if(input3<0) input3=(-1)*input3;
        if(input4<0) input4=(-1)*input4;
        if(input5<0) input5=(-1)*input5;
        if(input1%2==0) cnt++;
        if(input2%2==0) cnt++;
        if(input3%2==0) cnt++;
        if(input4%2==0) cnt++;
```

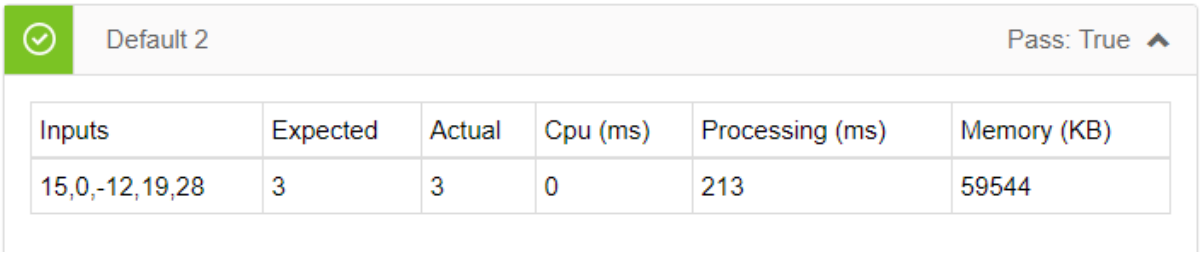
```
        if(input5%2==0) cnt++;


        return cnt;
}
```



Default 1 Pass: True

Inputs	Expected	Actual	Cpu (ms)	Processing (ms)	Memory (KB)
12,17,19,14,115	2	2	0	316	59544



Default 2 Pass: True

Inputs	Expected	Actual	Cpu (ms)	Processing (ms)	Memory (KB)
15,0,-12,19,28	3	3	0	213	59544

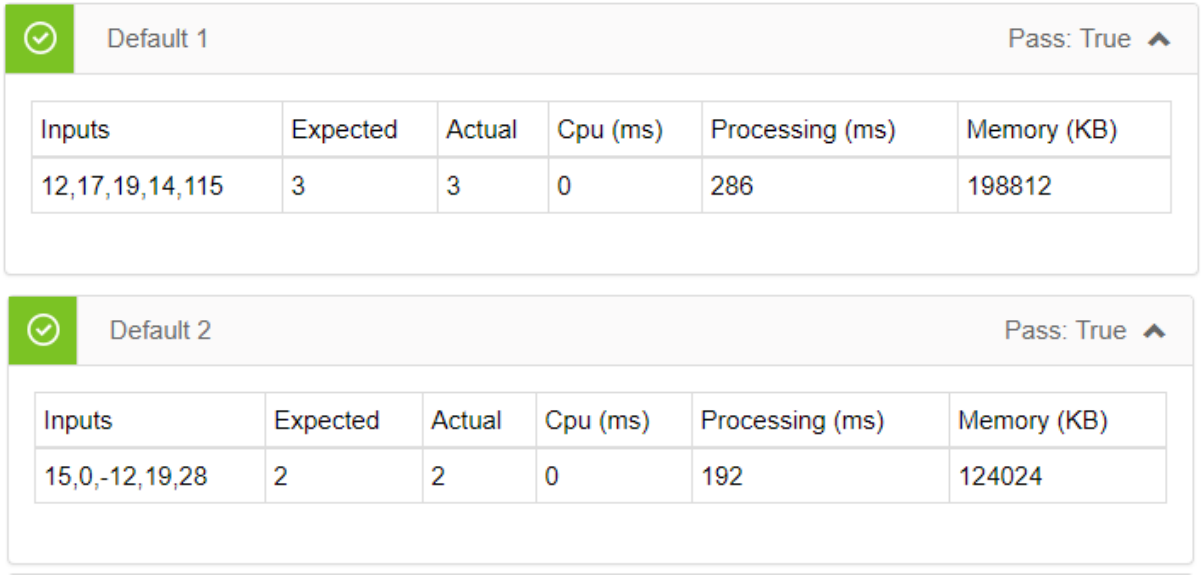
Question 8 : Of given 5 numbers, how many are odd?

Test link: https://tests.mettl.com/authenticateKey/67147bd5

```
import java.io.*;
import  java.util.*;


// Read only region start
class UserMainCode
{
        public int countEvens(int input1,int input2,int input3,int input4,int input5){
                // Read only region end
                int cnt=0;
   if(input1<0) input1=(-1)*input1;
        if(input2<0) input2=(-1)*input2;
        if(input3<0) input3=(-1)*input3;
```

```
        if(input4<0) input4=(-1)*input4;
        if(input5<0) input5=(-1)*input5;
        if(input1%2!=0) cnt++;
        if(input2%2!=0) cnt++;
        if(input3%2!=0) cnt++;
        if(input4%2!=0) cnt++;
        if(input5%2!=0) cnt++;
        return cnt;
        }
}
```



Default 1 Pass: True

Inputs	Expected	Actual	Cpu (ms)	Processing (ms)	Memory (KB)
12,17,19,14,115	3	3	0	286	198812

Default 2 Pass: True

Inputs	Expected	Actual	Cpu (ms)	Processing (ms)	Memory (KB)
15,0,-12,19,28	2	2	0	192	124024

Question 9 : Of 5 numbers, how many are even or odd?

Test link: https://tests.mettl.com/authenticateKey/607636d7

```
import java.io.*;
import  java.util.*;


// Read only region start
```

```
class UserMainCode

{


        public int countEvensOdds(int input1,int input2,int input3,int input4,int input5,String
input6){

                // Read only region end

                int cnt=0;

                if(input6.equalsIgnoreCase("odd")){

                   if(input1<0) input1=(-1)*input1;

        if(input2<0) input2=(-1)*input2;

        if(input3<0) input3=(-1)*input3;

        if(input4<0) input4=(-1)*input4;

        if(input5<0) input5=(-1)*input5;

        if(input1%2!=0) cnt++;

        if(input2%2!=0) cnt++;

        if(input3%2!=0) cnt++;

        if(input4%2!=0) cnt++;

        if(input5%2!=0) cnt++;

        }

        else if(input6.equalsIgnoreCase("even")){

                        if(input1<0) input1=(-1)*input1;

        if(input2<0) input2=(-1)*input2;

        if(input3<0) input3=(-1)*input3;

        if(input4<0) input4=(-1)*input4;

        if(input5<0) input5=(-1)*input5;

        if(input1%2==0) cnt++;

        if(input2%2==0) cnt++;

        if(input3%2==0) cnt++;
```

```
        if(input4%2==0) cnt++;
        if(input5%2==0) cnt++;
                }
                return cnt;

        }
}
```

Default 1 — Pass: True

Inputs	Expected	Actual	Cpu (ms)	Processing (ms)	Memory (KB)
12,17,19,14,115,odd	3	3	0	433	59544

Default 2 — Pass: True

Inputs	Expected	Actual	Cpu (ms)	Processing (ms)	Memory (KB)
15,0,-12,19,28,odd	2	2	0	182	125080

Question 10: Is Prime?

Test link : https://tests.mettl.com/authenticateKey/b1efaa3d

```
import java.io.*;
import  java.util.*;

// Read only region start
class UserMainCode
{

        public int isPrime(int input1){
```

```
            // Read only region end
            int cnt=0;
            for(int i=1;i<=input1;i++){
                if(input1%i==0) cnt++;
            }
            if(cnt==2) return 2;
            else return 1;

        }
}
```

Default 1 Pass: True

Inputs	Expected	Actual	Cpu (ms)	Processing (ms)	Memory (KB)
7	2	2	0	407	125080

Default 2 Pass: True

Inputs	Expected	Actual	Cpu (ms)	Processing (ms)	Memory (KB)
10	1	1	0	185	58488

(11)--->FACTORIAL OF A NUMBER:

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

INPUT: 5

OUTPUT:120(1*2*3*4*5)

SOLUTION:

```
import java.io.*;
import  java.util.*;

// Read only region start
class UserMainCode
{

        public int nFactorial(int input1){
                // Read only region end
int i=1;
                int x=1;
                while(i<=input1){
        x=x*i;
        i++;
}
        return x;
```

```
        }
}
```

---

(12)---->nth FIBONACCI

~~~~~~~~~~~~~~~~~~~~~~~~~

INPUT:4

OUTPUT:2(0,1,1,2)

SOLUTION:

```
import java.io.*;
import  java.util.*;

// Read only region start
class UserMainCode
{

        public long nthFibonacci(int input1){
                // Read only region end
                int a=0;
                int b=1;
                int c=0;
                int d=3;
                while(d<=input1){
```
~~~~~~~~~~~~~~~~~~~~~~~~~

```
                c=a+b;
                    a=b;
                    b=c;
                    d++;
                }
                return c;
        }
}
```

---

(13)----->Nth PRIME:

~~~~~~~~~~~~~~~~~~~~~~~~~

INPUT:5

OUTPUT:11

SOLUTION:

```
import java.io.*;
import  java.util.*;

// Read only region start
class UserMainCode
{

        public int NthPrime(int input1){
```
~~~~~~~~~~~~~~~~~~~~~~~~~

```
        // Read only region end

    int k=2;
    int d=0,i,c=0;
    int p=0;
    while(d<=input1){
        for(i=2;i<k/2;i++){
            if(k%i==0){
                c++;
            }

        }
        if(c==0){
            d++;
            p=k;
        }
        k++;
        c=0;
    }
    return p;
    }
}
```

---

(14)--->NUMBER OF PRIME NUMBERS IN A SPECIFIED RANGE:

~~~~~~

INPUT:2 20

OUTPUT:8(2,3,5,7,11,13,17,19)

SOLUTION:

```
import java.io.*;
import  java.util.*;

// Read only region start
class UserMainCode
{

    public int countPrimesInRange(int input1,int input2){
        // Read only region end
    int k=2;
    int d=input1,i,c=0;
    int p=0;
        int cou=0;
    while(d<=input2){
        for(i=2;i<d;i++){
            if(d%i==0){
                c++;
            }
```
~~~~~~

```
            }

            if(c==0){
                cou++;
                System.out.println(d);
            }
            d++;
            c=0;
        }
        return cou;

        }
}
```

---

(15)----->ALL DIGITS COUNT:

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

INPUT:292

OUTPUT:3

SOLUTION:

```
import java.io.*;
import  java.util.*;
```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

```
// Read only region start
class UserMainCode
{

    public int allDigitsCount(int input1){
        // Read only region end
        int c=0,r;
        while(input1>0){
        r=input1%10;
            c++;
            input1=input1/10;
        }
        return c;
    }
}
```

---

(16)------>UNIQUE DIGITS COUNT:

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

INPUT:292

OUTPUT:2

SOLUTION:
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

```
import java.io.*;
import  java.util.*;

// Read only region start
class UserMainCode
{

        public int uniqueDigitsCount(int input1){
                // Read only region end
        int c=0,r,i;
                int h[]=new int[10];
                while(input1>0){
                r=input1%10;
                        h[r]++;
                        input1=input1/10;
                }
                for(i=0;i<10;i++){
                        if(h[i]>0){
                                c++;
                        }
                }
                return c;
        }
}
```

---

(17)------>NON-REPEATED DIGITS COUNT:

~~~~~~

INPUT:292

OUTPUT:1

SOLUTION:

```
import java.io.*;
import  java.util.*;

// Read only region start
class UserMainCode
{

        public int nonRepeatDigitsCount(int input1){
                // Read only region end

        int c=0,r,i;
                int h[]=new int[10];
                while(input1>0){
                r=input1%10;
                        h[r]++;
                        input1=input1/10;
```

```
            }
            for(i=0;i<10;i++){
                if(h[i]==1){
                    c++;
                }
            }
            return c;
        }
}
```

---

(18)---->DIGIT SUM:

~~~~~~~~~~~~~~~~~~~~

INPUT:-9999

OUTPUT:-9

INPUT:9999

OUTPUT:9

SOLUTION:

```
import java.io.*;
import  java.util.*;

// Read only region start
```
~~~~~~~~~~~~~~~~~~~~

```
class UserMainCode
{

    public int digitSum(int input1){
        // Read only region end
        boolean b=true;
        int r,sum=0;
        int x=input1,res=0;
        input1=Math.abs(input1);
        while(b){
            while(input1>0){
                r=input1%10;
                sum=sum+r;
                input1=input1/10;
            }
            if(sum<10){
                b=false;
            }
            else{
                input1=sum;
                sum=0;
            }
        }
        if(x<0){
```

```
                res=-sum;
            }
            else{
                res=sum;
            }
            return res;
        }
}
```

---

(19)---->EVEN DIGIT'S SUM:

~~~~~~~~~~~~~~~~~~~~~~~~~~~~

INPUT:962

OUTPUT:8

SOLUTION:

```
import java.io.*;
import  java.util.*;

// Read only region start
class UserMainCode
{

        public int EvenDigitsSum(int input1){
```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~

```
        // Read only region end

        int r,sum=0;
            while(input1>0){
                r=input1%10;
                if(r%2==0){
                sum=sum+r;
                }
                input1=input1/10;
            }

        return sum;
    }
}
```

----

(20)----->ODD DIGIT'S SUM:

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~`

INPUT:9625

OUTPUT:14

SOLUTION:

```
import java.io.*;
import  java.util.*;
```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

```
// Read only region start
class UserMainCode
{

    public int OddDigitsSum(int input1){
        // Read only region end

        int r,sum=0;
            while(input1>0){
                r=input1%10;
                if(r%2==1){
                sum=sum+r;
                }
                input1=input1/10;
            }

        return sum;
    }
}
```

21.digitSum opt: sum of even or odd digits
https://tests.mettl.com/authenticateKey/a05abbcf

if argument2 is odd we have to add odd numbers in given input1

if it is even we have to add even numbers in given input1.

code:

```
import java.io.*;
import  java.util.*;

// Read only region start
class UserMainCode
{

    public int EvenOddDigitsSum(int input1,String input2){
        // Read only region end
        // Write code here...
        if(input2.equals("odd"))
        {
            int sum=0;
        while(input1>0)
           {
            int r=input1%10;
            if(r%2==1)
```

```
    {
    sum+=r;
    }
    input1/=10;


 }
return sum;
}
 else
 {
    int sum=0;
 while(input1>0)
   {
    int r=input1%10;
    if(r%2==0)
    {
    sum+=r;
    }
    input1/=10;


 }
return sum;
```

```
        }
    }
}
```

22.Is Palindrome Number?
https://tests.mettl.com/authenticateKey/28c41d9d

ex: 12321

if given number is palindrome return 2 else return 1;

```
int isPalinNum(int input1)
{
    // Read only region end
    // Write code here



    int temp=input1;
    int rev=0;
    while(input1>0)
    {
        rev=rev*10+input1%10;
        input1/=10;
    }
    if(rev!=temp)
        return 1;
```

```
    return 2;


}
```

23.Is Palindrome Possible? https://tests.mettl.com/authenticateKey/f4fdb02

if given number is 21251 it is possible to form palindrome by rearranging its digits as 21512 or 12521 so it should return 2

if given number is 2125 it is not possible to form palindrome by rearranging its digits so it should return 1

code:

```
int isPalinNumPossible(int input1)
{
   int h1[26]={0};
   int i;
      while(input1>0)
      {
         h1[input1%10]++;
         input1/=10;


      }
```

```
    int odd=0;
    for(i=0;i<10;i++)
    {
       if(h1[i]&1)
          odd++;
       if(odd>1)
          return 1;
    }
    return 2;

}
```

24.Create PIN using alpha, beta, gamma
https://tests.mettl.com/authenticateKey/be582d9f

ex-1

input1=123

input2=582

input3=175

then PIN=8122

ex-2

input1=190

input2=267

input3=853

then PIN=9150

PIN should be 4 digit

units digit=least of units position of three input numbers

tens digit=least of tens position of three input numbers

hundreds digit=least of hundreds position of three input numbers

thousands digit=maximum of all the digits in three input numbers.

code:

```
import java.io.*;
import  java.util.*;

// Read only region start
class UserMainCode
{

    public int createPIN(int input1,int input2,int input3){
        // Read only region end
        // Write code here...
        int u1=input1%10,u2=input2%10,u3=input3%10;
        int t1=(input1/10)%10,t2=(input2/10)%10,t3=(input3/10)%10;
        int h1=input1/100,h2=input2/100,h3=input3/100;
        int u=Math.min(u1,Math.min(u2,u3));
```

```
        int t=Math.min(t1,Math.min(t2,t3));
        int h=Math.min(h1,Math.min(h2,h3));
        int
th=Math.max(u1,Math.max(u2,Math.max(u3,Math.max(t1,Math.max(t2,M
ath.max(t3,Math.max(h1,Math.max(h2,h3)))))))));
        int num=th*1000+h*100+t*10+u;
                return num;


    }
}
```

25.Weight of a hill pattern
https://tests.mettl.com/authenticateKey/d612c0e6

pattern will be given we have to find weight of the pattern

code:

```
int totalHillWeight(int input1,int input2,int input3)
{
    // Read only region end
    // Write code here
        int sum=0,i,j;
        for(i=0;i<input1;i++)
        {
```

```
        for(j=0;j<=i;j++)
            sum+=input2;
        input2=input2+input3;
        //weight=input2+input3;
      }


   return sum;
}
```

26.Return second word in Uppercase
https://tests.mettl.com/authenticateKey/4a72723f

wipro technologies bangalore

o/p:TECHNOLOGIES

```
public class UserMainCode
{
   public string secondWordUpperCase(string input1)
   {


     String s[]=input1.split(" ");
      if(s.length==1)
```

```
            return "LESS";
        String s1=s[1];
        s1=s1.toUpperCase();
        return s1;
}
}
```

27.is Palindrome (string) https://tests.mettl.com/authenticateKey/ffe8042

Madam, madam,madaM, all are palindromes

if palindrome return 2

else return 1

code:

```
import java.io.*;
import  java.util.*;

// Read only region start
class UserMainCode
{

    public int isPalindrome(String input1){
```

```
        // Read only region end
        // Write code here...
        input1=input1.toLowerCase();
        int i,flag=1;
        for(i=0;i<input1.length()/2;i++)
        {
           if(input1.charAt(i)!=input1.charAt(input1.length()-i-1))
           {
              flag=0;
              break;
           }
        }
        if(flag==0)
           return 1;
        return 2;

    }
}
```

28.weight of string https://tests.mettl.com/authenticateKey/387952fc

if input2 is 0 we have to neglect vowels and find weight of input1

if input2 is 1 we have to consider all the alphabets of input1

ex-

input1=wipro

input2=0

sum=23+16+18=57

input1=wipro

input2=1

sum=23+9+16+18+15=81

code:

```
import java.io.*;
import  java.util.*;

// Read only region start
class UserMainCode
{

    public int weightOfString(String input1,int input2){
        // Read only region end
```

```
// Write code here...

String small="abcdefghijklmnopqrstuvwxyz";
int sum=0,i;
for(i=0;i<input1.length();i++)

{
if(input2==0)
{
   char c=input1.charAt(i);
   if(Character.isUpperCase(c))
      c=Character.toLowerCase(c);
   if(c!='a'&&c!='e'&&c!='i'&&c!='o'&&c!='u')
   {
   int index=small.indexOf(c);
   if(index>=0)
      sum+=index+1;
   }
   else
      sum+=0;

}
   else
   {
```

```
            char c=input1.charAt(i);
        if(Character.isUpperCase(c))
            c=Character.toLowerCase(c);
            int index=small.indexOf(c);
        if(index>=0)
            sum+=index+1;
            else
                sum+=0;
        }
}
      return sum;
   }
}
```

29.Most Frequent Digit https://tests.mettl.com/authenticateKey/916310b8

input1=123

input2=223

input3=412

input4=498

1 occurs 2 times

2 occurs 4 times

3 occurs 2 times

4 occurs 2 times

8 occurs 1time

9 occurs 1 time

so output should be 2 as it occurs maximum number of times

if 2 digits are occuring same number of times then the maximum number should be the answer.

code:

```
int MostFrequentDigit(int input1,int input2,int input3,int input4)
{
    // Read only region end
    // Write code here
    int h[10]={0};
    int i;
    if(input1==0&&input2==0&&input3==0&&input4==0)
        return 0;
    if(input1==0)
        h[0]++;
    if(input2==0)
        h[0]++;
    if(input3==0)
        h[0]++;
```

```
if(input4==0)
   h[0]++;
while(input1>0)
{
   h[input1%10]++;
   input1/=10;
}
while(input2>0)
{
   h[input2%10]++;
   input2/=10;
}
while(input3>0)
{
   h[input3%10]++;
   input3/=10;
}
while(input4>0)
{
   h[input4%10]++;
   input4/=10;
}
int index,max=-1;
for(i=0;i<10;i++)
```

```
    {
        if(max<=h[i])
        {
            max=h[i];
            index=i;
        }
    }
    return index;

}
```

30.FindStringCode https://tests.mettl.com/authenticateKey/e4df74e5

world wide web world=[23-4]+[15-12]+18=19+3+18=40

wide=[23-5]+[9-4]=18+5=23

web=26

output 402326

code:

```
import java.io.*;
import  java.util.*;
```

```
// Read only region start
class UserMainCode
{

    public int findStringCode(String input1){
        // Read only region end
        // Write code here...
        int sum=0,sum1=0;
        char c1,c2;
        int i1,i2,i,j;
        String small=new String("abcdefghijklmnopqrstuvwxyz");
        String cap=new String("ABCDEFGHIJKLMNOPQRSTUVWXYZ");
        String s[]=input1.split(" ");
        String res=new String("");
        for(i=0;i<s.length;i++)
            System.out.println(s[i]);
        for(i=0;i<s.length;i++)
        {
            System.out.println(s[i]);
            if(s[i].length()%2==0)
            {

                for(j=0;j<s[i].length()/2;j++)
            {
```

```
    c1=s[i].charAt(j);
    c2=s[i].charAt(s[i].length()-j-1);
    System.out.println(c1+" "+c2);
    if(Character.isLowerCase(c1))
    i1=small.indexOf(c1)+1;
    else
       i1=cap.indexOf(c1)+1;
    System.out.println(i1);
    if(Character.isLowerCase(c2))
       i2=small.indexOf(c2)+1;
       else
          i2=cap.indexOf(c2)+1;
       System.out.println(i2);

       sum=i1-i2;
       sum1+=Math.abs(sum);



}


   }
       else
       {
```

```
for(j=0;j<s[i].length()/2;j++)
{
c1=s[i].charAt(j);
c2=s[i].charAt(s[i].length()-j-1);
//System.out.println(c1+" "+c2);
if(Character.isLowerCase(c1))
i1=small.indexOf(c1)+1;
else
   i1=cap.indexOf(c1)+1;

if(Character.isLowerCase(c2))
   i2=small.indexOf(c2)+1;
   else
      i2=cap.indexOf(c2)+1;
   System.out.println(i2);

   sum=i1-i2;
   sum1+=Math.abs(sum);

   }
   char c3=s[i].charAt(s[i].length()/2);
   //System.out.println(c3);
   if(Character.isLowerCase(c3))
```

```
                sum1+=small.indexOf(c3)+1;
                else
                   sum1+=cap.indexOf(c3)+1;




            }
        System.out.println(sum1);
        String s1=String.valueOf(sum1);
        res+=s1;
        sum1=0;

    }
      System.out.println(res);
      int r=Integer.parseInt(res);
      return r;

    }
}
```

31.*****GET CODE THROUGH STRINGS*****

```
import java.io.*;
import  java.util.*;

// Read only region start
class UserMainCode
{

        public int getCodeThroughStrings(String input1){
                // Read only region end
                // Write code here...
          String ar[]=input1.split(" ");
                int tot=0,len=0;
          for(int i=0;i<ar.length;i++){
                  len+=ar[i].length();
          }
          int sum=0;
          while(len>10){
                  tot=len;
                  sum=0;
                  while(tot>0){
                          sum+=tot%10;
                          tot/=10;
                  }
                  len=sum;
          }
                return len;
        }
```

32.***String addition***

```
import java.util.Scanner;
public class prob {
    static String addString(String input1,String input2) {
        int a,b,carry=0,sum=0,mark=0,j=0;
        String ans="";
        StringBuilder s1=new StringBuilder();
        if(input1.length()>input2.length()){
            mark=0;
            j=input2.length()-1;
            for(int i=input1.length()-1;i>=0;i--) {
                a=input1.charAt(i)-48;
                if(mark!=input2.length()) {
                    b=input2.charAt(j)-48;
                    j--;
                    mark++;
                }
                else b=0;
                sum=a+b+carry;
                if(sum>10) {
                    carry=sum/10;
                    sum=sum%10;
                }
                else{
                  carry=0;}
                ans=ans+sum;
            }
        }
```

```
else {
    mark=0;
    j=input1.length()-1;
    for(int i=input2.length()-1;i>=0;i--) {
        a=input2.charAt(i)-48;
        if(mark!=input1.length()) {
            b=input1.charAt(j)-48;
            j--;
            mark++;
        }
        else b=0;
        sum=a+b+carry;
        if(sum>10) {
            carry=sum/10;
            sum=sum%10;
        }
        else{
          carry=0;}
        ans=ans+sum;
    }

}
s1.append(ans);
s1=s1.reverse();
String s2="";
for(int i=0;i<s1.length();i++) {
     if(s1.charAt(i)!='0') {
                s2=s2+String.valueOf(s1.charAt(i));
```

```
        }
      }


   return String.valueOf(s2);
   }
}
```

33.Simple Enocoded Array

```
import java.io.*;
import  java.util.*;

// Read only region start
class UserMainCode
{

        public class Result{
                public final int output1;
                public final int output2;

                public Result(int out1, int out2){
                        output1 = out1;
                        output2 = out2;
                }
        }

   public Result findOriginalFirstAndSum(int[] input1,int input2){
        // Read only region end
```

```
        //Write code here...
                    int sum=input1[input1.length-1];
       for(int i=input1.length-2;i>=0;i--){
                      input1[i]=input1[i]-input1[i+1];
                      sum+=input1[i];
                }
                    Result r=new Result(input1[0],sum);
                    return r;
    }
}
```

34.***DECREASING SEQUENCE*********

```
import java.io.*;
import  java.util.*;

// Read only region start
class UserMainCode
{

            public class Result{
                    public final int output1;
                    public final int output2;

                    public Result(int out1, int out2){
                              output1 = out1;
                              output2 = out2;
```

```
                }
        }

public Result decreasingSeq(int[] input1,int input2){
        // Read only region end
    //Write code here...
          int[] ar=input1.clone();
                Arrays.sort(ar);
                if(Arrays.equals(ar,input1)) return new Result(0,0);
                if(input1.length==1) return new Result(0,0);
    int temp=0,subs=0,max=0,count=0;
    for(int i=0;i<input1.length-1;i++){
       if(input1[i]>input1[i+1]){
          temp=i;
          count=0;
          while(temp<input1.length-1){
             if(input1[temp]>input1[temp+1]){
             temp++;
             count++;
             }
             else {
                subs+=1;
                break;
             }
          }
       i=temp+1;}
       if(temp==input1.length-1) subs+=1;
       if(max<count+1) max=count+1;
```

```
        }
                return new Result(subs,max);
    }
}
```

35. ********MOST FREQUENTLY OCCURING*****

```
import java.io.*;
import  java.util.*;

// Read only region start
class UserMainCode
{

        public int mostFrequentlyOccurringDigit(int[] input1,int input2){
                // Read only region end
                // Write code here...
                int[] ar=new int[10];
                int temp=0,max=0,num=0;
                for(int i=0;i<input1.length;i++){
                        temp=input1[i];
                        while(temp>0){
                                ar[temp%10]+=1;
                                temp=temp/10;
                        }
                }
                for(int j=0;j<ar.length;j++){
                        if(ar[j]>max){
                                max=ar[j];
                                num=j;
```

```
                        }
                        if(ar[j]==max){
                                if(j>num){
                                        num=j;
                                        max=ar[j];
                                }
                        }
                }
        return num;
        }
}
```

36. ********SUM OF POWER OF DIGITS*******

```
import java.io.*;
import  java.util.*;
import java.lang.Math.*;
// Read only region start
class UserMainCode
{

        public int sumOfPowerOfDigits(int input1){
                // Read only region end
                // Write code here...
                Integer sum=0,r=0,prev=0;
                Double f1,f2;
                while(input1>0){
                        r=Integer.valueOf(input1%10);
                        f1=Double.valueOf(r);
                        f2=Double.valueOf(prev);
```

```
                        f1=Math.pow(f1,f2);
                        sum+=f1.intValue();
                        prev=Integer.valueOf(r);
                        input1/=10;
                }
        return sum;
        }
}
```

37.*****SUM OF SUMS OF DIGITS*****

```
import java.io.*;
import  java.util.*;

// Read only region start
class UserMainCode
{

        public int sumOfSumsOfDigits(int input1){
                // Read only region end
                // Write code here...
                int last=0,current=0,r=0,sum=0;
                while(input1>0){
                        r=input1%10;
                        current=r+last;
                        input1/=10;
                        sum=sum+current;
                        last=last+r;
                }
        return sum;
```

```
}
}
```

38. ******IDENTIFY POSSIBLE WORDS*****

```
import java.io.*;
import  java.util.*;

// Read only region start
class UserMainCode
{

    public String identifyPossibleWords(String input1,String input2){
        // Read only region end
        // Write code here...
        String[] ar=input2.split(":");
        String temp="",fin="";
        int count=0;
        for(int i=0;i<ar.length;i++){
            temp=ar[i];
            count=0;
            if(temp.length()==input1.length()){
                for(int j=0;j<temp.length();j++){
                    if(input1.charAt(j)!='_'){

    if(Character.toUpperCase(input1.charAt(j))==Character.toUpperCase(temp.charAt(j))){
                                count++;
                            }
                    }
                }
                if(count==temp.length()-1) fin=fin+temp.toUpperCase()+":";
```

```
                }
            }
            if(fin=="") return "ERROR-009";
            return fin.substring(0,fin.length()-1);
        }
}
```

39.****ENCODED 3 STRINGS*****

```
import java.io.*;
import  java.util.*;

// Read only region start
class UserMainCode
{

    public class Result{
        public final String output1;
        public final String output2;
        public final String output3;

        public Result(String out1, String out2, String out3){
            output1 = out1;
            output2 = out2;
            output3 = out3;
        }
    }
```

```
    public Result encodeThreeStrings(String input1,String input2,String input3){
            // Read only region end
        //Write code here...
String f1="",f2="",f3="",m1="",m2="",m3="",l1="",l2="",l3="";
        String out1="",out2="",out3="";
        int d=0;
        //task1
        //input1
        if(input1.length()%3==0){
            d=input1.length()/3;
            f1=input1.substring(0,d);
            m1=input1.substring(d,2*d);
            l1=input1.substring(2*d);
        }
        else if(input1.length()%3==1){
            d=input1.length()/3;
            f1=input1.substring(0,d);
            m1=input1.substring(d,2*d+1);
            l1=input1.substring((2*d)+1);

        }
        else{
            d=input1.length()/3;
            f1=input1.substring(0,d+1);
            m1=input1.substring(d+1,2*d+1);
            l1=input1.substring(2*d+1);
        }
        //input2
```

```
if(input2.length()%3==0){
    d=input2.length()/3;
    f2=input2.substring(0,d);
    m2=input2.substring(d,2*d);
    l2=input2.substring(2*d);
}
else if(input2.length()%3==1){
    d=input2.length()/3;
    f2=input2.substring(0,d);
    m2=input2.substring(d,2*d+1);
    l2=input2.substring((2*d)+1);

}
else{
    d=input2.length()/3;
    f2=input2.substring(0,d+1);
    m2=input2.substring(d+1,2*d+1);
    l2=input2.substring(2*d+1);
}
//input3
if(input3.length()%3==0){
    d=input3.length()/3;
    f3=input3.substring(0,d);
    m3=input3.substring(d,2*d);
    l3=input3.substring(2*d);
}
else if(input3.length()%3==1){
    d=input3.length()/3;
```

```
            f3=input3.substring(0,d);
            m3=input3.substring(d,2*d+1);
            l3=input3.substring((2*d)+1);

        }
        else{
            d=input3.length()/3;
            f3=input3.substring(0,d+1);
            m3=input3.substring(d+1,2*d+1);
            l3=input3.substring(2*d+1);
        }
        out1=f1+f2+f3;
        out2=m1+m2+m3;
        out3=l1+l2+l3;
        //task2
        String out3_="";
        for(int k=0;k<out3.length();k++){
            if(Character.isUpperCase(out3.charAt(k))){
                out3_=out3_+String.valueOf(Character.toLowerCase(out3.charAt(k)));
            }
            else{
                out3_=out3_+String.valueOf(Character.toUpperCase(out3.charAt(k)));
            }
        }
        return new Result(out1,out2,out3_);
    }
}
```

//40.Generate series and find Nth element

```
import java.io.*;
import  java.util.*;

// Read only region start
class UserMainCode
{

        public int seriesN(int input1,int input2,int input3,int input4){
                // Read only region end
                // Write code here...
                int i=3,diff=0,next=0;
                while(i<input4){
                        diff=input2-input1;
                        next=input3+diff;
                        input1=input2;
                        input2=input3;
                        input3=next;
                        i++;
                }
                return next;
        }
}
```

//// 41.Find result after alternate add_sub on N

```
using System;
using System.Collections.Generic;
```

```
//Read only region start
public class UserMainCode
{
    public int AddSub(int input1,int input2)
    {
        //Read only region end
        //Write code here
                    int glob=0;


        if(input2==1){


                            for(int i=0;i<=input1;i++){
                                    if(i%2==0){
                                            glob=glob+(input1-i);
                                    }
                                    else glob=glob-(input1-i);
                            }
                    }
                    else{
                            for(int i=0;i<=input1;i++){
                                    if(i%2==0 && i!=0){
                                            glob=glob-(input1-i);
                                    }
                                    else glob=glob+(input1-i);
                            }
                    }
            return glob;
```

```
            }
    }
//42.find password

import java.io.*;
import  java.util.*;

// Read only region start
class UserMainCode
{

    public int findPassword(int input1,int input2,int input3,int input4,int input5){
        // Read only region end
        // Write code here...
        int[] h1=new int[10];
        int[] h2=new int[10];
        int[] h3=new int[10];
        int[] h4=new int[10];
        int[] h5=new int[10];
        int t1=input1,t2=input2,t3=input3,t4=input4,t5=input5;
        int stable_sum=0,unstable_sum=0,i;

        while(input1>0)
        {
            h1[input1%10]++;
            input1/=10;
        }
        while(input2>0)
```

```
{
   h2[input2%10]++;
   input2/=10;
}
while(input3>0)
{
   h3[input3%10]++;
   input3/=10;
}
while(input4>0)
{
   h4[input4%10]++;
   input4/=10;
}
while(input5>0)
{
   h5[input5%10]++;
   input5/=10;
}
for(i=0;i<10;i++)
{
   System.out.println(h1[i]+" "+h2[i]+" "+h3[i]+" "+h4[i]+" "+h5[i]);
   //System.out.print(" ");
}
int c=0;
for(i=0;i<10;i++)
{
   if(h1[i]!=0)
```

```
    {
        c=h1[i];
        break;
    }
}
//System.out.print(c);
for(i=0;i<10;i++)
{
    if(h1[i]!=0)
    {
        if(c!=h1[i])
        {
        unstable_sum+=t1;
        break;
         }
      }

}
//System.out.print(unstable_sum);
if(i==10)
    stable_sum+=t1;
for(i=0;i<10;i++)
{
    if(h2[i]!=0)
    {
        c=h2[i];
        break;
    }
```

```
 }
 for(i=0;i<10;i++)
 {
    if(h2[i]!=0)
    {
       if(c!=h2[i])
       {
       unstable_sum+=t2;
       break;
        }
      }

}
 if(i==10)
    stable_sum+=t2;
 for(i=0;i<10;i++)
 {
    if(h3[i]!=0)
    {
       c=h3[i];
       break;
    }
 }
 for(i=0;i<10;i++)
 {
    if(h3[i]!=0)
    {
       if(c!=h3[i])
```

```
        {
        unstable_sum+=t3;
        break;
         }
       }

}
 if(i==10)
    stable_sum+=t3;
 for(i=0;i<10;i++)
 {
    if(h4[i]!=0)
    {
       c=h4[i];
       break;
    }
 }
 for(i=0;i<10;i++)
 {
    if(h4[i]!=0)
    {
       if(c!=h4[i])
       {
       unstable_sum+=t4;
       break;
        }
      }
```

```
}
 if(i==10)
    stable_sum+=t4;
 for(i=0;i<10;i++)
 {
    if(h5[i]!=0)
    {
       c=h5[i];
       break;
    }
 }
 for(i=0;i<10;i++)
 {
    if(h5[i]!=0)
    {
       if(c!=h5[i])
       {
       unstable_sum+=t5;
       break;
        }
      }

}
 if(i==10)
    stable_sum+=t5;
 System.out.print(stable_sum);
 System.out.print(unstable_sum);
 return stable_sum-unstable_sum;
```

```
    }
}
```

//43.Calculate sum of non-prime index values

```
import java.io.*;
import  java.util.*;

// Read only region start
class UserMainCode
{

    public int sumOfNonPrimeIndexValues(int[] input1,int input2){
        // Read only region end
        // Write code here...
        int sum=input1[0]+input1[1];
        int i,j,flag;
        for(i=3;i<input2;i++)
        {
            flag=1;
            for(j=2;j<=Math.sqrt(i);j++)
            {
                if(i%j==0)
                {
                    flag=0;
                    break;
                }
            }
            System.out.println(flag);
            if(flag==0)
```

```
            sum+=input1[i];
        }
        return sum;

    }
}
```

//44 find digit to be removed to form palindrome

```
import java.io.*;
import  java.util.*;

// Read only region start
class UserMainCode
{

    public int digitRemove_Palin(int input1){
        // Read only region end
        // Write code here...
        int[] h=new int[10];
        int t=input1;
        int r,rev=0;
        while(input1>0)
        {
            r=input1%10;
            rev=rev*10+r;
            input1/=10;
        }
```

```
        if(rev==t)
            return -1;
        input1=t;
        while(input1>0)
        {
            h[input1%10]++;
            input1/=10;
        }
        //String s=String.valueOf(input1);

        int index=-1,i;
        for(i=0;i<10;i++)
        {
            if(h[i]%2==1)
            {
            index=i;
            }
        }
        System.out.print(index);
        return index;

    }
}
```

//45.The “Nambiar Number” Generator

```
class UserMainCode
{

	public int nnGenerator(String input1){
		// Read only region end
		String s=input1;
		int len=s.length();
		int a[]=new int[len];
		for(int i=0 ;i <len ;i++)
		{
			a[i]=(s.charAt(i)-'0');
		}
		System.out.println(Arrays.toString(a));
		int i=0;
		String temp="";
		int k=a[i];
		int evenflag,oddflag;
		if(k%2==0)
		{
			evenflag=1;
			oddflag=0;
		}
		else
		{
			evenflag=0;
			oddflag=1;
		}
```

```
while(i<len)
{
        if(i==len-1)
        {
                System.out.print(k);
                temp+=k;
                break;
        }
        if((k%2!=0)&&(oddflag==1))
        {
                k+=a[i+1];
                i++;
        }
  else if((k%2==0)&&(evenflag==1))
        {
                k+=a[i+1];
                i++;
        }
        else
        {
                System.out.print(k+" ");
                temp+=k;
                i=i+1;
                k=a[i];
                if(k%2==0)
       {
                evenflag=1;
                  oddflag=0;
```

```
                    }
                else
                    {
                        evenflag=0;
                        oddflag=1;
                    }
                        }

                }
                return Integer.parseInt(temp);
        }
}
```

////46.User Id Generation

```
class UserMainCode
{

    public String userIdGeneration(String input1,String

input2,int input3,int input4){
        // Read only region end
        // Write code here...

        int s1=input1.length();
        int s2=input2.length();
        String longer="";
        String smaller="";
        String output1="";
```

```
if(s1==s2)
{
if(input1.compareTo(input2)>0)
{
   longer=input1;
   smaller=input2;
}
   else
   {
      longer=input2;
      smaller=input1;
   }
}

if(s1>s2){
   longer=input1;
   smaller=input2;
}
else if(s1<s2)
{
   longer=input2;
   smaller=input1;
}
String pin=input3+"";
String output=smaller.charAt(0)+longer+pin.charAt

(input4-1)+pin.charAt(pin.length()-input4);
for(int i=0;i<output.length();i++)
```

```
        {
            if(Character.isLowerCase(output.charAt
(i)))
            {
                output1+=Character.toUpperCase
(output.charAt(i));
            }
            else
            {
                output1+=Character.toLowerCase
(output.charAt(i));
            }
        }
        return output1;
        }
    }
```

////47.Message controlled Robot movement

```
import java.io.*;
import  java.util.*;

// Read only region start
class UserMainCode
```

```
{

    public String moveRobot(int input1,int input2,String input3,String input4){
        // Read only region end
    //Read only region end
    //Write code here
    String path[]=input3.split("-");
        int x=Integer.parseInt(path[0]);
        int y=Integer.parseInt(path[1]);
        String pos=path[2];
        String arr[]=input4.split(" ");
        int f=0;
        for(String s:arr)
        {
            if(s.equals("R"))
            {
                if(pos.equals("N"))
                    pos="E";
                else if(pos.equals("E"))
                    pos="S";
                else if(pos.equals("S"))
                    pos="W";
                else
                    pos="N";
            }
            else if(s.equals("L"))
            {
                if(pos.equals("N"))
```

```
                pos="W";
        else if(pos.equals("E"))
                pos="N";
        else if(pos.equals("S"))
                pos="E";
        else
                pos="S";
}
else if(f!=1)
{
        if(pos.equals("N"))
        {
                if(input2>y)
                        y=y+1;
                else
                        f=1;
        }
        else if(pos.equals("S"))
        {
                if(y>0)
                        y=y-1;
                else
                        f=1;
        }
        else if(pos.equals("E"))
        {
                if(input1>x)
                        x=x+1;
```

```
                                        else
                                                f=1;
                                }
                                else
                                {
                                        if(x>0)
                                                x=x-1;
                                        else
                                                f=1;
                                }
                        }
                }
                if(f!=1)
                return String.valueOf(x)+"-"+String.valueOf(y)+"-"+String.valueOf(pos);
                else
                        return String.valueOf(x)+"-"+String.valueOf(y)+"-"+String.valueOf(pos)+"-
"+"ER";


    }
}
```