81) String Processing - V

Write a program to read a string array, concatenate the array elements one by one separated by comma and return the final string as output.
Include a class UserMainCode with a static method **concatString** which accepts the string array. The return type is the string.
Create a Class Main which would be used to accept the string array and call the static method present in UserMainCode.

Input and Output Format:
Input consists of an integer n which is the number of elements followed by n string values.
Output consists of the string.
Refer sample output for formatting specifications.

Sample Input 1:
3
AAA
BBB
CCC
Sample Output 1:
AAA,BBB,CCC

```
public class Main {
      public static void main(String[] args) {
            Scanner sc=new Scanner(System.in);
            int n=sc.nextInt();
            String[] s=new String[n];
            for(int i=0;i<n;i++)
                  s[i]=sc.next();
            System.out.println(User.concatString(s));
      }
}


public class User {
      public static String concatString (String s[])
      {
    StringBuffer sb=new StringBuffer();
    sb.append(s[0]);
    for(int i=1;i<s.length;i++)
    {
      sb.append(",");
      sb.append(s[i]);
    }
    return sb.toString();
                                      }
}
```

82) Unique Number

GIven three integers (a,b,c) , Write a program that returns the number of unique integers among the three.
Include a class UserMainCode with a static method **calculateUnique** which accepts three integers and returns the count as integer.
Create a Class Main which would be used to accept three integers and call the static method present in UserMainCode.

Input and Output Format:
Input consists of three integers.
Output consists of a integer.
Refer sample output for formatting specifications.

Sample Input 1:
12
4
3
Sample Output 1:
3

Sample Input 2:
4
-4
4
Sample Output 2:
2

```
public class Main {
      public static void main(String[] args) {
            Scanner sc=new Scanner(System.in);
            int a=sc.nextInt();
            int b=sc.nextInt();
            int c=sc.nextInt();
            System.out.println(User.calculateUnique(a,b,c));
      }
}

public class User {
      public static int calculateUnique(int a,int b,int c)
      {
    int count=0;
    int[] s={a,b,c};
    int[] res=new int[3];
    for(int i=0;i<s.length;i++)
    {
        res[i]=Math.abs(s[i]);
    }
    count=0;
    for(int i=0;i<res.length-1;i++)
    {

      if(res[i]==res[i+1])
      {
            count++;
      }
    }
    return count+1;
      }
}
```

```
public class Main {

      public static void main(String[] args) {
       int ct1=0;
            Scanner sc=new Scanner(System.in);
            int a=sc.nextInt();
            int b=sc.nextInt();
            int c=sc.nextInt();

                if(a!=b)
                  ct1=ct1+1;

                if(a!=c)
                ct1=ct1+1;

                if(b!=c)
                ct1=ct1+1;

      if((a==b) & (b==c))
            System.out.println("output "+(ct1+1));
      else

            System.out.println("output "+ct1);

      }
}
```

```
public class Main {


      public static void main(String[] args) {



            Scanner sc=new Scanner(System.in);
            int a=sc.nextInt();
            int b=sc.nextInt();
            int c=sc.nextInt();

                  int d=0;
            if(a!=b&&a!=c)
            {
                  d=3;
            }
            else if(a==b&&a==c)
            {
                  d=1;
            }
            else if(a!=b&&a==c)

            {
                  d=2;
            }
            else if(a==b&&a!=c)
            {
                  d=2;
            }

            System.out.println(d);
      }
                                       }
```

83) Math Calculator

Write a program that accepts three inputs, first two inputs are operands in int form and third one being one of the following five operators: +, -, *, /, %. Implement calculator logic and return the result of the given inputs as per the operator provided. In case of division, Assume the result would be integer.
Include a class UserMainCode with a static method **calculator** which accepts two integers, one operand and returns the integer.
Create a Class Main which would be used to accept three integers and call the static method present in UserMainCode.

Input and Output Format:
Input consists of two integers and a character.
Output consists of a integer.
Refer sample output for formatting specifications.

Sample Input 1:
23
2
*
Sample Output 1:
46

```
public class Main {
      public static void main(String[] args) {
            Scanner sc=new Scanner(System.in);
            int a=sc.nextInt();
            int b=sc.nextInt();
            char op=sc.next().charAt(0);
            System.out.println(User.calculateUnique(a,b,op));
      }
}
```

```
public class User {
      public static int calculateUnique(int a,int b,char op)
      {
    int res=0;
    switch(op){

    case '+':
      res=a+b;
    case '-':
      res=Math.abs(a-b);
    case '*':
      res=a*b;
    case '/':
      res=Math.round(a/b);
    case '%':
      res=Math.round(a%b);
    }
    return res;
      }
}
```

84) Scores

Write a program to read a integer array of scores, if 100 appears at two consecutive locations return true else return false.
Include a class UserMainCode with a static method **checkScores** which accepts the integer array. The return type is boolean.
Create a Class Main which would be used to accept the integer array and call the static method present in UserMainCode.

Input and Output Format:
Input consists of an integer n which is the number of elements followed by n integer values.
Output consists of a string that is either 'TRUE' or 'FALSE'.
Refer sample output for formatting specifications.

Sample Input 1:
3
1
100
100
Sample Output 1:
TRUE

Sample Input 2:
3
100
1
100
Sample Output 2:
FALSE

```java
public class User {
      public static boolean scanArray(int s[])
      {
   boolean b=false;

         for(int i=0;i<s.length-1;i++)
         {
               if(s[i]==100&&s[i+1]==100)
               {
                     b=true;
               break;
               }
               else
                     b=false;
         }


   return b;
      }
}
```

85) ArrayFront

Write a program to read a integer array and return true if one of the first 4 elements in the array is 9 else return false.
Note: The array length may be less than 4.

Include a class UserMainCode with a static method **scanArray** which accepts the integer array. The return type is true / false.
Create a Class Main which would be used to accept the integer array and call the static method present in UserMainCode.

Input and Output Format:
Input consists of an integer n which is the number of elements followed by n integer values.
Output consists of TRUE / FALSE.
Refer sample output for formatting specifications.

Sample Input 1:
6
1
2
3
4
5
6
Sample Output 1:
FALSE

Sample Input 2:
3
1
2
9
Sample Output 2:
TRUE

```
public class Main {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int[] s= new int[n];
        for(int i=0;i<n;i++)
            s[i]=sc.nextInt();
        boolean b=User.scanArray (s);
        System.out.println(b);
    }
}
```

```
public class User {
    public static boolean scanArray(int s[])
    {
  boolean b=false;
  if(s.length>4)
```

```
   {
         for(int i=0;i<4;i++)
         {
               if(s[i]==9)
                     b=true;
               else
                     b=false;
         }
   }
   else
   {
         for(int i=0;i<s.length;i++)
         {
               if(s[i]==9)
                     b=true;
               else
                     b=false;
         }
   }
   return b;
      }
}
```

86) Word Count

Given a string array (s) and non negative integer (n) and return the number of elements in the array which have same number of characters as the givent int N.
Include a class UserMainCode with a static method **countWord** which accepts the string array and integer. The return type is the string formed based on rules.
Create a Class Main which would be used to accept the string and integer and call the static method present in UserMainCode.

Input and Output Format:
Input consists of a an integer indicating the number of elements in the string array followed the elements and ended by the non-negative integer (N).
Output consists of a integer .
Refer sample output for formatting specifications.

Sample Input 1:
4
a
bb
b
ccc
1
Sample Output 1:
2

Sample Input 2:
5
dog
cat
monkey
bear
fox
3
Sample Output 2:
3

```
public class Main {
      public static void main(String[] args) {
            Scanner sc=new Scanner(System.in);
            int n=sc.nextInt();
            String[] s= new String[n];
            for(int i=0;i<n;i++)
                  s[i]=sc.next();
            int n1=sc.nextInt();
            System.out.println(User.countWord (s,n1));
      }
}
```

```
public class User {
      public static int countWord (String s[],int n1)
      {
   int count=0;
   for(int i=0;i<s.length;i++)
   {
         if(s[i].length()==n1)
               count++;
   }
   return count;
      }
}
```

87) Find Distance

Write a Program that accepts four int inputs(x1,y1,x2,y2) as the coordinates of two points. Calculate the distance between the two points using the below formula.
Formula : square root of((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2))
Then, Round the result to return an int
Include a class UserMainCode with a static method **findDistance** which accepts four integers. The return type is integer representing the formula.
Create a Class Main which would be used to accept the input integers and call the static method

present in UserMainCode.

Input and Output Format:
Input consists of four integers.
Output consists of a single integer.
Refer sample output for formatting specifications.

Sample Input 1:
3
4
5
2
Sample Output 1:
3

Sample Input 2:
3
1
5
2
Sample Output 2:
2

```
public class Main {
      public static void main(String[] args) {
            Scanner sc=new Scanner(System.in);
            int x1=sc.nextInt();
            int y1=sc.nextInt();
            int x2=sc.nextInt();
            int y2=sc.nextInt();
            System.out.println(User.findDistance(x1,y1,x2,y2));
      }
}


public class User {
      public static int findDistance(int x1,int y1,int x2,int y2)
      {
    double d=((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));
      int res=(int)Math.ceil(Math.sqrt(d));
      return res;
      }
}
```

88) Word Count - II

Write a program to read a string and count the number of words present in it.
Include a class UserMainCode with a static method **countWord** which accepts the string. The return type is the integer giving out the count of words.

Create a Class Main which would be used to accept the string and call the static method present in UserMainCode.

Input and Output Format:
Input consists of a string.
Output consists of integer.
Refer sample output for formatting specifications.

Sample Input 1:
Today is Sunday
Sample Output 1:
3

```java
public class Main {
        public static void main(String[] args) {
                Scanner sc=new Scanner(System.in);
                String s=sc.nextLine();
                System.out.println(User.countWord(s));
        }
}
```

```java
public class User {
        public static int countWord(String s)
        {
        StringTokenizer st=new StringTokenizer(s," ");
        int count =st.countTokens();
        return count;
        }
}
```

89) Sum of Max & Min

Write a Program that accepts three integers, and returns the sum of maximum and minimum numbers.
Include a class UserMainCode with a static method getSumMaxMin which accepts three integers.
The return type is integer representing the formula.
Create a Class Main which would be used to accept the input integers and call the static method present in UserMainCode.

Input and Output Format:
Input consists of three integers.
Output consists of a single integer.
Refer sample output for formatting specifications.

Sample Input 1:
12
17
19

Sample Output 1:
31

```
public class Main {
        public static void main(String[] args) {
                Scanner sc=new Scanner(System.in);
                int a=sc.nextInt();
                int b=sc.nextInt();
                int c=sc.nextInt();
                System.out.println(User.getSumMaxMin (a,b,c));
        }
}

public class User {
        public static int getSumMaxMin (int a,int b,int c)
        {
        int sum=0;
        int[] s={a,b,c};
        Arrays.sort(s);
        sum=s[0]+s[2];
        return sum;

        }
}
```

90) Decimal to Binary Conversion

Write a Program that accepts a decimal number n, and converts the number to binary.

Include a class UserMainCode with a static method **convertDecimalToBinary** which accepts an integer. The return type is long representing the binary number.

Create a Class Main which would be used to accept the input integer and call the static method present in UserMainCode.

Input and Output Format:
Input consists of single integer.
Output consists of a single long.
Refer sample output for formatting specifications.

Sample Input 1:

5

Sample Output 1:

101

```
public class Main {
        public static void main(String[] args) {
                Scanner sc=new Scanner(System.in);
                int n=sc.nextInt();
                System.out.println(User.convertDecimalToBinary(n));
        }
}

public class User {
        public static long convertDecimalToBinary(int n)
        {
        String x=   Integer.toBinaryString(n);
        long res= Integer.parseInt(x);


                long res=Integer.parseInt(Integer.toBinaryString(n));



        return res;
        }
}
```