Course: JAVA **Session:** Control and Looping **Timestamp:** 2021-8-31 10:16:53 **Register Number:** RA2031241010094

Q. Mobius Function

Java Program to determine Mobius function.
The MOBIUS function M(N) for a natural number N is defined as follows:
M(N) = 1 if N = 1 [Condition 1]

Source Code

```
import java.util.*;
public class TestClass {
  static int mobius(int n)
  {
    int p=0;
    if(n%2==0)
    {
      n=n/2;
      p++;
      if(n%2==0)
        return 0;
    }
    for(int i=3;i<=Math.sqrt(n);i=i+2)
    {
      if(n%i==0)
      {
        n=n/i;
        p++;
        if(n%i==0)
          return 0;
      }
    }
    return (p%2==0)?-1:1;
  }
  public static void main(String[] args) {
       Scanner sc=new Scanner(System.in);
       int N=sc.nextInt();
       System.out.println(mobius(N));

  }
}
```

Sample Input

78

Sample Output

-1

Result

Thus, Program " **Mobius Function** " has been successfully executed

Course: JAVA **Session:** Control and Looping **Timestamp:** 2021-8-31 10:17:16 **Register Number:** RA2031241010094

Q. Magician Numbers

This problem practices the addition of 2-digit numbers. Encourage the students to share the methods that they use to solve the problem. For example some students may use place value while others will find it easier to use a rounding method.
91 + 19

place value: 91 + 9 + 10
rounding: 91 + 20 1

This problem also offers the opportunity for students to "play" with numbers. As well as practising addition the students are encouraged to look for patterns in their answers. This play encourages students to increase their understanding of numbers and how they relate to one another. It also helps develop problem solving skill and creativity.

As numbers are 'reversed' they swap places. (eg. 41 to 14) It is therefore important to discuss what is happening to the place value of the numbers.

If an Integer N, write a program to reverse the given number.

Input
The first line contains an integer T, total number of testcases. Then follow T lines, each line contains an integer N.

Output
Display the reverse of the given number N

Source Code

```
import java.util.*;
public class TestClass {
  public static void main(String[] args) {
       Scanner sc=new Scanner(System.in);
       int j=sc.nextInt();
       int rev=0;
      for(int i=0;i<j;i++)
       {
         int a=sc.nextInt();
         while(a>0)
         {
           int digit=a%10;
           rev=rev*10+digit;
           a=a/10;
         }
         System.out.println(rev);
        rev=0;
      }
 }
}
```

Sample Input

```
3
12345
2512
9009
```

Sample Output

```
54321
2152
9009
```

Result

Thus, Program " **Magician Numbers** " has been successfully executed

Course: JAVA **Session:** Control and Looping **Timestamp:** 2021-8-31 10:17:40 **Register Number:** RA2031241010094

Q. Minimum Distances

Consider an array of n integers,A=[a subscript 0 ,a subscript 1,...,a subscript n-1] . The distance between two indices, i and j, is denoted by d subscript i,j = |i-j|.

Given A, find the minimum d subscript i,j such that a subscript i=a subscript j and i not equal to j. In other words, find the minimum distance between any pair of equal elements in the array. If no such value exists, print -1 .

Note: |a| denotes the absolute value of a .

Input Format

The first line contains an integer ,n , denoting the size of array A.

The second line contains n space-separated integers describing the respective elements in array A .

Constraints

1<=n<=10 power 3
1<=a subscript i <=10 power 5

Output Format

Print a single integer denoting the minimum d subscript i,j in A; if no such value exists, print -1.

Source Code

```
import java.util.*;
public class TestClass {
  public static void main(String[] args) {
  Scanner sc=new Scanner(System.in);
      int n=sc.nextInt();
      int[] a=new int[n];
      for(int i=0;i<n;i++)
        a[i]=sc.nextInt();
      int ans=Integer.MAX_VALUE;
      for(int i=0;i<n;i++)
        for(int j=i+1;j<n;j++)
          if(a[i]==a[j]&&j-i<ans)
            ans=j-i;
      System.out.println(ans==Integer.MAX_VALUE?-1:ans);
 }
}
```

Sample Input

```
6
7 1 3 4 1 7
```

Sample Output

```
3
```

Result

Thus, Program " **Minimum Distances** " has been successfully executed

Course: JAVA **Session:** Control and Looping **Timestamp:** 2021-8-31 10:18:15 **Register Number:** RA2031241010094

Q. To check two inputs are equal

Write a Java program that reads in two floating-point numbers and tests whether they are the same up to three decimal places.

Note:

Use Double Data Type

Source Code

```
import java.util.*;
public class TestClass {
  public static void main(String[] args) {
  Scanner sc=new Scanner(System.in);
       double a=sc.nextDouble();
       double b=sc.nextDouble();
       if(a==b)
         System.out.println("They are the same");
       else
         System.out.println("They are different");
 }
}
```

Sample Input

1256.3210
1256.3215

Sample Output

They are different

Result

Thus, Program " **To check two inputs are equal** " has been successfully executed

Course: JAVA **Session:** Control and Looping **Timestamp:** 2021-8-31 10:18:36 **Register Number:** RA2031241010094

Q. Julius Caesar

Julius Caesar protected his confidential information by encrypting it in a cipher. Caesars cipher rotated every letter in a string by a fixed number, "K", making it unreadable by his enemies. Given a string, "S", and a number, "K", encrypt "S" and print the resulting string.

Note: The cipher only encrypts letters; symbols, such as "-" remain unencrypted.

Input Format

The first line contains an integer, "N" , which is the length of the unencrypted string.

The second line contains the unencrypted string, "S".

The third line contains the integer encryption key, "K" , which is the number of letters to rotate.

Constraints:

1 <= N <=100
0<=K <= 100

S is a valid ASCII string and does not contain any spaces.

Output Format

For each test case, print the encoded string.

Explanation

Each unencrypted letter is replaced with the letter occurring "K" spaces after it when listed alphabetically. Think of the alphabet as being both case-sensitive and circular; if "K" rotates past the end of the alphabet, it loops back to the beginning (i.e.: the letter after "z" is "a", and the letter after is "Z" is "A").

Selected Examples:
"m" (ASCII 109) becomes "o" (ASCII 111).
"i" (ASCII 105) becomes "k" (ASCII 107).
"-" remains the same, as symbols are not encoded.
"O" (ASCII 79) becomes "Q" (ASCII 81).
"z" (ASCII 122) becomes "b" (ASCII 98); because "z" is the last letter of the alphabet, "a" (ASCII 97) is the next letter after it in lower-case rotation.

Source Code

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;
public class TestClass {
  public static void main(String[] args) {
      Scanner sc=new Scanner(System.in);
      int numchar=Integer.parseInt(sc.nextLine());
      char[] inputstr=sc.nextLine().toCharArray();
      int rotatevalue=sc.nextInt();
      for(int i=0;i<numchar;i++)
      {
        char currentchar=inputstr[i];
        if(Character.isLetter(currentchar))
        {
          char rotatedchar=(char)((int)currentchar+rotatevalue%26);
          if(Character.isUpperCase(currentchar))
          {
            inputstr[i]=((int)rotatedchar<=90)?rotatedchar:(char)((rotatedchar-(int)'Z')+(int)'A'-1);
          }
          else
          {
            inputstr[i]=((int)rotatedchar<=122)?rotatedchar:(char)((rotatedchar-(int)'z')+(int)'a'-1);
          }
        }
      }
      System.out.println(inputstr);

  }
}
```

Sample Input

```
11
middle-Outz
2
```

Sample Output

```
okffng-Qwvb
```

Result

Thus, Program " **Julius Caesar** " has been successfully executed

Course: JAVA **Session:** Control and Looping **Timestamp:** 2021-8-31 10:19:21 **Register Number:** RA2031241010094

Q. SORT

Write a bubble-sort program in java to sort the string in array
Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in wrong order.

Source Code

```
import java.util.*;
public class TestClass {
  public static void main(String[] args) {
  Scanner sc=new Scanner(System.in);
      String temp;
      int n,i,j;
      n=sc.nextInt();
      String name[]=new String[n];
      for(i=0;i<n;i++)
      {
        name[i]=sc.next();
      }
      for(i=0;i<n-1;i++)
      {
        for(j=i+1;j<n;j++)
        {
          if(name[i].compareTo(name[j])>0)
          {
            temp=name[i];
            name[i]=name[j];
            name[j]=temp;
          }
        }
      }
      for(i=0;i<n-1;i++)
      {
        System.out.print(name[i]+" ");
      }
      System.out.println(name[n-1]);
  }
}
```

Sample Input

```
5
Ada
Cpp
Lisp
Java
Scala
```

Sample Output

```
Ada Cpp Java Lisp Scala
```

Result

Thus, Program " **SORT** " has been successfully executed

Course: JAVA **Session:** Control and Looping **Timestamp:** 2021-8-31 10:19:44 **Register Number:** RA2031241010094

Q. Sherlock and Squares

Watson gives two integers (A and B) to Sherlock and asks if he can count the number of square integers between A and B (both inclusive).

Note: A square integer is an integer which is the square of any integer. For example, 1, 4, 9, and 16 are some of the square integers as they are squares of 1, 2, 3, and 4, respectively.

Input Format

The first line contains T , the number of test cases. T test cases follow, each in a new line.

Each test case contains two space-separated integers denoting A and B.

Constraints
1<=T<=100
1<=A<=B<=109

Output Format
For each test case, print the required answer in a new line.

Explanation for First Test Case:

In the range 3,9 there are two square numbers which are 4 & 9
In the range 17, 24 there are no square numbers

Source Code

```
import java.io.*;
import java.util.*;
import java.math.*;
public class TestClass {
  public static void main(String[] args) {
  Scanner sc=new Scanner(System.in);
      int t=sc.nextInt();
      for(int i=0;i<t;i++)
      {
        int A=sc.nextInt();
        int B=sc.nextInt();
        int start=(int)Math.sqrt(A);
        int end=(int)Math.sqrt(B);
        int square=end-start;
        square+=(Math.pow(start,2)>=A)?1:0;
        System.out.println(square);
      }
 }
}
```

Sample Input

```
2
3 9
17 24
```

Sample Output

```
2
0
```

Result

Thus, Program " **Sherlock and Squares** " has been successfully executed

Course: JAVA **Session:** Control and Looping **Timestamp:** 2021-8-31 10:20:12 **Register Number:** RA2031241010094

Q. Niven Number

In recreational mathematics, a harshad number (or Niven number) in a given number base, is an integer that is divisible by the sum of its digits when written in that base. Harshad numbers in base n are also known as n-harshad (or n-Niven) numbers. Harshad numbers were defined by D. R. Kaprekar, a mathematician from India. The word "harshad" comes from the Sanskrit hara (joy) + da (give), meaning joy-giver. The term Niven number arose from a paper delivered by Ivan M. Niven at a conference on number theory in 1977. All integers between zero and n are n-harshad numbers.

Write a Program in Java to input a number and check whether it is a Harshad Numbero r Niven Number or not..

An integer (in base 10) that is divisible by the sum of its digits.

Source Code

```
import java.util.*;
public class TestClass {
  public static void main(String[] args) {
  Scanner sc=new Scanner(System.in);
      int rem,sum=0;
      int n=sc.nextInt();
      int m=n;
      while(m>0)
      {
        rem=m%10;
        sum=sum+rem;
        m=m/10;
      }
      int div=n%sum;
      if(div==0)
        System.out.println("Yes");
      else
        System.out.println("No");
 }
}
```

Sample Input

195

Sample Output

Yes

Result

Thus, Program " **Niven Number** " has been successfully executed

Course: JAVA **Session:** Control and Looping **Timestamp:** 2021-8-31 10:20:33 **Register Number:** RA2031241010094

Q. Compare two strings lexicographically

Write a java program to compare two strings lexicographically

If the strings are equal then mentions as follows:

Input 1:

SRM
SRM

Output:

SRM is equal to SRM

Source Code

```
import java.util.*;
public class TestClass {
  public static void main(String[] args) {
  Scanner sc=new Scanner(System.in);
       String s1=sc.nextLine();
       String s2=sc.nextLine();
       int res=s1.compareTo(s2);
       if(res>0)
         System.out.println(s1+" is greater than "+s2);
       else if(res<0)
         System.out.println(s1+" is less than "+s2);
       else
         System.out.println(s1+" is equal to "+s2);
 }
}
```

Sample Input

SRM UNIVERSITY
SRM

Sample Output

SRM UNIVERSITY is greater than SRM

Result

Thus, Program " **Compare two strings lexicographically** " has been successfully executed

Course: JAVA **Session:** Control and Looping **Timestamp:** 2021-8-31 10:20:58 **Register Number:** RA2031241010094

Q. PATTERNS - L6

Write a java program to display the following patterns

Source Code

```
import java.util.*;
public class TestClass {
  public static void main(String[] args) {
  Scanner sc=new Scanner(System.in);
       int n=sc.nextInt();
       for(int i=n;i>=1;i--)
       {
         for(int j=n;j>=i;j--)
         {
           System.out.print(j+" ");
         }
         System.out.println();
       }
 }
}
```

Sample Input

7

Sample Output

```
7
7 6
7 6 5
7 6 5 4
7 6 5 4 3
7 6 5 4 3 2
7 6 5 4 3 2 1
```

Result

Thus, Program " **PATTERNS - L6** " has been successfully executed