Q1. Write a program to swap to two numbers.

ANS.

**import** java.util.\*;

**class** Swap\_With {

**public** **static** **void** main(String[] args) {

**int** x, y, t;// x and y are to swap

       Scanner sc = **new** Scanner(System.in);

       System.out.println("Enter the value of X and Y");

       x = sc.nextInt();

       y = sc.nextInt();

       System.out.println("before swapping numbers: "+x +"  "+ y);

       /\*swapping \*/

       t = x;

       x = y;

       y = t;

       System.out.println("After swapping: "+x +"   " + y);

       System.out.println( );

    }

**Output:**

Enter the value of X and Y

2

3

before swapping numbers: 2 3

After swapping: 3 2

Q2. Write a program to find the largest number among three numbers entered by the user.

ANS. import java.util.Scanner;

public class Exercise3 {

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

System.out.print("Input the 1st number: ");

int num1 = in.nextInt();

System.out.print("Input the 2nd number: ");

int num2 = in.nextInt();

System.out.print("Input the 3rd number: ");

int num3 = in.nextInt();

if (num1 > num2)

if (num1 > num3)

System.out.println("The greatest: " + num1);

if (num2 > num1)

if (num2 > num3)

System.out.println("The greatest: " + num2);

if (num3 > num1)

if (num3 > num2)

System.out.println("The greatest: " + num3);

}

}

Output:

Input the 1st number: 25

Input the 2nd number: 78

Input the 3rd number: 87

The greatest: 87

Q3. Write a program to check whether a year entered by a user is Leap year or not.

ANS.

class Test

{

    static boolean checkYear(int year)

    {

        // If a year is multiple of 400,

        // then it is a leap year

        if (year % 400 == 0)

            return true;

        // Else If a year is multiple of 100,

        // then it is not a leap year

        if (year % 100 == 0)

            return false;

        // Else If a year is multiple of 4,

        // then it is a leap year

        if (year % 4 == 0)

            return true;

        return false;

    }

    // Driver method

    public static void main(String[] args)

    {

        int year = 2000;

        System.out.println( checkYear(2000)? "Leap Year" :

                           "Not a Leap Year" );

    }

}

OUTPUT : LEAP YEAR

.Q4. Write a program to display Fibonacci Series upto nth term. (Using loops)

ANS

. import java.util.Scanner;

public class Main

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

int n, t1 = 0, t2 = 1, nextTerm = 0, i;

System.out.println("Enter the n value: ");

n = sc.nextInt();

if(n == 0 || n == 1)

System.out.println(n);

else

nextTerm = t1 + t2;

for (i = 3; i <= n; ++i)

{

t1 = t2;

t2 = nextTerm;

nextTerm = t1 + t2;

}

System.out.println(t2);

}

}

Output

Enter the n value: 3  
1

Q5. Write a program to check whether a number is Prime or Not.

ANS.

import java.util.Scanner;

class PrimeCheck

{

public static void main(String args[])

{

int temp;

boolean isPrime=true;

Scanner scan= new Scanner(System.in);

System.out.println("Enter any number:");

//capture the input in an integer

int num=scan.nextInt();

scan.close();

for(int i=2;i<=num/2;i++)

{

temp=num%i;

if(temp==0)

{

isPrime=false;

break;

}

}

//If isPrime is true then the number is prime else not

if(isPrime)

System.out.println(num + " is a Prime Number");

else

System.out.println(num + " is not a Prime Number");

}

}

Output 1:

Enter any number:

6

6 is not a Prime Number

Q6. Print this pattern using loops

For n=5

    \*

  \* \*

  \* \* \*

\* \* \* \*

\* \* \* \* \*

ANS.

public class pattern

{

 public static void PyramidPattern(int n)

  {

        for (int i=0; i<n; i++) // for number of rows(n)

{ for (int j=n-i; j>1; j–) // a loop for spaces

            {

                System.out.print(” “); //to print space

            }

            for (int j=0; j<=i; j++ ) //for number of columns

            {

                System.out.print(“\* “); //to print star

            }

            System.out.println(); //end-line after every row

        }

    }

    public static void main(String args[]) //driver function,

    {

        int n = 5;

        PyramidPattern(n);

    }

}

Q7.Write a program that takes n elements from the user and displays the second largest element of an array.

ANS.

import java.util.Arrays;

public class SecondLargestInArrayExample1{

public static int getSecondLargest(int[] a, int total){

Arrays.sort(a);

return a[total-2];

}

public static void main(String args[]){

int a[]={1,2,5,6,3,2};

int b[]={44,66,99,77,33,22,55};

System.out.println("Second Largest: "+getSecondLargest(a,6));

System.out.println("Second Largest: "+getSecondLargest(b,7));

}}

Q8. Given an array and a number, d, perform d left rotations on the array.

ANS.

import java.io.\*;

import java.util.\*;

public class Solution {

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

int N = scan.nextInt();

int n = scan.nextInt();

int[] A = new int[N];

for (int i=0; i<N; ++i) {

A[i] = scan.nextInt();

}

for (int i=0; i<n; ++i) {

rotateArray(A);

}

for (int a : A) {

System.out.print(a+" ");

}

System.out.println("");

}

private static void rotateArray(int[] A) {

int t = A[0];

for (int i=0; i<A.length-1; ++i) {

A[i] = A[i+1];

}

A[A.length-1] = t;

}

}

Q9. Round student grades according to Sam's rules

ANS.

import java.util.\*;

public class Solution {

public static int getRoundedGrade(int grade) {

if (grade >= 38) {

int mod5 = grade % 5;

if (mod5 > 2) {

grade += 5 - mod5;

}

}

return grade;

}

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

int n = in.nextInt();

for(int a0 = 0; a0 < n; a0++){

int grade = in.nextInt();

System.out.println(getRoundedGrade(grade));

}

in.close();

}

}

Q10. CAMELCASE

ANS.

public class Solution {

    public static void main(String[] args) throws IOException {

        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));

        BufferedWriter bufferedWriter = new BufferedWriter(new FileWriter(System.getenv("OUTPUT\_PATH")));

        String s = bufferedReader.readLine();

        int result = Result.camelcase(s);

        bufferedWriter.write(String.valueOf(result));

        bufferedWriter.newLine();

        bufferedReader.close();

        bufferedWriter.close();

    }

}