**LOVELY PROFESSIONAL UNIVERSITY**

**ABHISHEK KUMAR**

**(12012026)**

**Q.1 Print count using Recursion take the user input.**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a number: ");

int n = scanner.nextInt();

printCount(n);

}

public static void printCount(int n) {

if (n <= 0) {

return;

}

printCount(n-1);

System.out.println(n);

}

}

**Q.2 Find the power of 2 using Recursion. Also take the input from the user.**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the power: ");

int power = scanner.nextInt();

int result = powerOfTwo(power);

System.out.println("2^" + power + " = " + result);

}

public static int powerOfTwo(int power) {

if (power == 0) {

return 1;

}

return 2 \* powerOfTwo(power - 1);

}

}

**Q.3 Take the input from the user and find the term of the fibonacci series using recursion and for loop.**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the nth term of Fibonacci series: ");

int n = scanner.nextInt();

System.out.println("The " + n + "th term of Fibonacci series using recursion is: " + fR(n));

System.out.println("The " + n + "th term of Fibonacci series using for loop is: " + fI(n));

}

public static int fR(int n) {

if (n == 0) {

return 0;

} else if (n == 1) {

return 1;

}

return fR(n - 1) + fR(n - 2);

}

public static int fI(int n) {

if (n == 0) {

return 0;

} else if (n == 1) {

return 1;

}

int a = 0;

int b = 1;

int c = 0;

for (int i = 2; i <= n; i++) {

c = a + b;

a = b;

b = c;

}

return c;

}

}

**Q.4 Find the sum of the numbers using recursion and for loop.**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the number of elements: ");

int n = scanner.nextInt();

int[] arr = new int[n];

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

System.out.print("Enter element " + (i+1) + ": ");

arr[i] = scanner.nextInt();

}

int sumRecursive = findSumRecursive(arr, n);

System.out.println("The sum of the numbers using recursion is: " + sumRecursive);

int sumIterative = findSumIterative(arr, n);

System.out.println("The sum of the numbers using for loop is: " + sumIterative);

}

public static int findSumRecursive(int[] arr, int n) {

if (n == 0) {

return 0;

}

return arr[n-1] + findSumRecursive(arr, n-1);

}

public static int findSumIterative(int[] arr, int n) {

int sum = 0;

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

sum += arr[i];

}

return sum;

}

}