**LAB # 03**

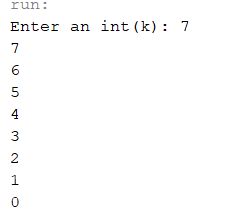
**RECURSION**

**OBJECTIVE:** To understand the complexities of the recursive functions and a way to reduce these complexities.

# LAB TASK

1. Write a program which takes an integer value (k) as input and prints the sequence of numbers from k to 0 in descending order.

**Code: Output:**

import java.util.Scanner;

public class labtsks {

public static void printDescending(int k) {

if (k < 0) return;

System.out.println(k);

printDescending(k - 1);

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter an int(k): ");

int k = sc.nextInt();

printDescending(k);

}

}

1. Write a program to reverse your full name using Recursion.

**Code: Output:**

public class labtsks {

public static String rev(String name) {

if (name.isEmpty()) return name;

return rev(name.substring(1)) + name.charAt(0);

}

public static void main(String[] args) {

String name = "Suhaib";

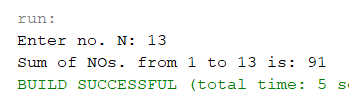
System.out.println("Reversed name: " + rev(name));

}

}

1. Write a program to calculate the sum of numbers from 1 to N using recursion. N should be user input.

**Code: Output:**

 import java.util.Scanner;

public class labtsks {

public static int sum(int n) {

if (n == 0) return 0;

return n + sum(n - 1); }

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter no. N: ");

int N = sc.nextInt();

int result = sum(N);

System.out.println("Sum of NOs. from 1 to " + N + " is: " + result);

}

}

1. Write a recursive program to calculate the sum of elements in an array.

**Code: Output:**

public class labtsks {

public static int sumArray(int[] arr, int index) {

if (index == arr.length) return 0;

return arr[index] + sumArray(arr, index + 1); }

public static void main(String[] args) {

int[] arr = {3, 6, 9, 12, 15};

int result = sumArray(arr, 0);

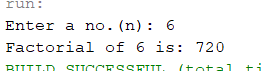
System.out.println("Sum of array elements: " + result);

}

}

1. Write a recursive program to calculate the factorial of a given integer n

**Code: Output:**

import java.util.Scanner;

public class labtsks {

public static int fact(int n) {

if (n == 0) return 1;

return n \* fact(n - 1); }

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter a no.(n): ");

int n = sc.nextInt();

int result = fact(n);

System.out.println("Factorial of " + n + " is: " + result);

}

}

1. Write a program to count the digits of a given number using recursion.

**Code: Output:**

public class labtsks {

public static int countDigits(int num) {

if (num == 0) return 0;

return 1 + countDigits(num / 10); }

public static void main(String[] args) {

int number = 202363;

System.out.println("Number of digits: " + countDigits(number));

}

}

# HOME TASK

1. Write a java program to find the N-th term in the Fibonacci series using Memoization.

**Code: Output:**

import java.util.HashMap;

import java.util.Map;

public class labtsks {

private static Map<Integer, Integer> memoTab = new HashMap<>();

public static int fiboMemoize(int n) {

if (n == 0) return 0;

if (n == 1) return 1;

if (memoTab.containsKey(n)) {

return memoTab.get(n); }

int result = fiboMemoize(n - 1) + fiboMemoize(n - 2);

memoTab.put(n, result);

return result; }

public static void main(String[] args) {

int n = 16;

System.out.println("Fibonacci value for n=" + n + " is: " + fiboMemoize(n));

} }

1. Write a program to count the digits of a given number using recursion.

**Similar to Lab Task # 6**

1. Write a java program to check whether a given string is a palindrome or not.

**Code: Output:**

public class labtsks {

public static boolean isPalindrome(String str, int start, int end) {

if (start >= end) return true;

 if (str.charAt(start) != str.charAt(end)) return false;

return isPalindrome(str, start + 1, end - 1); }

public static void main(String[] args) {

String str = "Suhaib";

if (isPalindrome(str, 0, str.length() - 1)) {

System.out.println("YES");

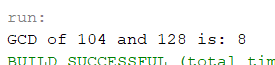
} else {

System.out.println("NO"); }

} }

1. Write a recursive program to find the (GCD) of two numbers using Euclid's algorithm.

**Code: Output:**

public class labtsks {

public static int gcd(int a, int b) {

if (b == 0) return a;

return gcd(b, a % b); }

public static void main(String[] args) {

int a = 104, b = 128;

System.out.println("GCD of " + a + " and " + b + " is: " + gcd(a, b));

} }