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:

```java
package rollno.pkg015;
import java.util.Scanner;
public class Rollno015 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter an integer value: ");
        int k = scanner.nextInt();
        printDescending(k);
    }
    public static void printDescending(int k) {
        if (k < 0) {
            return;
        }
        System.out.println(k + " ");
        printDescending(k - 1);
    }
}
```

OUTPUT:

```
Output - rollno 015 (run)
run:
Enter an integer value: 15
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 BUILD SUCCESSFUL (total time: 3 seconds)
```

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

CODE:

```java
package rollno.pkg015;
import java.util.Scanner;
public class Rollno015 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter your full name: ");
        String name = scanner.nextLine();
        System.out.print("Reversed name: ");
        reverseString(name);
        System.out.println();
    }
    public static void reverseString(String str) {
        if (str.isEmpty()) {
            return;
        }
        System.out.print(str.charAt(str.length() - 1));
        reverseString(str.substring(0, str.length() - 1));
    }
}
```

OUTPUT:

```
Output - rollno 015 (run)
run:
Enter your full name: laiba jaweed
Reversed name: deewaj abial
BUILD SUCCESSFUL (total time: 6 seconds)
```

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

CODE:

```java
package rollno.pkg015;
import java.util.Scanner;
public class Rollno015 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a positive integer N: ");
        int N = scanner.nextInt();
        int sum = calculateSum(N);
        System.out.println("Sum of numbers from 1 to " + N + " is: " + sum);
    }
    public static int calculateSum(int n) {
        if (n == 1) {
            return 1;
        }
        return n + calculateSum(n - 1);
    }
}
```

OUTPUT:

```
Output - rollno 015 (run)
run:
Enter a positive integer N: 15
Sum of numbers from 1 to 15 is: 120
BUILD SUCCESSFUL (total time: 6 seconds)
```

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

CODE:

```java
package rollno.pkg015;
import java.util.Scanner;
public class Rollno015 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the size of the array: ");
        int size = scanner.nextInt();
        int[] arr = new int[size];
        System.out.println("Enter the elements of the array:");
        for (int i = 0; i < size; i++) {
            arr[i] = scanner.nextInt();
        }
        int sum = calculateSum(arr, size - 1);
        System.out.println("Sum of array elements is: " + sum);
    }
    public static int calculateSum(int[] arr, int index) {
        if (index < 0) {
            return 0;
        }
        return arr[index] + calculateSum(arr, index - 1);
    }
}
```

OUTPUT:

Output - rollno 015 (run) #2

```
run:
Enter the size of the array: 3
Enter the elements of the array:
2
4
6
Sum of array elements is: 12
BUILD SUCCESSFUL (total time: 7 seconds)
```

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

CODE:

```java
package rollno.pkg015;
import java.util.Scanner;
public class Rollno015 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
         System.out.print("Enter a positive integer: ");
         int n = scanner.nextInt();
         int result = calculateFactorial(n);
         System.out.println("Factorial of " + n + " is: " + result);
    }
    public static int calculateFactorial(int n) {
        if (n == 0 || n == 1) {
            return 1;
        }
        return n * calculateFactorial(n - 1);
    }
}
```

OUTPUT:

Output - rollno 015 (run) #2

```
run:
Enter a positive integer: 7
Factorial of 7 is: 5040
BUILD SUCCESSFUL (total time: 5 seconds)
```

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

CODE:

```java
package rollno.pkg015;
import java.util.Scanner;
public class Rollno015 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
         System.out.print("Enter a number: ");
         int number = scanner.nextInt();
         int digitCount = countDigits(number);
         System.out.println("Number of digits in " + number + " is: " + digitCount);
    }
    public static int countDigits(int num) {
        if (num == 0) {
            return 0;
        }
        return 1 + countDigits(num / 10);
    }
}
```

OUTPUT:

```
run:
Enter a number: 10526098
Number of digits in 10526098 is: 8
BUILD SUCCESSFUL (total time: 9 seconds)
```

HOME TASK

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

CODE:

```java
package labtask3;
import java.util.Scanner;
import java.util.HashMap;

public class Labtask3 {
    private static HashMap<Integer, Long> memo = new HashMap<>();

    public static long fibonacci(int n) {
        if (n <= 1) {
            return n;
        }
        if (memo.containsKey(n)) {
            return memo.get(n);
        }
        long result = fibonacci(n - 1) + fibonacci(n - 2);
        memo.put(n, result);
        return result;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the position (N) of Fibonacci series: ");
        int n = scanner.nextInt();
        System.out.println("The " + n + "-th term in the Fibonacci series is: " + fibonacci(n));
        scanner.close();
    }
```

OUTPUT:

```
run:
Enter the position (N) of Fibonacci series: 10
The 10-th term in the Fibonacci series is: 55
BUILD SUCCESSFUL (total time: 4 seconds)
```

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

CODE:

```java
package rollno.pkg015;
import java.util.Scanner;
public class Rollno015 {
    public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();
        int digitCount = countDigits(Math.abs(number));
        System.out.println("Number of digits in " + number + " is: " + digitCount);
    }
    public static int countDigits(int num) {
        if (num == 0) {
            return 0;
        }
        return 1 + countDigits(num / 10);
    }
}
```

OUTPUT:

Output - rollno 015 (run) #2

```
run:
Enter a number: 10200543
Number of digits in 10200543 is: 8
BUILD SUCCESSFUL (total time: 8 seconds)
```

3. Write a java program to check whether a given string is a palindrome or not. A palindrome is a string that reads the same forwards and backwards.Print "YES" if the string is a palindrome, otherwise print "NO".

CODE:

```java
package rollno.pkg015;
import java.util.Scanner;
public class Rollno015 {
    public static void main(String[] args) {
     Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String str = scanner.nextLine();
        if (isPalindrome(str)) {
            System.out.println("YES");
        } else {
            System.out.println("NO");
        }
    }
    public static boolean isPalindrome(String str) {
        if (str.length() <= 1) {
            return true;
        }
        if (str.charAt(0) == str.charAt(str.length() - 1)) {
            return isPalindrome(str.substring(1, str.length() - 1));
        }
        return false;
    }
}
```

OUTPUT:

```
run:
Enter a string: madam
YES
BUILD SUCCESSFUL (total time: 3 seconds)
```

4. Write a recursive program to find the greatest common divisor (GCD) of two numbers using Euclid's algorithm.

CODE:

```java
package rollno.pkg015;
import java.util.Scanner;
public class Rollno015 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the first number: ");
        int a = scanner.nextInt();
        System.out.print("Enter the second number: ");
        int b = scanner.nextInt();
        int gcd = findGCD(a, b);
        System.out.println("The GCD of " + a + " and " + b + " is: " + gcd);
    }
    public static int findGCD(int a, int b) {
        if (b == 0) {
            return a;
        }
        return findGCD(b, a % b);
    }
}
```

OUTPUT:

```
run:
Enter the first number: 4
Enter the second number: 12
The GCD of 4 and 12 is: 4
BUILD SUCCESSFUL (total time: 8 seconds)
```