

LAB # 02

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.

```
import java.util.Scanner;

no usages
public class Main {
    no usages
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a value for k: ");
        int k = scanner.nextInt();
        for (int i = k; i >= 0; i--) {
            System.out.print(i + " ");
        }
    }
}
```

Output:

```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition
Enter a value for k: 10
10 9 8 7 6 5 4 3 2 1 0
Process finished with exit code 0
```

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

```
public class Main {
```

no usages

```
| public static void main(String[] args) {  
|     String fullName = "Nabeera Siddiqui";  
|     String reversedName = reverseName(fullName);  
|     System.out.println("Reversed Name: " + reversedName);  
| }  
  
2 usages
```

```
| public static String reverseName(String name) {  
|     if (name.isEmpty()) {  
|         return name;  
|     } else {  
|         return reverseName(name.substring(beginIndex: 1)) + name.charAt(0);  
|     }  
| }  
| }
```

Output:

```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition  
Reversed Name: iuqiddiS areebaN
```

```
Process finished with exit code 0
```

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

```
import java.util.Scanner;
```

no usages

```
public class Main {
```

2 usages

```
    public static int calculateSum(int n) {  
        if (n == 1) {  
            return 1;  
        } else {  
            return n + calculateSum(n - 1);  
        }  
    }  
}
```

no usages

```
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
  
        System.out.print("Enter a positive integer (N): ");  
        int n = scanner.nextInt();  
  
        if (n < 1) {  
            System.out.println("Please enter a positive integer.");  
        } else {  
            int sum = calculateSum(n);  
            System.out.println("The sum of numbers from 1 to " + n + " is: " + sum);  
        }  
    }  
}
```

Output:

```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition
```

```
Enter a positive integer (N): 5
```

```
The sum of numbers from 1 to 5 is: 15
```

```
Process finished with exit code 0
```

HOME TASK

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

```

import java.util.HashMap;
no usages
public class Main {
    3 usages
    private static HashMap<Integer, Integer> memo = new HashMap<>();

    3 usages
    public static int fibonacci(int n) {
        if (n <= 1) {
            return n;
        }

        if (memo.containsKey(n)) {
            return memo.get(n);
        }
        int result = fibonacci(n - 1) + fibonacci(n - 2);
        memo.put(n, result);
        return result;
    }
    no usages
    public static void main(String[] args) {
        int n = 10;

        int result = fibonacci(n);

        System.out.println("The " + n + "-th Fibonacci term is: " + result);
    }
}

```

Output:

```

"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition
The 10-th Fibonacci term is: 55

```

```

Process finished with exit code 0

```

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

```

public class Main {
    2 usages
    public static int countDigits(int number) {
        if (number < 10) {
            return 1;
        } else {
            return 1 + countDigits( number: number / 10);
        }
    }

    no usages
    public static void main(String[] args) {
        int number = 2003;
        int result = countDigits(number);
        System.out.println("Number of digits in " + number + " is: " + result);
    }
}

```

Output:

```

"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition
Number of digits in 2003 is: 4

```

```

Process finished with exit code 0

```