LAB # 03

RECURSION

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

LAB TASKS

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
import java.util.Scanner;

public class BSE038 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter an integer value (k): ");
        int k = scanner.nextInt();
        printDescending(k);
        scanner.close();
    }
    public static void printDescending(int k) {
        if (k < 0) return;
        System.out.println(k);
        printDescending(k - 1);
    }
}
```

OUTPUT :

```
java -cp /tmp/CufUlTpWUB/BSE038
Enter an integer value (k): 1
1
0
```

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

CODE :

```java
import java.util.Scanner;

public class BSE038 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter your full name: ");
        String name = scanner.nextLine();
        String reversedName = reverseName(name);
        System.out.println("Reversed Name: " + reversedName);
        scanner.close();
    }
    public static String reverseName(String name) {
        if (name.isEmpty()) return name;
        return reverseName(name.substring(1)) + name.charAt(0);
    }
}
```

Output :

```
java -cp /tmp/RmZR9JphtV/BSE038
Enter your full name: abdullah
Reversed Name: halludba
```

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

CODE:

```java
import java.util.Scanner;

public class BSE038 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter an integer N: ");
        int N = scanner.nextInt();
        int sum = calculateSum(N);
        System.out.println("Sum from 1 to " + N + " is: " + sum);
        scanner.close();
    }
    public static int calculateSum(int n) {
        if (n == 0) return 0;
        return n + calculateSum(n - 1);
    }
}
```

Output :

```
java -cp /tmp/S1eRHbI6Ys/BSE038
Enter an integer N: 5
Sum from 1 to 5 is: 15
```

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

Code:

```java
import java.util.Scanner;

public class BSE038 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of elements in the array: ");
        int size = scanner.nextInt();
        int[] array = new int[size];
        System.out.println("Enter the elements of the array:");
        for (int i = 0; i < size; i++) {
            array[i] = scanner.nextInt();
        }
        int sum = sumArray(array, size);
        System.out.println("Sum of array elements: " + sum);
        scanner.close();
    }
    public static int sumArray(int[] array, int size) {
        if (size == 0) return 0;
        return array[size - 1] + sumArray(array, size - 1);
    }
}
```

Output:

```
java -cp /tmp/g7DLo9n46w/BSE038
Enter the number of elements in the array: 1
Enter the elements of the array:
2
Sum of array elements: 2
```

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

Code:

```
import java.util.Scanner;

public class BSE038 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter an integer n to calculate its factorial: ");
        int n = scanner.nextInt();
        int factorial = calculateFactorial(n);
        System.out.println("Factorial of " + n + " is: " + factorial);
        scanner.close();
    }
    public static int calculateFactorial(int n) {
        if (n == 0) return 1;
        return n * calculateFactorial(n - 1);
    }
}
```

Output:

```
java -cp /tmp/FxVw4u6XXt/BSE038
Enter an integer n to calculate its factorial: 4
Factorial of 4 is: 24
```

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

Code:

```
import java.util.Scanner;

public class BSE038 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter an integer number: ");
        int number = scanner.nextInt();
        int digitCount = countDigits(number);
        System.out.println("Number of digits: " + digitCount);
        scanner.close();
    }
    public static int countDigits(int n) {
        if (n == 0) return 0;
        return 1 + countDigits(n / 10);
    }
}
```

HOME TASKS

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

Code:

```java
import java.util.Scanner;

public class BSE038 {
    private static int[] memo;
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the term N for Fibonacci series: ");
        int N = scanner.nextInt();
        memo = new int[N + 1];
        int fibonacci = fibonacci(N);
        System.out.println("Fibonacci term at position " + N + " is: " + fibonacci);
        scanner.close();
    }
    public static int fibonacci(int n) {
        if (n <= 1) return n;
        if (memo[n] != 0) return memo[n];
        memo[n] = fibonacci(n - 1) + fibonacci(n - 2);
        return memo[n];
    }
}
```

Output:

```
java -cp /tmp/LkDfngbgpW/BSE038
Enter the term N for Fibonacci series: 4
Fibonacci term at position 4 is: 3
```

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

Code:

```java
import java.util.Scanner;

public class BSE038 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter an integer number: ");
        int number = scanner.nextInt();
        int digitCount = countDigits(number);
        System.out.println("Number of digits: " + digitCount);
        scanner.close();
    }
    public static int countDigits(int n) {
        if (n == 0) return 0;
        return 1 + countDigits(n / 10);
    }
}
```

Output:

```
Enter an integer number: 4
Number of digits: 1
```

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
import java.util.Scanner;

public class BSE038 {
    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");
        }
        scanner.close();
    }
    public static boolean isPalindrome(String str) {
        return checkPalindrome(str, 0, str.length() - 1);
    }
    public static boolean checkPalindrome(String str, int left, int right) {
        if (left >= right) return true;
        if (str.charAt(left) != str.charAt(right)) return false;
        return checkPalindrome(str, left + 1, right - 1);
    }
}
```

Output:

```
java -cp /tmp/fIvQuuCvSp/BSE038
Enter a string: racecar
YES
```

4. Write a recursive program to find the greatest common divisor (GCD) of two numbers

Code:

```java
import java.util.Scanner;

public class BSE038 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter first number: ");
        int a = scanner.nextInt();
        System.out.print("Enter second number: ");
        int b = scanner.nextInt();
        int gcd = findGCD(a, b);
        System.out.println("GCD of " + a + " and " + b + " is: " + gcd);
        scanner.close();
    }
    public static int findGCD(int a, int b) {
        if (b == 0) return a;
        return findGCD(b, a % b);
    }
}
```

```
java -cp /tmp/ovH22fx6zk/BSE03
Enter first number: 1
Enter second number: 4
GCD of 1 and 4 is: 1
```