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.

```
static void printDescending(int k) {
    if (k < 0) {
        return;
    }
    System.out.println(k);
    printDescending(k - 1);
}
public static void main(String[] args) {

    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter a number: ");
    int k = scanner.nextInt();
    printDescending(k);
```

OUTPUT:

```
Enter a number: 4
4
3
2
1
0
```

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

```
static void reverseName(String name){
    if (name.length() > 0) {
        reverseName(name.substring(1));
        System.out.print(name.charAt(0));
}

public static void main(String[] args) {
    String fullName = "A.Ahad";
    System.out.println("Original Name: " + fullName);
    System.out.print("Reversed Name: ");
    reverseName(fullName);
```

OUTPUT:

```
run:
Original Name: A.Ahad
Reversed Name: dahA.AB
```

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

```
static int sum(int n) {
    if (n == 0) {
        return 0;
    } else {
        return n + sum(n - 1);
    }
}
 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 = sum(N);
    System.out.println("The sum of numbers from 1 to " + N + " is: " + result);
```

OUTPUT:

```
Enter a positive integer: 12
The sum of numbers from 1 to 12 is: 78
```

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

```
public static int calculateSum(int[] arr, int index) {
    if (index < 0) {
        return 0;
    }
    return arr[index] + calculateSum(arr, index - 1);
}
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
    int[] arr = {12, 22, 13, 14, 15};

    int sum = calculateSum(arr, arr.length - 1);
    System.out.println("Sum of elements in the array: " + sum);
```

OUTPUT:

```
run:
Sum of elements in the array: 76
```

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

```
    public static int factorial(int n) {
        if (n <= 1) {
            return 1;
        }
        return n * factorial(n - 1);
    }
    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 = factorial(n);
        System.out.println("Factorial of " + n + " is: " + result);
```

OUTPUT:

```
Enter a positive integer: 11
Factorial of 11 is: 39916800
```

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

```
    public static int countDigits(int number) {
        if (number == 0) {
        return 0;
    }
    return 1 + countDigits(number / 10);
}
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: " + digitCount);
```

OUTPUT:

```
Enter a number: 12002
Number of digits: 5
```

HOME TASK

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

```
    private static HashMap<Integer, Long> memo = new HashMap<>();
       public static long fibonacci(int n) {
    if (n == 0) return 0;
    if (n == 1) return 1;
   if (memo.containsKey(n)) {
        return memo.get(n);
    }
    long fibValue = fibonacci(n - 1) + fibonacci(n - 2);
    memo.put(n, fibValue);
    return fibValue;
}

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

OUTPUT:

```
run:
Enter the n-th term to find in the Fibonacci series: 6
The 6-th term in the Fibonacci series is: 8
```

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

```
      public static int countDigits(int number) {
        if (number == 0) {
        return 0;
      }
      return 1 + countDigits(number / 10);
  }
   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: " + digitCount);
```

OUTPUT:

```
Enter a number: 12002
Number of digits: 5
```

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 thestring is a palindrome, otherwise print "NO".

```
private static boolean isPalindrome(String str) {
  int left = 0;
  int right = str.length() - 1;

  while (left < right) {
     if (str.charAt(left) != str.charAt(right)) {
        return false;
     }
     left++;
     right--;
  }
  return true;
}
public static void main(String[] args) {
 Scanner scanner = new Scanner(System.in);
  System.out.print("Enter a string: ");
  String input = scanner.nextLine();
  String cleanedInput = input.replaceAll("\\s+", "").toLowerCase();
  if (isPalindrome(cleanedInput)) {
     System.out.println("YES");
  } else {
     System.out.println("NO");
  }
```

OUTPUT:

```
run:
Enter a string: A.Ahad
NO
```

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

```
    private static HashMap<Integer, Long> memo = new HashMap<>();
   private static int gcd(int a, int b) {
    if (b == 0) {
        return a;
    }
    return gcd(b, a % b);
}
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 = gcd(a, b);

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

OUTPUT:

```
Enter the first number: 18
Enter the second number: 4
The GCD of 18 and 4 is: 2
```