

LAB NO 3

RECURSION

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

Question 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.

INPUT:

```
import java.util.Scanner;
public class Anas {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter an integer: ");
        if (scanner.hasNextInt()) {
            int k = scanner.nextInt();

            for (int i = k; i >= 0; i--) {
                System.out.println(i);
            }
        } else {
            System.out.println("wrong value");
        }
        scanner.close();
    }
}
```

OUTPUT:

```
Enter an integer: 8
8
7
6
5
4
3
2
1
0
```

Question 2:

Write a program to reverse your full name using Recursion.

INPUT:

```
package com.mycompany.anas;

import java.util.Scanner;

public class Anas {
    public static String reverse(String name) {
        if (name.length() <= 1) {
            return name;
        }
        return reverse(name.substring(1)) + name.charAt(0);
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter your full name: ");
        String fullName = scanner.nextLine();
        String reversedName = reverse(fullName);
        System.out.println("Reversed name: " + reversedName);
        scanner.close();
    }
}
```

OUTPUT:

```
Enter your full name: Anas Ahmed
Reversed name: demhA sanA
-----
```

Question 3:

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

INPUT:

```
package com.mycompany.anas;

import java.util.Scanner;

public class Anas {

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

    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 <= 0) {
            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);
        }

        scanner.close();
    }
}
```

OUTPUT:

```
Enter a positive integer N: 5
The sum of numbers from 1 to 5 is: 15
```

Question 4:

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

INPUT:

```
package com.mycompany.anas;

import java.util.Scanner;

public class Anas {
    public static int calculateSum(int[] arr, int n) {
        if (n == 0) {
            return 0;
        }
        return arr[n - 1] + calculateSum(arr, n - 1);
    }

    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 " + size + " elements:");
        for (int i = 0; i < size; i++) {
            arr[i] = scanner.nextInt();
        }

        int sum = calculateSum(arr, size);
        System.out.println("The sum of the array elements is: " + sum);

        scanner.close();
    }
}
```

OUTPUT:

```
Enter the size of the array: 4
Enter 4 elements:
2
3
4
5
The sum of the array elements is: 14
```

Question 5:

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

INPUT:

```
package com.mycompany.anas;

import java.util.Scanner;

public class Anas {
    public static long factorial(int n) {
        if (n == 0 || 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 non-negative integer: ");
        int n = scanner.nextInt();
        if (n < 0) {
            System.out.println("Factorial is not defined .");
        } else {
            long result = factorial(n);
            System.out.println("The factorial of " + n + " is: " + result);
        }

        scanner.close();
    }
}
```

OUTPUT:

```
Enter a non-negative integer: 10
The factorial of 10 is: 3628800
```

Question 6:

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

INPUT:

```
package com.mycompany.anas;

import java.util.Scanner;

public class Anas {
    public static int countDigits(int num) {
        if (num == 0) {
            return 0;
        }

        return 1 + countDigits(num / 10);
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = scanner.nextInt();
        num = Math.abs(num);
        int digitCount = countDigits(num);
        System.out.println("The number of digits in the given number is: " + digitCount);
        scanner.close();
    }
}
```

OUTPUT:

```
Enter a number: 5
The number of digits in the given number is: 1
```

HOME TASKS

Question 1:

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

INPUT:

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

public class Anas {

    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 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 term (N) to find in the Fibonacci series: ");
        int n = scanner.nextInt();
        if (n < 0) {
            System.out.println("Please enter a non-negative integer.");
        } else {
            long nthFibonacci = fibonacci(n);
            System.out.println("The " + n + "-th term in the Fibonacci series is: " + nthFibonacci);
        }
        scanner.close();
    }
}
```

OUTPUT:

```
Enter the term (N) to find in the Fibonacci series: 7
The 7-th term in the Fibonacci series is: 13
```

Question 2:

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

INPUT:

```
package com.mycompany.anas;

import java.util.Scanner;

public class Anas {
    public static int countDigits(int num) {
        if (num == 0) {
            return 0;
        }
        return 1 + countDigits(num / 10);
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = scanner.nextInt();
        if (num == 0) {
            System.out.println("The number of digits in the given number is: 1");
        } else {
            num = Math.abs(num);
            int digitCount = countDigits(num);
            System.out.println("The number of digits in the given number is: " + digitCount);
        }
        scanner.close();
    }
}
```

OUTPUT:

```
Enter a number: 15
The number of digits in the given number is: 2
-----
```


Question 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".

INPUT:

```
import java.util.Scanner;

public class Anas {
    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) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();
        String sanitizedInput = input.replaceAll("\\s+", "").toLowerCase();
        if (isPalindrome(sanitizedInput, 0, sanitizedInput.length() - 1)) {
            System.out.println("YES");
        } else {
            System.out.println("NO");
        }

        scanner.close();
    }
}
```

OUTPUT:

```
Enter a string: ANas
NO
```

Question 4:

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

INPUT:

```
public class Anas {  
  
    public static int findGCD(int a, int b) {  
        if (b == 0) {  
            return a;  
        }  
        return findGCD(b, a % b);  
    }  
  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.print("Enter the first number: ");  
        int num1 = scanner.nextInt();  
        System.out.print("Enter the second number: ");  
        int num2 = scanner.nextInt();  
        num1 = Math.abs(num1);  
        num2 = Math.abs(num2);  
        int gcd = findGCD(num1, num2);  
        System.out.println("The GCD of " + num1 + " and " + num2 + " is: " + gcd);  
  
        scanner.close();  
    }  
}
```

OUTPUT:

```
Enter the first number: 5  
Enter the second number: 7  
The GCD of 5 and 7 is: 1
```