LAB # 03

RECURSION

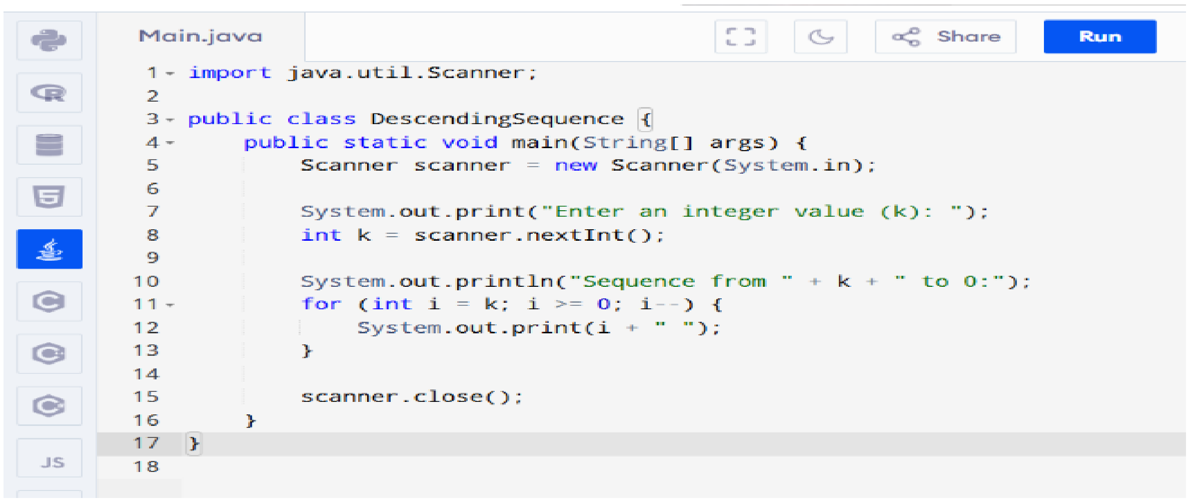
OBJECTIVE:

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

LAB TASK

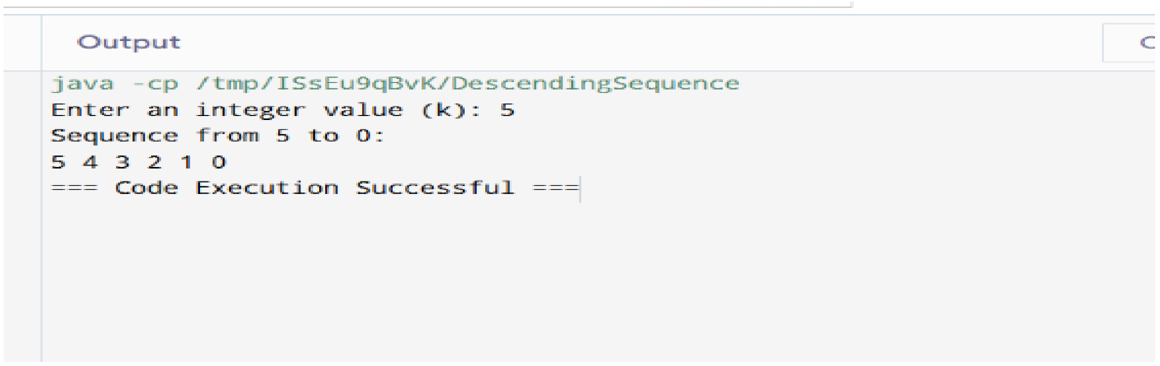
1. Write a program which takes an integer value (k) as input and prints the sequence of numbers from k to 0 in descending order.

CODE:



```java
import java.util.Scanner;

public class DescendingSequence {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter an integer value (k): ");
        int k = scanner.nextInt();

        System.out.println("Sequence from " + k + " to 0:");
        for (int i = k; i >= 0; i--) {
            System.out.print(i + " ");
        }

        scanner.close();
    }
}
```

OUTPUT:

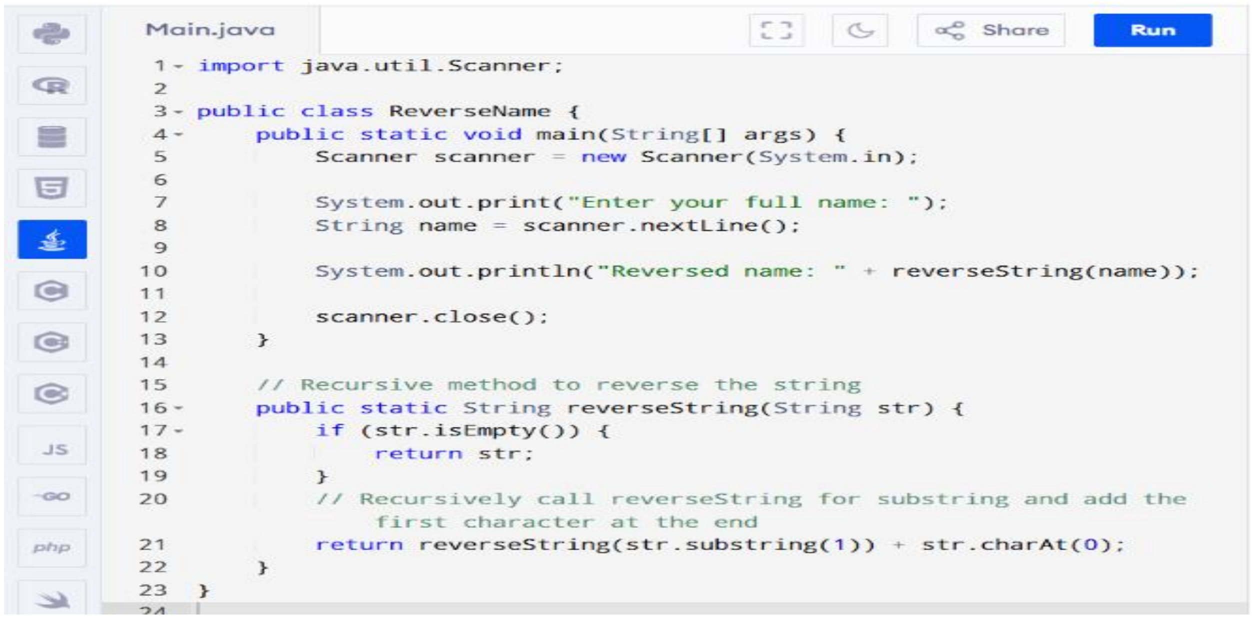


```
java -cp /tmp/ISsEu9qBvK/DescendingSequence
Enter an integer value (k): 5
Sequence from 5 to 0:
5 4 3 2 1 0
=== Code Execution Successful ===
```

2023F-BSE-044
ZAINAB AHMED

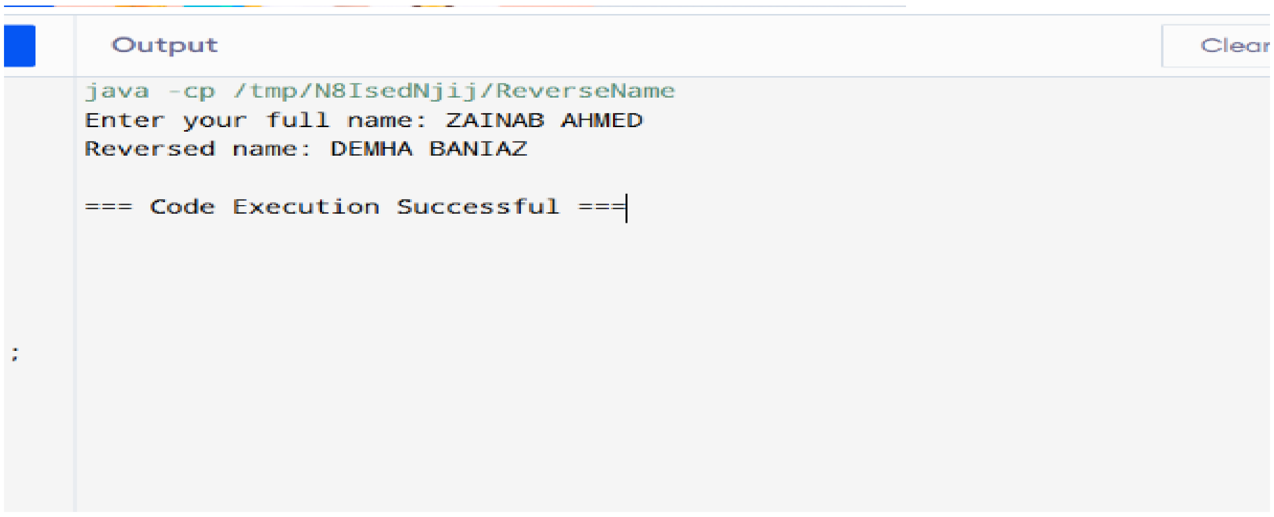
2. Write a program to reverse your full name using Recursion.

CODE:



```java
import java.util.Scanner;

public class ReverseName {
    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.println("Reversed name: " + reverseString(name));

        scanner.close();
    }

    // Recursive method to reverse the string
    public static String reverseString(String str) {
        if (str.isEmpty()) {
            return str;
        }
        // Recursively call reverseString for substring and add the
            first character at the end
        return reverseString(str.substring(1)) + str.charAt(0);
    }
}
```

OUTPUT:

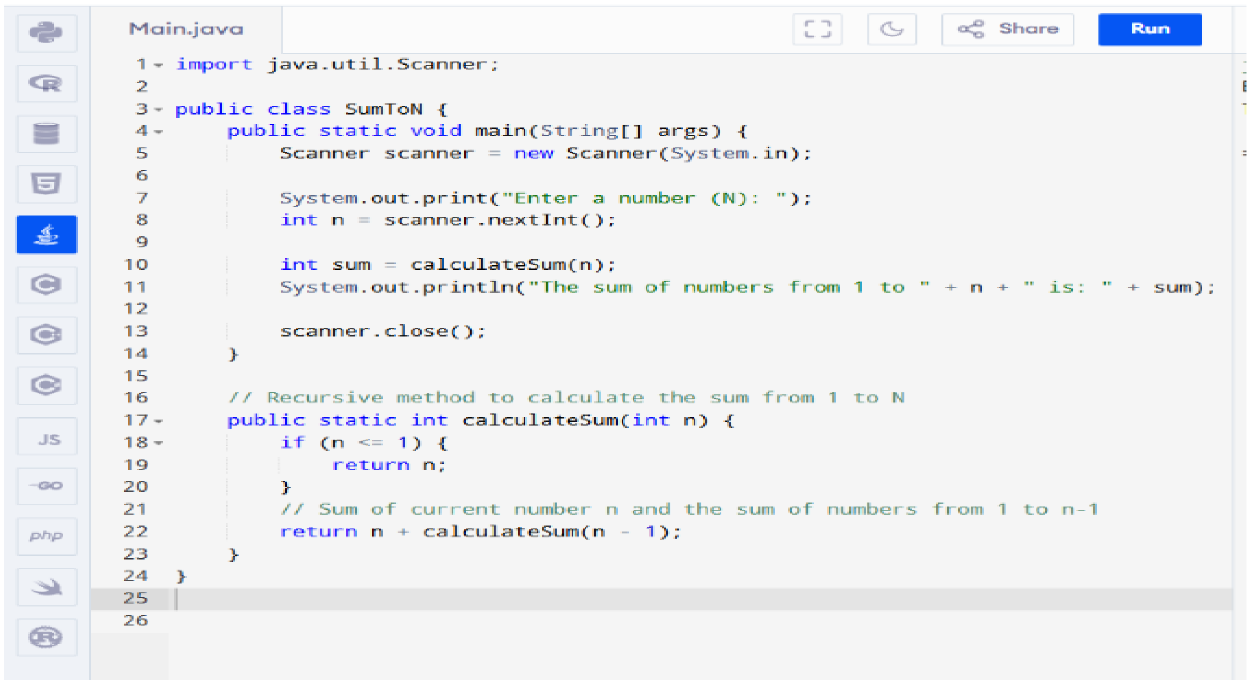


```
java -cp /tmp/N8IsedNjij/ReverseName
Enter your full name: ZAINAB AHMED
Reversed name: DEMHA BANIAZ

=== Code Execution Successful ===
```

2023F-BSE-044
ZAINAB AHMED

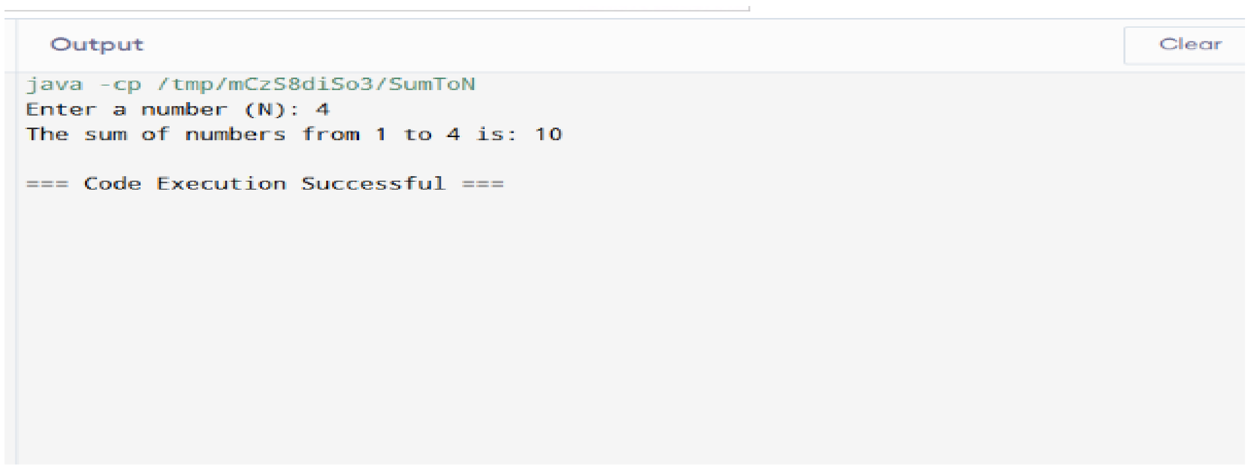
3. Write a program to calculate the sum of numbers from 1 to N using recursion. N should be user input.

CODE:



```java
import java.util.Scanner;

public class SumToN {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a number (N): ");
        int n = scanner.nextInt();

        int sum = calculateSum(n);
        System.out.println("The sum of numbers from 1 to " + n + " is: " + sum);

        scanner.close();
    }

    // Recursive method to calculate the sum from 1 to N
    public static int calculateSum(int n) {
        if (n <= 1) {
            return n;
        }
        // Sum of current number n and the sum of numbers from 1 to n-1
        return n + calculateSum(n - 1);
    }
}
```

OUTPUT:

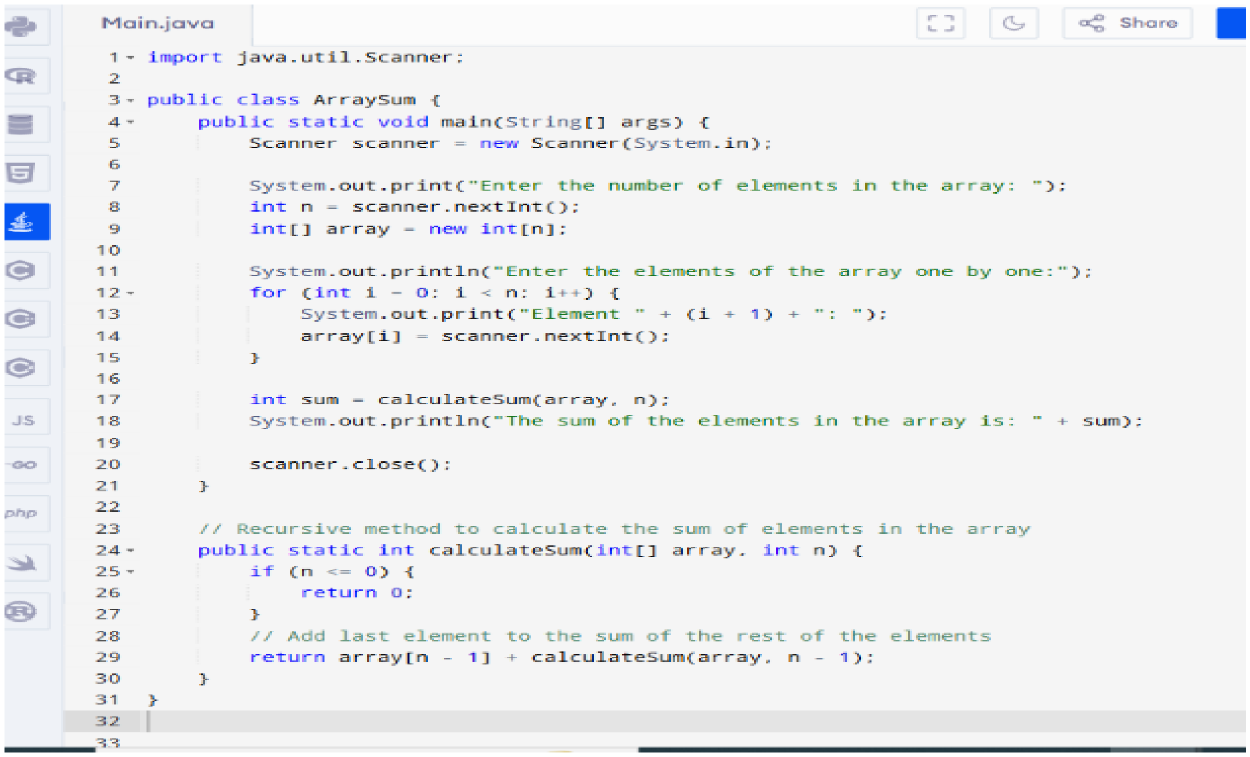


```
java -cp /tmp/mCzS8diSo3/SumToN
Enter a number (N): 4
The sum of numbers from 1 to 4 is: 10

=== Code Execution Successful ===
```

2023F-BSE-044
ZAINAB AHMED

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

CODE:

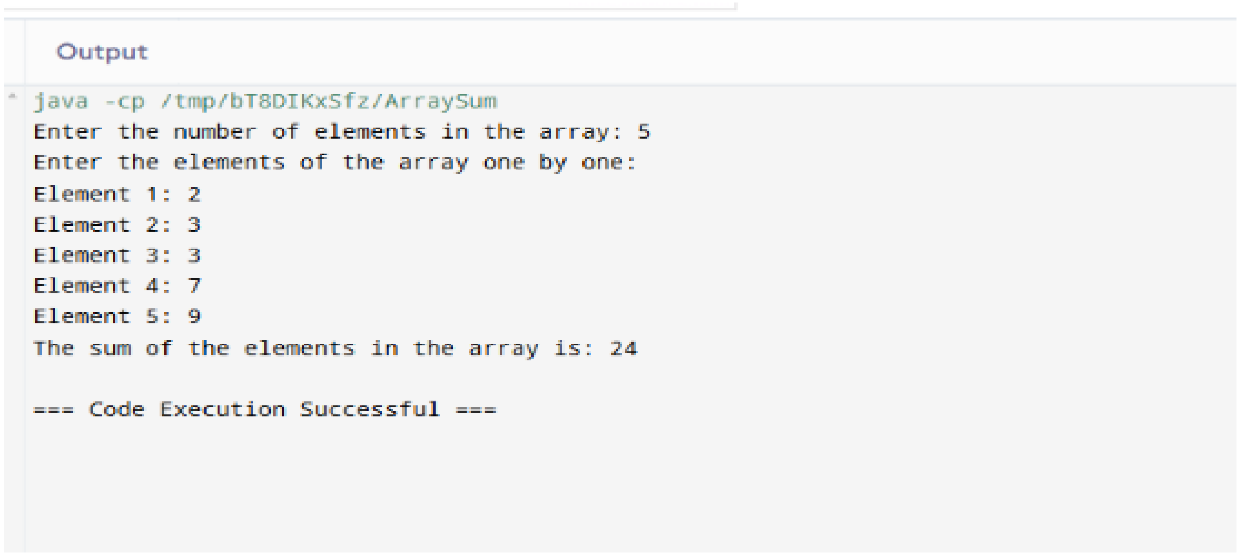


Main.java

Share

```java
import java.util.Scanner;

public class ArraySum {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of elements in the array: ");
        int n = scanner.nextInt();
        int[] array = new int[n];

        System.out.println("Enter the elements of the array one by one:");
        for (int i = 0; i < n; i++) {
            System.out.print("Element " + (i + 1) + ": ");
            array[i] = scanner.nextInt();
        }

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

        scanner.close();
    }

    // Recursive method to calculate the sum of elements in the array
    public static int calculateSum(int[] array, int n) {
        if (n <= 0) {
            return 0;
        }
        // Add last element to the sum of the rest of the elements
        return array[n - 1] + calculateSum(array, n - 1);
    }
}
```

OUTPUT:



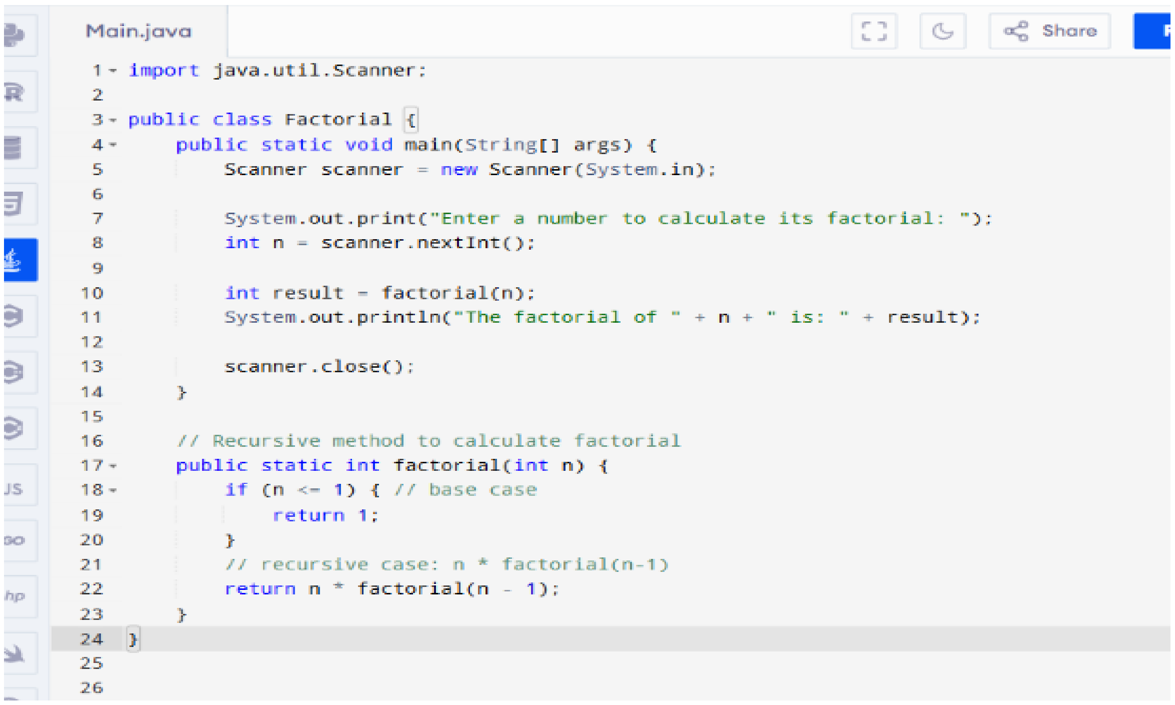
Output

```
java -cp /tmp/bT8DIKxSfz/ArraySum
Enter the number of elements in the array: 5
Enter the elements of the array one by one:
Element 1: 2
Element 2: 3
Element 3: 3
Element 4: 7
Element 5: 9
The sum of the elements in the array is: 24

=== Code Execution Successful ===
```

2023F-BSE-044
ZAINAB AHMED

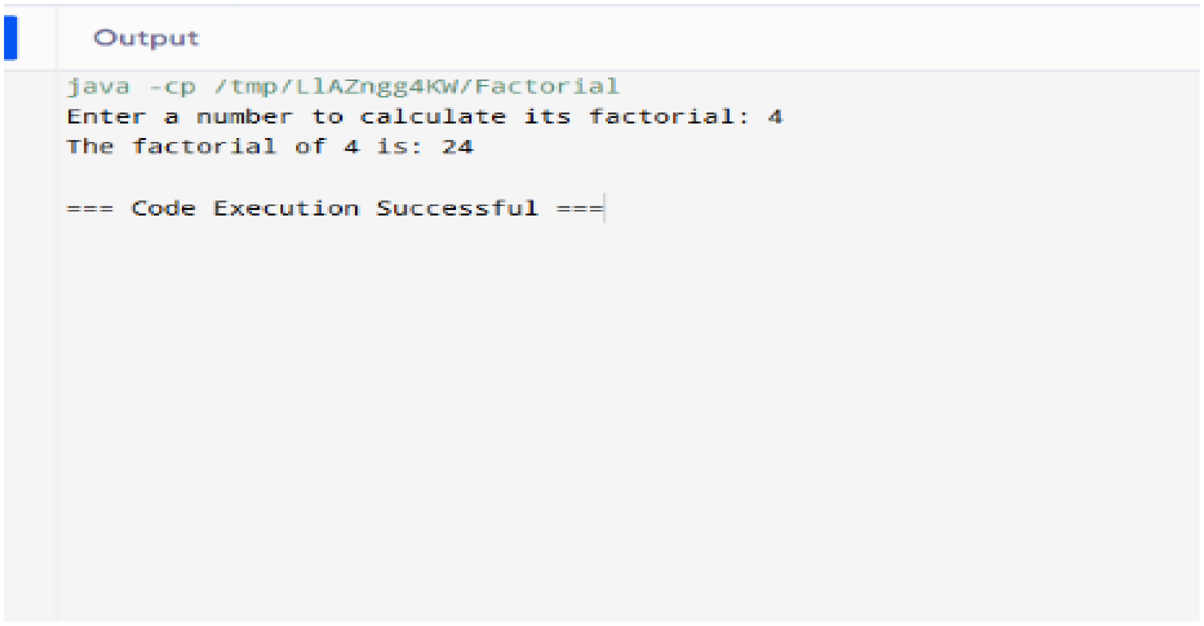
5. Write a recursive program to calculate the factorial of a given integer n

CODE:



```java
import java.util.Scanner;

public class Factorial {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a number to calculate its factorial: ");
        int n = scanner.nextInt();

        int result = factorial(n);
        System.out.println("The factorial of " + n + " is: " + result);

        scanner.close();
    }

    // Recursive method to calculate factorial
    public static int factorial(int n) {
        if (n <= 1) { // base case
            return 1;
        }
        // recursive case: n * factorial(n-1)
        return n * factorial(n - 1);
    }
}


```

OUTPUT:

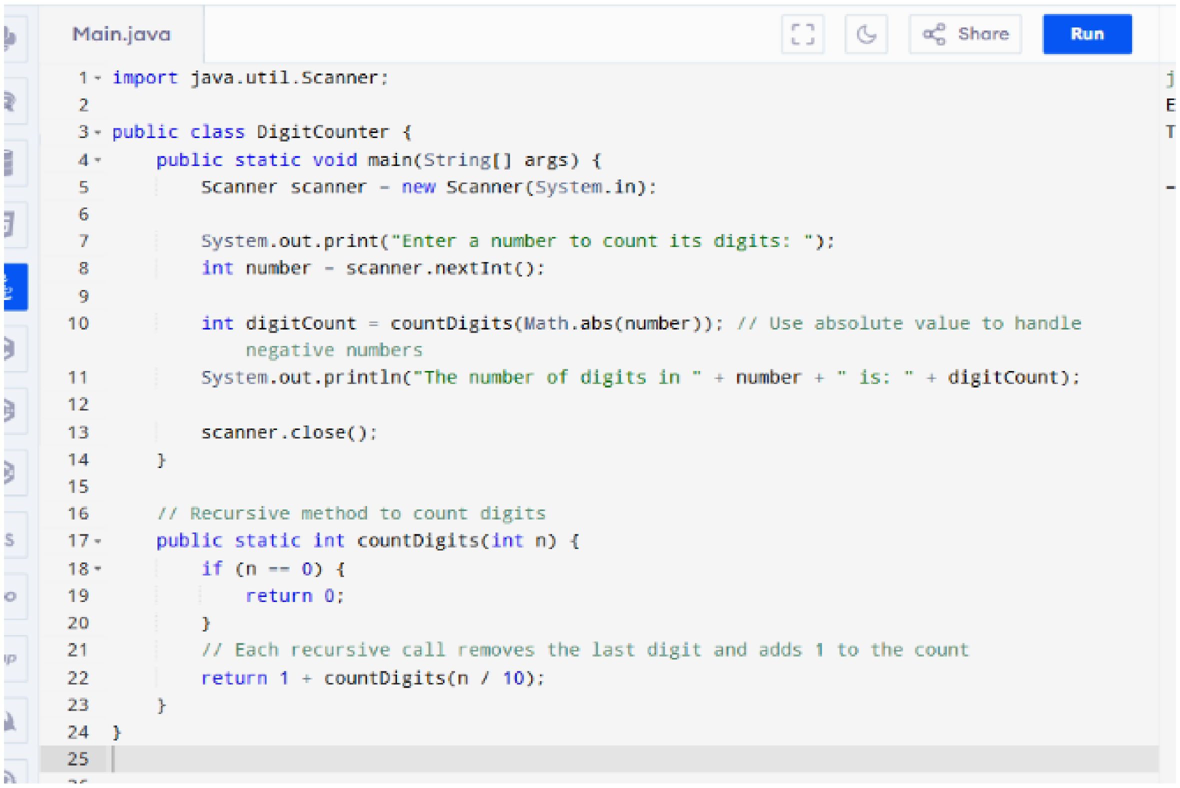


```
java -cp /tmp/L1AZngg4KW/Factorial
Enter a number to calculate its factorial: 4
The factorial of 4 is: 24

=== Code Execution Successful ===
```

2023F-BSE-044
ZAINAB AHMED

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

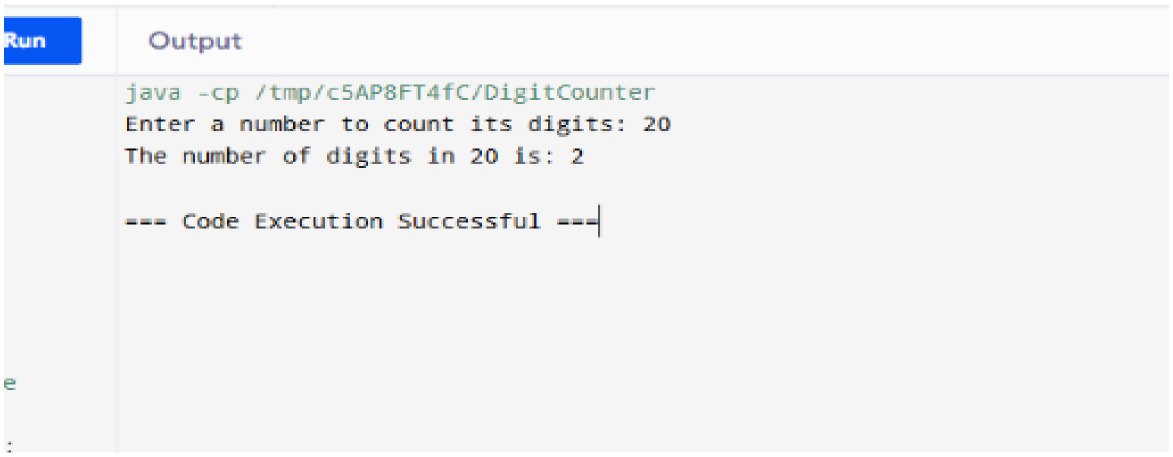
CODE:



Main.java

```java
import java.util.Scanner;

public class DigitCounter {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a number to count its digits: ");
        int number = scanner.nextInt();

        int digitCount = countDigits(Math.abs(number)); // Use absolute value to handle
            negative numbers
        System.out.println("The number of digits in " + number + " is: " + digitCount);

        scanner.close();
    }

    // Recursive method to count digits
    public static int countDigits(int n) {
        if (n == 0) {
            return 0;
        }
        // Each recursive call removes the last digit and adds 1 to the count
        return 1 + countDigits(n / 10);
    }
}
```

OUTPUT:



Output

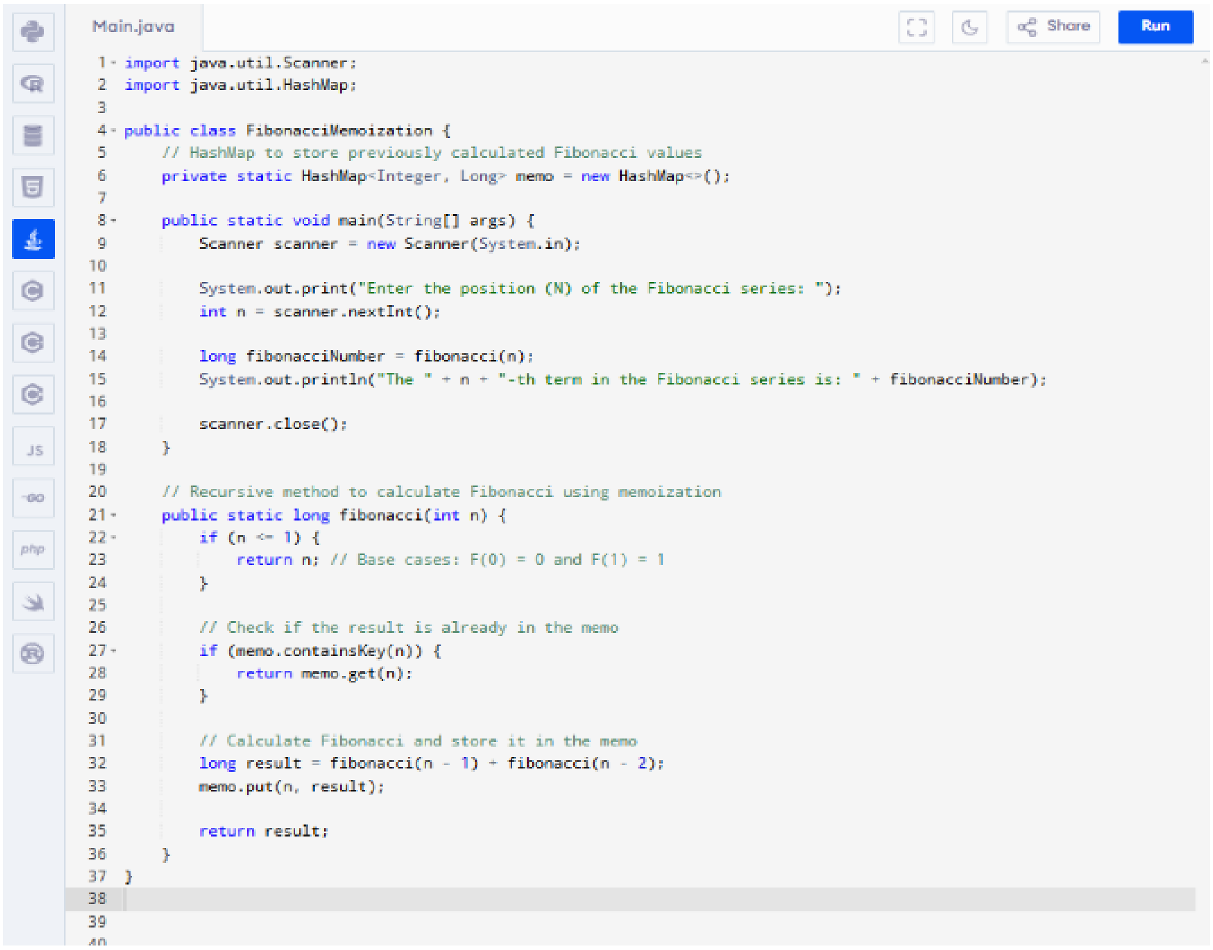
```
java -cp /tmp/c5AP8FT4fC/DigitCounter
Enter a number to count its digits: 20
The number of digits in 20 is: 2

=== Code Execution Successful ===
```

2023F-BSE-044
ZAINAB AHMED

HOME TASK

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

CODE:

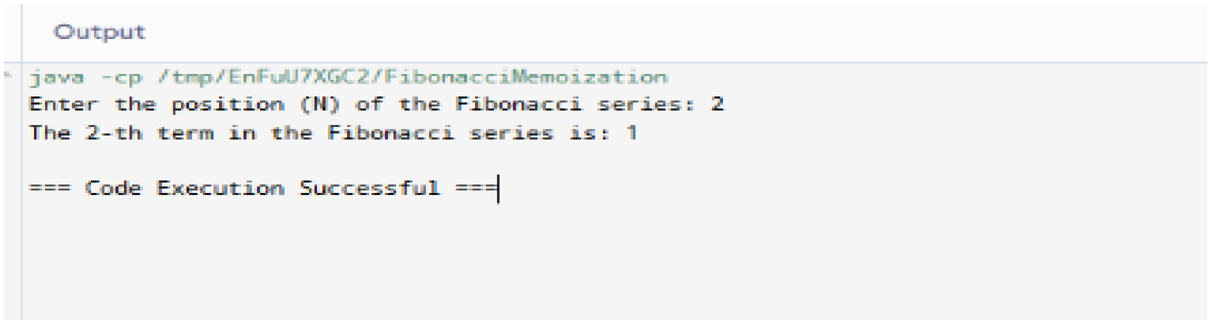


Main.java

Share Run

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

public class FibonacciMemoization {
    // HashMap to store previously calculated Fibonacci values
    private static HashMap<Integer, Long> memo = new HashMap<>();

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the position (N) of the Fibonacci series: ");
        int n = scanner.nextInt();

        long fibonacciNumber = fibonacci(n);
        System.out.println("The " + n + "-th term in the Fibonacci series is: " + fibonacciNumber);

        scanner.close();
    }

    // Recursive method to calculate Fibonacci using memoization
    public static long fibonacci(int n) {
        if (n <= 1) {
            return n; // Base cases: F(0) = 0 and F(1) = 1
        }

        // Check if the result is already in the memo
        if (memo.containsKey(n)) {
            return memo.get(n);
        }

        // Calculate Fibonacci and store it in the memo
        long result = fibonacci(n - 1) + fibonacci(n - 2);
        memo.put(n, result);

        return result;
    }
}
```

OUTPUT:



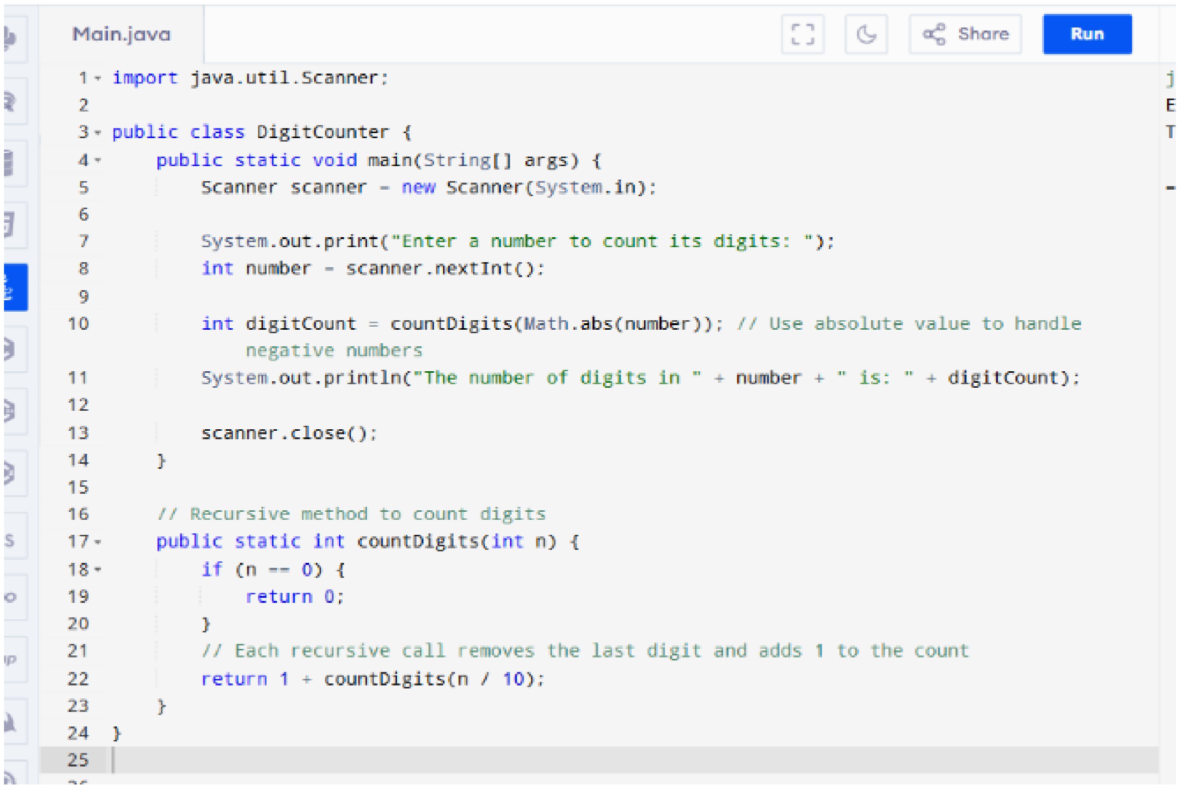
Output

```
java -cp /tmp/EnFuU7XGC2/FibonacciMemoization
Enter the position (N) of the Fibonacci series: 2
The 2-th term in the Fibonacci series is: 1

=== Code Execution Successful ===
```

2023F-BSE-044
ZAINAB AHMED

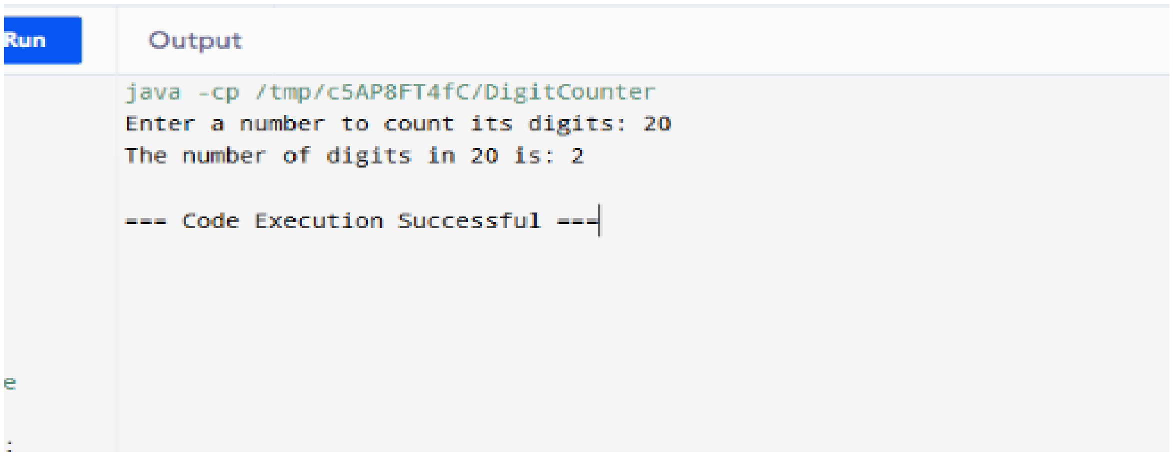
2. Write a program to count the digits of a given number using recursion

CODE:



```java
import java.util.Scanner;

public class DigitCounter {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a number to count its digits: ");
        int number = scanner.nextInt();

        int digitCount = countDigits(Math.abs(number)); // Use absolute value to handle
            negative numbers
        System.out.println("The number of digits in " + number + " is: " + digitCount);

        scanner.close();
    }

    // Recursive method to count digits
    public static int countDigits(int n) {
        if (n == 0) {
            return 0;
        }
        // Each recursive call removes the last digit and adds 1 to the count
        return 1 + countDigits(n / 10);
    }
}
```

OUTPUT:

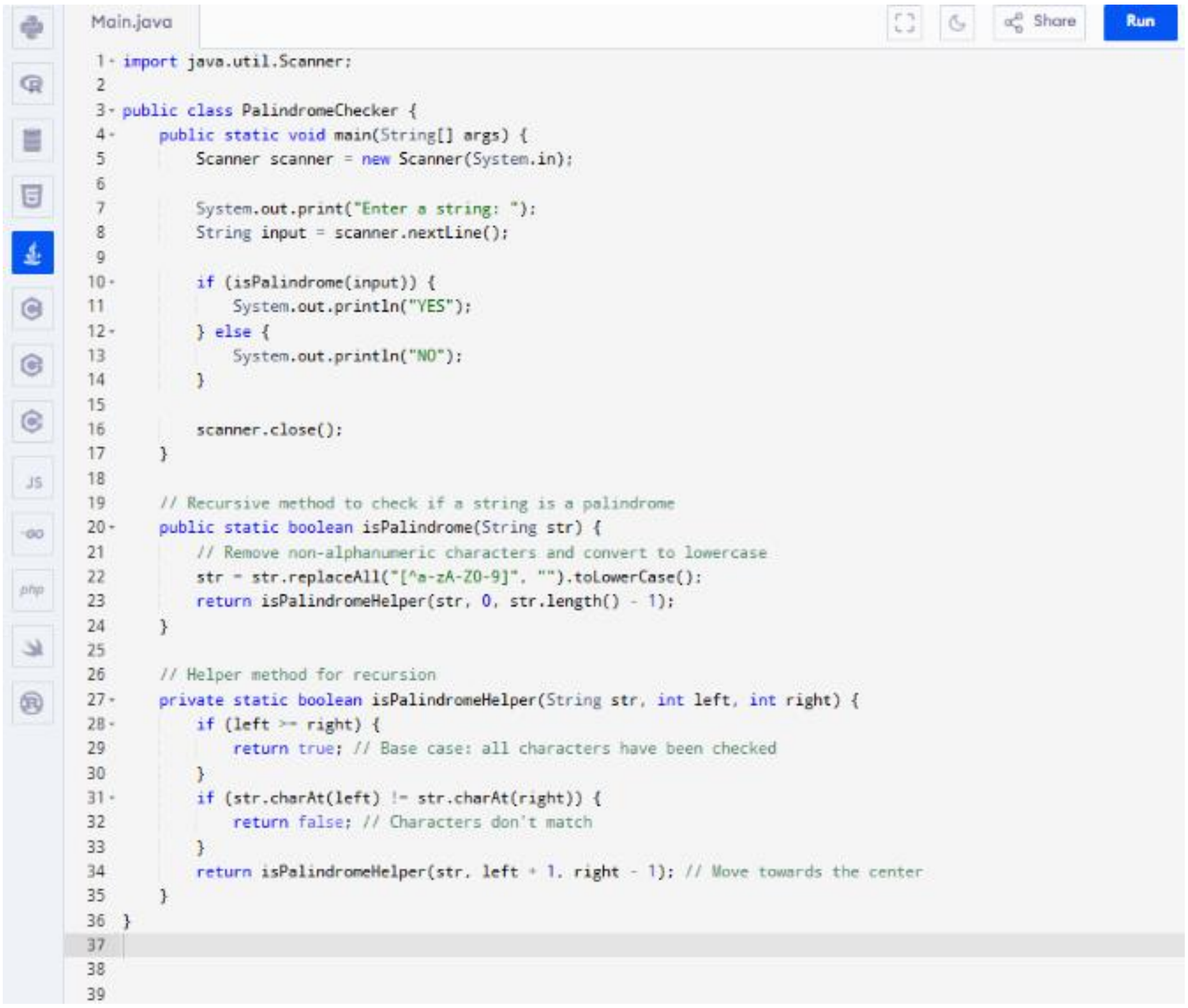


```
java -cp /tmp/c5AP8FT4fC/DigitCounter
Enter a number to count its digits: 20
The number of digits in 20 is: 2

=== Code Execution Successful ===
```

2023F-BSE-044
ZAINAB AHMED

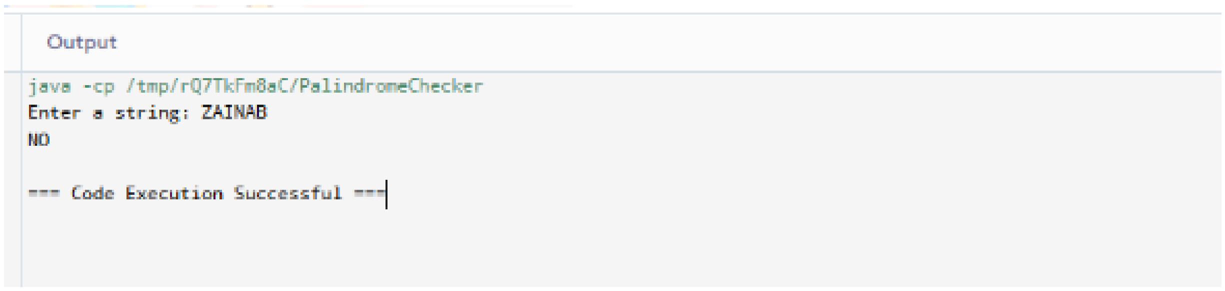
3. Write a java program to check whether a given string is a palindrome or not. A palindrome is a string that reads the same forwards and backwards.Print "YES" if the string is a palindrome, otherwise print "NO".

CODE:



```java
import java.util.Scanner;

public class PalindromeChecker {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a string: ");
        String input = scanner.nextLine();

        if (isPalindrome(input)) {
            System.out.println("YES");
        } else {
            System.out.println("NO");
        }

        scanner.close();
    }

    // Recursive method to check if a string is a palindrome
    public static boolean isPalindrome(String str) {
        // Remove non-alphanumeric characters and convert to lowercase
        str = str.replaceAll("[^a-zA-Z0-9]", "").toLowerCase();
        return isPalindromeHelper(str, 0, str.length() - 1);
    }

    // Helper method for recursion
    private static boolean isPalindromeHelper(String str, int left, int right) {
        if (left >= right) {
            return true; // Base case: all characters have been checked
        }
        if (str.charAt(left) != str.charAt(right)) {
            return false; // Characters don't match
        }
        return isPalindromeHelper(str, left + 1, right - 1); // Move towards the center
    }
}
```

OUTPUT:

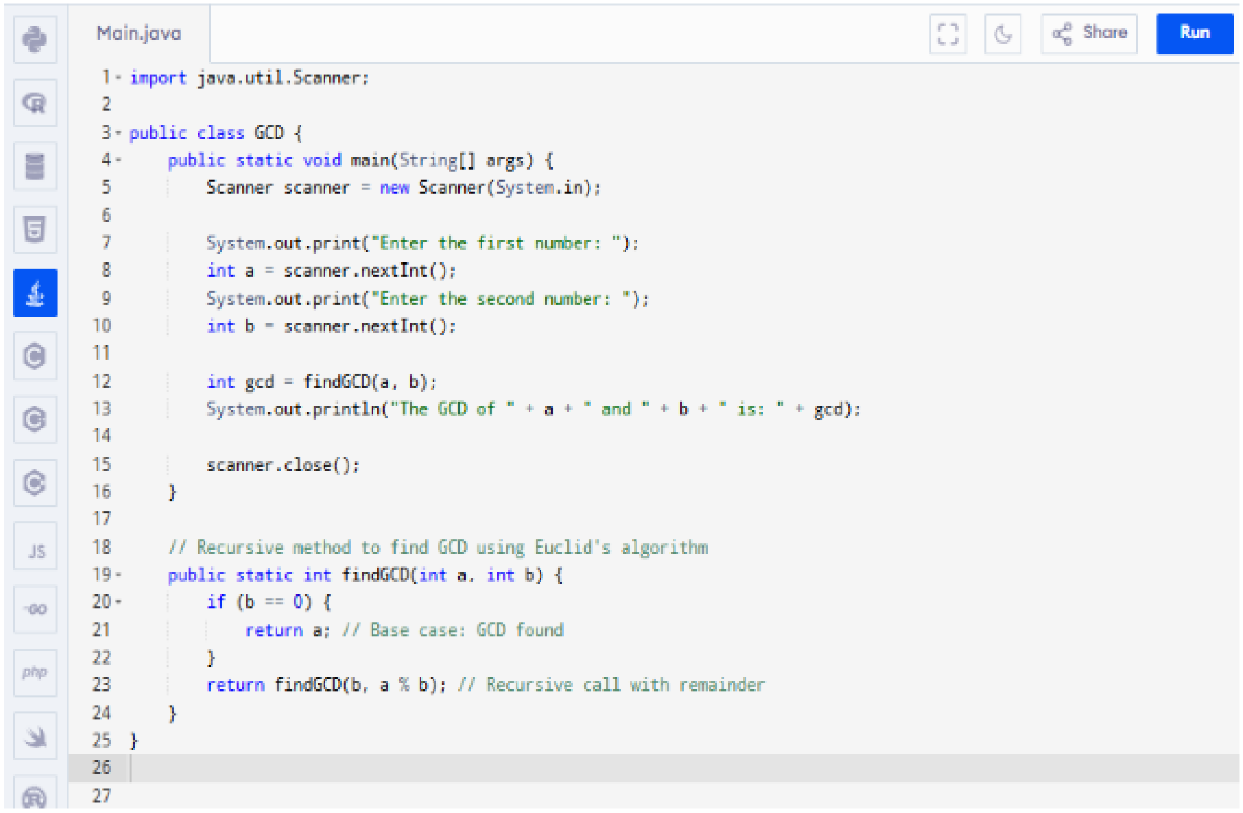


```
java -cp /tmp/rQ7TkFm8aC/PalindromeChecker
Enter a string: ZAINAB
NO

=== Code Execution Successful ===
```

2023F-BSE-044
ZAINAB AHMED

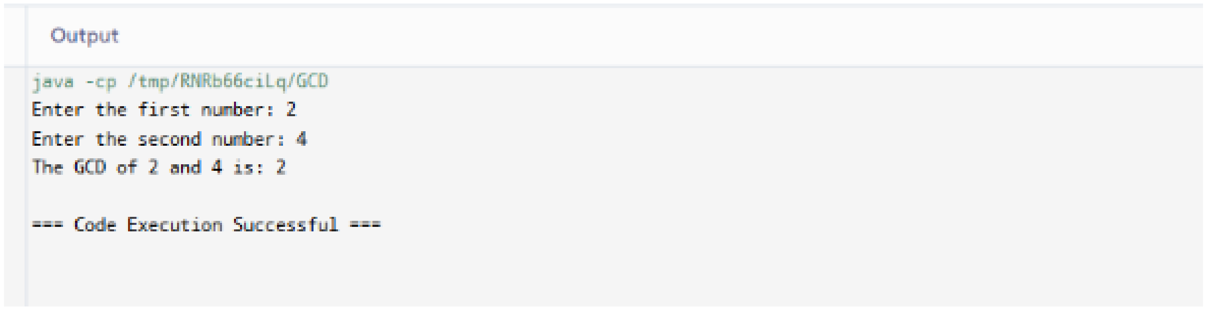
4. Write a recursive program to find the greatest common divisor (GCD) of two numbers using Euclid's algorithm.

CODE:



```java
import java.util.Scanner;

public class GCD {
    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);

        scanner.close();
    }

    // Recursive method to find GCD using Euclid's algorithm
    public static int findGCD(int a, int b) {
        if (b == 0) {
            return a; // Base case: GCD found
        }
        return findGCD(b, a % b); // Recursive call with remainder
    }
}
```

OUTPUT:



```
java -cp /tmp/RNRb66ciLq/GCD
Enter the first number: 2
Enter the second number: 4
The GCD of 2 and 4 is: 2

=== Code Execution Successful ===
```

2023F-BSE-044
ZAINAB AHMED