Task 6: Searching for a Sequence in a Stack

Given a stack and a smaller array representing a sequence, write a function that determines if the sequence is present in the stack. Consider the sequence present if, upon popping the elements, all elements of the array appear consecutively in the stack.

Java Function: -

```java
package com.wipro.assignment;
import java.util.Stack;

public class SequenceInStack {

    public static boolean isSequenceInStack(Stack<Integer> stack,
int[] sequence) {
        // Create a temporary stack to reverse the original stack
        Stack<Integer> tempStack = new Stack<>();

        // Reverse the original stack into the temporary stack
        while (!stack.isEmpty()) {
            tempStack.push(stack.pop());
        }

        // Reset the original stack
        while (!tempStack.isEmpty()) {
            stack.push(tempStack.pop());
        }

        // Check for the sequence in the original stack
        for (int i = sequence.length - 1; i >= 0; i--) {
            if (stack.isEmpty() || stack.pop() != sequence[i]) {
                return false;
            }
        }

        return true;
```

```java
    }

    public static void main(String[] args) {
        Stack<Integer> stack = new Stack<>();
        stack.push(1);
        stack.push(2);
        stack.push(3);
        stack.push(4);
        stack.push(5);

        int[] sequence = {3, 4, 5};

        System.out.println(isSequenceInStack(stack, sequence)); // Should print true

        int[] sequence2 = {2, 3, 5};
        System.out.println(isSequenceInStack(stack, sequence2)); // Should print false
    }
}
```
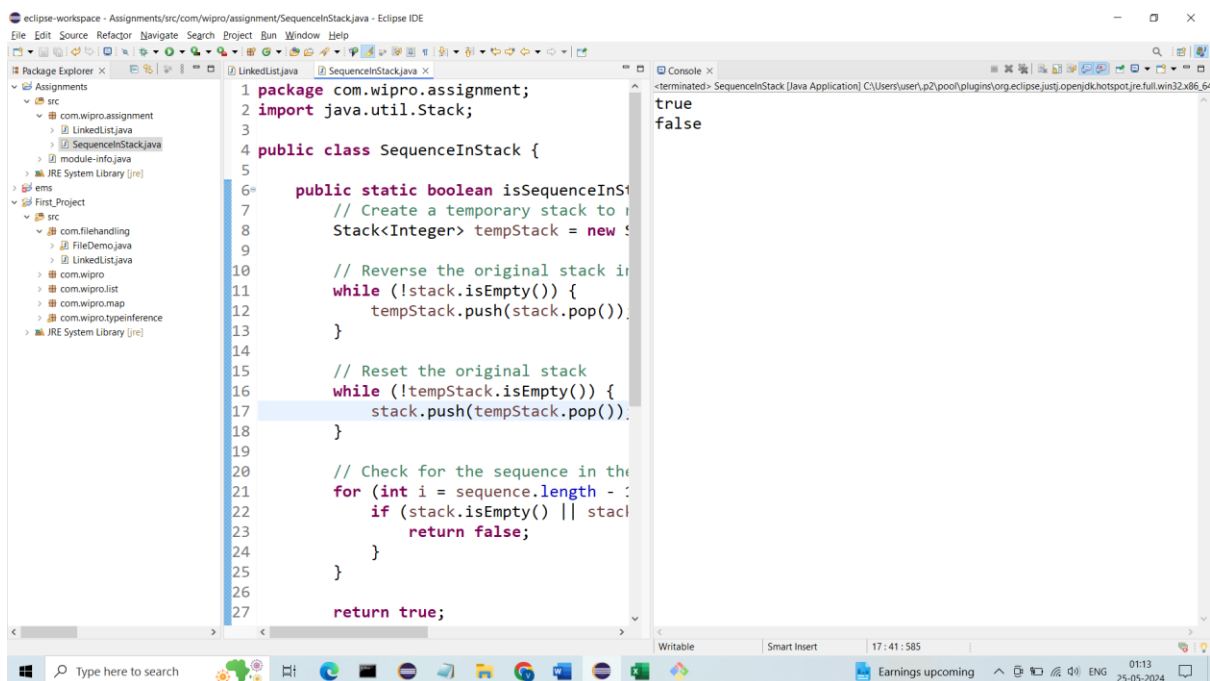
Output-:

Explanation-:

Reverse the Stack: We first reverse the stack using a temporary stack. This helps us process the elements in the order they were pushed.

Check the Sequence: After reversing the stack, we check if the elements in the sequence array match the elements popped from the stack, in reverse order.

Main Function: The main function is provided to demonstrate the use of the isSequenceInStack function with a sample stack and sequences.