**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.**

**Sol:**

**Program:**

import java.util.Stack;

public class StackSequenceChecker {

public static boolean isSequenceInStack(Stack<Integer> stack, int[] sequence) {

// Start checking from the end of the sequence

int sequenceIndex = sequence.length - 1;

// Iterate over the stack from the top

while (!stack.isEmpty() && sequenceIndex >= 0) {

if (stack.peek() == sequence[sequenceIndex]) {

// Element matches, pop from the stack and move to the next element in the sequence

stack.pop();

sequenceIndex--;

} else {

// Element does not match, continue to the next element in the stack

stack.pop();

}

}

// If sequenceIndex is -1, all elements in the sequence were found consecutively in the stack

return sequenceIndex == -1;

}

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};

boolean result = isSequenceInStack(stack, sequence);

System.out.println("Is the sequence present in the stack? " + result); // Output should be true

stack.clear();

stack.push(1);

stack.push(2);

stack.push(4);

stack.push(5);

stack.push(3);

result = isSequenceInStack(stack, sequence);

System.out.println("Is the sequence present in the stack? " + result); // Output should be false

}

}