Name sineha darshan talreja

Cs231237

Dsa lab assignment 6

Q1)

```
import java.util.Stack;

public class StackOperations {
  public static String reverseString(String input) {
    Stack<Character> stack = new Stack<>();
    for (char ch : input.toCharArray()) {
      stack.push(ch);
    }

    StringBuilder reversed = new StringBuilder();
    while (!stack.isEmpty()) {
      reversed.append(stack.pop());
    }

    return reversed.toString();
  }
   public static int reverseNumber(int number) {
    Stack<Integer> stack = new Stack<>();
    while (number > 0) {
      stack.push(number % 10);
      number /= 10;
    }
    int reversed = 0;
    int place = 1;
    while (!stack.isEmpty()) {
      reversed += stack.pop() * place;
      place *= 10;
    }

    return reversed;
  }
   public static int searchElement(Stack<Integer> stack, int element) {
        int positionFromTop = stack.search(element);

    // If element not found, return -1
    if (positionFromTop == -1) {
      return -1;
    }
else
      return stack.size() - positionFromTop;
    }
  }

  // 4. Create a peek() method that returns but does not remove the top of the stack
  public static int peek(Stack<Integer> stack) {
    if (stack.isEmpty()) {
      throw new IllegalStateException("Stack is empty");
    }
    return stack.peek();
  }
  public static void main(String[] args) {
     String originalString = "hello";
    System.out.println("Original String: " + originalString);
    System.out.println("Reversed String: " + reverseString(originalString));
     int originalNumber = 12345;
    System.out.println("\nOriginal Number: " + originalNumber);
    System.out.println("Reversed Number: " + reverseNumber(originalNumber))
    Stack<Integer> stack = new Stack<>();
    stack.push(10);
    stack.push(20);
    stack.push(30);
    stack.push(40);
    int elementToSearch = 30;
    System.out.println("\nStack: " + stack);
    int index = searchElement(stack, elementToSearch);
    System.out.println("Position of " + elementToSearch + " in stack: " + (index == -1 ? "Not Found" : index));
     System.out.println("\nTop element using peek(): " + peek(stack));
    System.out.println("Stack after peek(): " + stack); // Stack remains unchanged
  }
}
```