

## Exercise03 # Stack (21 marks)

### The problem defined

- Code for this exercise are provided with this document.

1. (7 marks) Create a class StackUtility. This class is a utility class that can manipulate a given stack. Write method

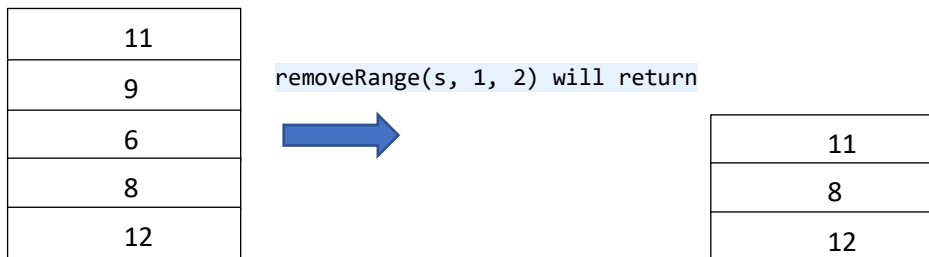
```
public static MyStack removeRange(MyStack s, int i, int j) throws Exception {
```

This method removes data from position i to j (inclusive) from the stack (the stack must actually change).

- Let position 0 be the top of the stack.
- You can assume that the stack has at least 1 data.
- You can assume i and j never indicate positions outside the stack, and i comes before j.
- i and j can be the same position.
- This method must work on both StackArray and StackLinkedList.
- **This method can only use methods originally defined in MyStack (you can call constructor though). You are not allowed to modify StackArray and StackLinkedList in order to complete this question (you get 0 mark if you do not follow this instruction).**

Example:

If your stack, s, originally looks like:



2. (14 marks) In this question we are actually going to write the removeRange method directly in class StackLinkedList and StackArray. The actual stack object must be modified.

```
public void removeRange(int i, int j) (can throws Exception in the linked list version)
```

- Let position 0 be the top of the stack.
- You can assume that the stack has at least 1 data.
- You can assume i and j never indicate positions outside the stack, and i comes before j.
- i and j can be the same position.
- **Do not change the JUnit file.**
- **Do not call methods define in MyStack. You must manipulate the array or linked list directly.**
- **If you don't follow this instruction in bold, you get 0 mark.**

How to submit

- Submit your modified StackLinkedList, StackArray, and StackUtility zipped together on MyCourseville (as an attachment to this assignment).
- JUnit Test cases used when marking will be different from the JUnit you have, but they test the same logic. So make sure your program works for any possible inputs.