

**CSE 4304-Data Structures Lab. Winter 2022-23**

**Date:** 05 Sep 2023

**Target Group:** SWE B

**Topic:** Stack

**Instructions:**

- Regardless of when you finish the tasks in the lab, you have to submit the solutions in the Google Classroom. The deadline will always be at 11.59 PM of the day in which the lab has taken place.
- Task naming format: <fullID>\_<Task><Lab><Group>.c/cpp. Example: 170041034\_T01L02A.cpp
- If you find any issues in the problem description/test cases, comment in the google classroom.
- If you find any test case that is tricky that I didn't include but others might forget to handle, please comment! I'll be happy to add.
- Use appropriate comments in your code. This will help you to easily recall the solution in the future.
- Obtained marks will vary based on the efficiency of the solution.
- Do not use the <bits/stdc++.h> library.

**Task-01:**

Implementing the basic operations of Stack

Stacks is a linear data structure that follows the Last In First Out (LIFO) principle. The last item to be inserted is the first one to be deleted. For example, you have a stack of trays on a table. The tray at the top of the stack is the first item to be moved if you require a tray from that stack.

The Insertion and Deletion of an element from the stack are a little bit different from the traditional operation. We define the two corresponding operations as Push() and Pop() from the stack.

The first line contains  $N$  representing the size of the stack. The lines contain the 'function IDs' and the required parameter (if applicable). Function ID 1, 2, 3, 4, 5, and 6 corresponds to push, pop, isEmpty, isFull (assume max size of Stack=5), size, and top. The return type of isEmpty and isFull is Boolean. Stop taking input once given -1.

Input	Output
5	
3	True
2	Underflow
1 10	10
1 20	10 20
5	2
1 30	10 20 30
6	30
2	10 20
1 40	10 20 40
1 50	10 20 40 50
4	False
1 60	10 20 40 50 60
4	True
5	5
1 60	Overflow
5	5
2	10 20 40 50
6	50
-1	

**Task 02:****Checking parenthesis in Mathematical Expressions**

Write a program that will take a mathematical expression as input and check whether it is properly parenthesized or not.

The first line of input will take an integer **N** signifying the number of test cases. The next lines will be **N** mathematical expressions. Each input expression may contain any single-digit number (0~9), operators (+ - x /) and any parenthesis ( )/[ ]/{ }.

The output will be Yes/No representing whether it is properly parenthesized.

Sample Input	Sample Output
8	Yes
[ 5 + ( 2 x 5 ) - ( 7 / 2 ) ]	No
[ 1 + { 3 x ( 2 / 3 ) } ] }	Yes
[ ( 1 + 1 ) ]	No
[ ( 1 + 1 ] )	Yes
[ ( ) ] { } { [ ( ) ( ) ] ( ) }	No
( ( (	No
[ 5 + ( 2 x 5 ) - ( 7 / 2 )	No
5 + ( 2 x 5 ) - ( 7 / 2 ) ]	No
( ) ) )	No
( ( ( ) )	

You can use STL Stack for this task.

**Task 03**

You are given a string of lower case English letters. You have to repeatedly remove adjacent equal letters so that after removal no two adjacent letters are equal.

First line the number of input test case **N** is given followed by **N** strings.

Try to make the running time complexity  $O(n)$ . You can use STL Stack.

Sample Input	Sample Output
5	
aaabbccc	[ac]
abcdddc	[abcdc]
aabbcc	[]
eabccbad	[ed]
eabccbadeabccbad	[eded]

**Task 04:**

In a round table conference there are  $N$  people sitting in a circle. You are given the rank of these people. For each person you have to find who is the next senior by rank going in a clockwise direction.

For person  $x$  to be the senior of person  $y$ , the  $\text{rank}[y] < \text{rank}[x]$ . If there is no senior for some persons mark it as  $-1$ ;

For each line the input is terminated by  $-1$ .

Try to make the running time complexity  $O(n)$ . You can use the STL stack.

Input	Output
4 5 2 25 -1	5 25 25 -1
13 7 6 12 -1	-1 12 12 13
11 13 21 3 20 -1	13 21 -1 20 21
12 17 1 5 0 2 2 7 18 25 20 12 5 1 2 -1	17 18 5 7 2 7 7 18 25 -1 25 17 12 2 12
10 20 30 40 50 -1	20 30 40 50 -1
50 40 30 20 10 -1	-1 50 50 50 50