The objective of this lab is to:

understand and find LIFO behavior in different problems.

ALERT!

1. This is an individual lab, you are strictly **NOT** allowed to discuss your solution with fellow colleagues, even not allowed asking how is he/she is doing, it may result in negative marking. You can **ONLY** discuss with your TAs or with me.
2. You are advised to attempt Task 01 before other tasks i.e. implement **STACK ADT** first, then Use that ADT in all the following lab tasks.
3. Pay attention to **GOOD coding conventions** e.g.
   - Proper indentation.
   - Meaning variable and function names.
   - Use camelCase naming convention
   - Use meaningful prompt lines/labels for all input/output
4. **Anyone caught in act of plagiarism would be awarded an "F" grade in this Lab.**

Task 01: [0 Marks]

Implement stack ADT using Templates.

Task 02: [10 Marks]

Given a string *s* containing just the characters *'(', ')', '{', '}', '[' and ']'*, determine if the input string is valid. An input string is valid if:

- Open parenthesis must be closed by the same type of parenthesis.
- Open parenthesis must be closed in the correct order.

Examples:

{ { } } ( )[( )]	valid
{] [}	Invalid
( )(	Invalid

Task 03: [10 Marks]

Given a string S of '(' and ')' parentheses, we add the minimum number of parentheses ('(' or ')', and in any positions) so that the resulting parentheses string becomes valid.
Formally, a parentheses string is valid if and only if:

- It is the empty string, or
- It can be written as AB (A concatenated with B), where A and B are valid strings, or
- It can be written as (A), where A is a valid string.

Given a parentheses string, return the minimum number of parentheses we must add to make the resulting string valid.

Examples:

Parenthesis string	number of parenthesis needed
( ))	1
(((	3

( )	0
( ))((	4

Task 04: [10 Marks]

Given a chemical formula (given as a string), return the count of each atom.
An atomic element always starts with an uppercase character, then zero or more lowercase letters, representing the name. 1 or more digits representing the count of that element may follow if the count is greater than 1. If the count is 1, no digits will follow. For example, H2O and H2O2 are possible, but H1O2 is impossible. Two formulas concatenated together produce another formula. For example, H2O2He3Mg4 is also a formula. A formula placed in parentheses, and a count (optionally added) is also a formula. For example, (H2O2) and (H2O2)3 are formulas.

Given a formula, output the count of all elements as a string in the following form: the first name (in sorted order), followed by its count (if that count is more than 1), followed by the second name (in sorted order), followed by its count (if that count is more than 1), and so on.

Example 1:

Input formula = "H2O"
Output: "H2O"

Explanation:
The count of elements is {'H': 2, 'O': 1}.

Example 2:

Input formula = "Mg(OH)2"
Output: "MgO2H2"

Explanation:
The count of elements are {'Mg': 1, 'O': 2, 'H': 2}.

Example 3:

Input formula = "K4(ON(SO3)2)2"
Output: "K4N2O14S4"

Explanation:
The count of elements are {'K': 4, 'N': 2, 'O': 14, 'S': 4}.

All atom names consist of lowercase letters, except for the first character which is uppercase. The length of formula will be in the range [1, 1000]. formula will only consist of letters, digits, and round parentheses, and is a valid formula as defined in the problem. Given a chemical formula, count the number of atoms of each atom.