Assignment # 2
Data Structures & Algorithms

Due Date: 03-10-2021 (10pm) **Marks: 50**

Submission Guidelines:

- Submit your assignment only on google classroom within deadline.
- Late submissions of upto 15 mins are allowed with a penalty of 20% on google classroom.
- Add a comment section atop of your code files, mentioning your name roll number etc, and a bit about what you did in this code. Follow the detailed guidelines shared by the TA.
- Name/rename your submission files as "Section-roll#-Assig#-filenmae.cpp", e.g., "A-20i0899-A2-main.cpp".
- Zero marks will awarded in following cases
  - Code with build errors
  - Submitting assignment through email, slate, or any other way other than the google classroom
  - Submitting after the late deadline

Stack Applications

In this assignment, you will use **Template** Stack (5marks) to evaluate mathematical expressions

Create a program that should take input a mathematical expression, evaluate it, and print the answer.

The menu should be:
- Input fully parenthesized infix expression
- Input postfix expression (put space after every operand or operator)

If infix expression is input then:
- Check for any parenthesis error, and check if expression is fully parenthesized or not
- Evaluate and display its result

The mathematical expression can contain
- Numbers as operands (numbers can be of single or multiple digits)
- Operators (+ , - , * , / , ^)
- Any type of parenthesis

^ exponent operator

In postfix notation, numbers should be separated from each other with a space to avoid any confusion. For example (32 – 4) when converted to postfix notation should NOT be written as 324- , but as 32 4 -

Following are some examples, as to how your program should behave with different expressions.

```
Input: [{3+(45/5)}-6]
Output: parenthesis OK
Answer: 6
```

```
Input:  (13*(6-4)) / 8)
Output: parenthesis problem

Input:  13*(6-4) / 8
Output: Not fully parenthesized

Input: 32 3 + 4 -
Output: 31
```