COP 3502C Summer 2019
Midterm Assignment # 1

Total points: 13

Introduction: For this assignment you have to write a c program that will take an infix expression as input and display the postfix expression of the input. After converting to the postfix expression, the program should evaluate the expression from the postfix and display the result.

What should you submit?
Write all the code in a single file and upload the .c file.

Please include the following commented lines in the beginning of your code to declare your authorship of the code:

```
/* COP 3502C Midterm Assignment One
This program is written by: Your Full Name */
```

Compliance with Rules: UCF Golden rules apply towards this assignment and submission. Assignment rules mentioned in syllabus, are also applied in this submission.The TA and Instructor can call any students for explaining any part of the code in order to better assess your authorship and for further clarification if needed.

Problem

We as humans write math expression in infix notation, e.g. 5 + 2 (the operators are written in-between the operands). In computer's language, however, it is preferred to have the operators on the right side of the operands, i.e. 5 2 +. For more complex expressions that include parenthesis and multiple operators, a compiler has to convert the expression into postfix first and then evaluate the resulting postfix.

Write a program that takes an "infix" expression as input, uses stacks to convert it into postfix expression, and finally evaluates it. It must support the following operations:

+ - / * ^ % (

Example

Infix expression: (7 - 3) / (2 + 2)

Postfix expression: 7 3 - 2 2 + /

Result: 1

Rubric:

1) If code does not compile in Eustis server: 0.
2) Checking the balance of the parenthesis: 2 point
3) Incorrect postfix expression per test case: -2 points
4) Correct postfix but incorrect evaluation per test case: -1 points
5) Handling single digit inputs: maximum 11 point
6) Handling two-digit inputs: 100 percent (if pass all test cases)

Read it: Some hints (but it is not the only way)

1. Use the uploaded multi stack code for stack implementation and modify the code as you need.
2. You will need to use stacks in three places
    a. One for the parenthesis check [char stack]
    b. One during infix to postfix [char stack]
    c. One during evaluation [int stack]
    For a and b above, you can use same array and same push, pop method as both of them are char. But for evaluation you have int stack and you might consider to create another push pop method to handle it. Maybe push_int, pop_int, etc. Or find other strategy to utilize existing push pop method
3. You can create **a function** for obtaining operator priority. That function should take an operator as input and return its priority as an integer. This function will help you a lot and reduce repeated code

4. During evaluation you will need to convert char into integer. Example for single digit:

```
char c = '5';
int x = c - '0';
```

See uploaded code String2IntegerExample.c file for more example and testing.

Please see the lecture slides for the explanation, steps and more test examples.

Late submission will not be accepted. An assignment submitted by email will not be graded and will not be replied.

Good Luck.