Lab Sheet 6

Lab Tasks:

A. Convert an infix expression to postfix.

The algorithm that you should follow:

Algorithm: Infix-to-Postfix (Q, P)

Here Q is an arithmetic expression in infix notation and this algorithm generates the postfix expression P using stack.

1. Scan the infix expression Q from left to right.
2. Initialize an empty stack.
3. Repeat step 4 to 5 until all characters in Q are scanned.
4. If the scanned character is an operand, add it to P.
5. If the scanned character is an operator Φ, then
    - (a) If stack is empty, push Φ to the stack.
    - (b) Otherwise repeatedly pop from stack and add to P each operator which has the same or higher precedence than Φ.

    - (c) Push Φ to the stack.
6. If scanned character is a left parenthesis "(", then push it to stack.
7. If scanned character is a right parenthesis ")", then
    - (a) Repeatedly pop from stack and add to P each operator until "(" is encountered.
    - (b) Remove "(" from stack.
8. If all the characters are scanned and stack is not empty, then
    - (a) Repeatedly pop the stack and add to P each operator until the stack is empty.
9. Exit.

Sample input/output:

Case 1:

```
Enter the infix expression : a+b*c

The corresponding postfix expression is: a b c * +
```

Case 2:

```
Enter the infix expression : ((4+8)(6-5))/((3-2)(2+2))
The corresponding postfix expression is: 4 8 + 6 5 - 3 2 - 2 2 + /
```

Reading materials:

Primary idea about this topic:

https://www.geeksforgeeks.org/convert-infix-expression-to-postfix-expression/