Assignment #2

Due: 10/17 23:59 KST

Introduction

In the world of mathematical notations, expressions can be represented in various formats, with ‘infix’ and ‘postfix’ notations being two commonly used forms. In this assignment, we will explore the process of converting infix expressions into postfix expressions using C++.

Infix Notation: Infix notation is the conventional mathematical notation we encounter in our daily lives. In this notation, operators are placed between operands, for example, "3 + 4." While infix notation is intuitive for humans, it may require additional rules to evaluate correctly due to operator precedence and parentheses.

Postfix Notation: In this notation, operators are placed after their operands, for example, "3 4 +." Postfix notation is characterized by its simplicity and the absence of the need for operator precedence rules or parentheses. It is particularly favored in computer science and calculators because it is easy to process and evaluate.

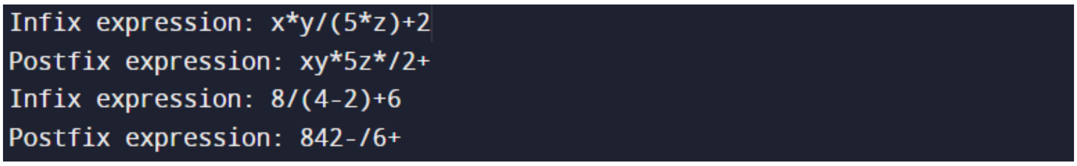
Implementation

The objective of this assignment is to create a C++ program that converts infix notation to postfix notation using STL stack.

Assignment Requirements:

1. The program receives the infix repeatedly until it ends.
2. For each iteration, ask the user to enter an infix notation expression.
3. Convert the entered infix expression to postfix expression and display the result.
4. Operands in the expression should include uppercase letters ‘A’ to ‘Z’, lowercase letters ‘a’ to ‘z’, and integer values.
5. Operators in the expression should include ‘(‘, ‘)’, ‘+’, ‘-‘, ‘*’, and ‘/’.
6. Remember to implement your code using the STL stack covered in the lab session.

Example:



```
Infix expression: x*y/(5*z)+2
Postfix expression: xy*5z*/2+
Infix expression: 8/(4-2)+6
Postfix expression: 842-/6+
```

Submission

Name the C++ source code file according to the format (student number-name.cpp) and include simple comments explaining your code.

e.g., 202X-2XXXX-Gildong Hong.cpp