**Program Calculator**

Due: PM:5:00, May 19, 2021

* **Program design target:**  
  In lecture 3, we introduced stacks and queues that can store same object data in different order. The stack is used to provide first-in-last-out function. In this program homework, you have to firstly implement the template stack data structure as we have introduced the template queue class in the lecture hours. Then the infix to postfix (page52-page57) and evaluation (page 51) functions that we introduced in lecture notes can be used to evaluate the value of an infix expression. I write a main.cpp shown below for you reference to develop your own programs.

**#include "stack.h"**

**#include "postfix.h"**

**float Eval(vector <Token> & );**

**int main() {**

**string infix = "((4^(3-1)+2)\*3)/(8-4)";**

**char cont;**

**do {**

**system("CLS");**

**cout << "Demo program of a NTUST 2021 calculator by J.-J. Chen\n";**

**cout << "Enter the infix expression:\nINFIX: ";**

**cin >> infix;**

**vector <Token> pfix;**

**pfix = postfix(infix);**

**cout << "\nEvaluate" << infix << '=' << Eval(pfix) << endl;**

**cout << "\n\nContinue?(Y/N) ";**

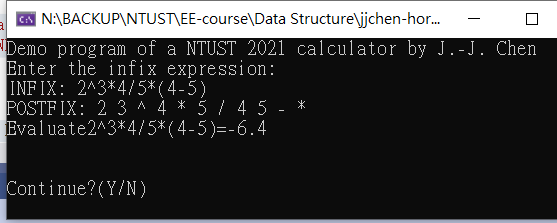
**cin >> cont;**

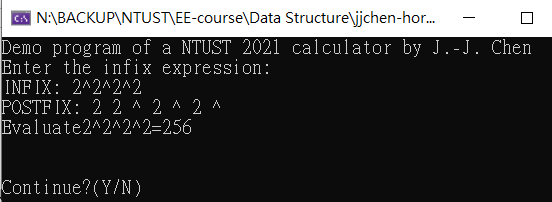
**} while (cont == 'Y' || cont == 'y');**

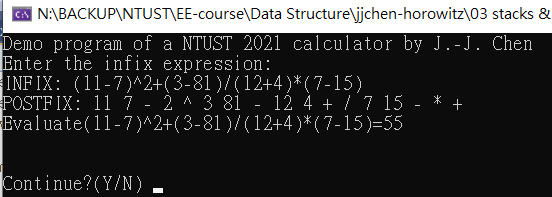
**return 0;**

**}**

* An executable calculator.exe is provided for you to verify the calculator function. Some execution results are shown below.







* **What to do: (75%)**

Some functions are partially finished and you are asked to make the program complete. The execution result is shown in the above figure.

1. You will have to submit the complete project such that the TA can recompile your programs to test correctness.
2. You have to write a short report to describe
   1. What is all about the program?
   2. What functions you have designed to provide a calculator program?
   3. How will you improve this program?

* **Criteria**

1. (65%) Basic +, -, \*, / operations are correct
2. (15%) infix expressions that comprise (, ), ^, can be evaluated correctly.
3. (10%) unary operation such as (-3) can be recognized and evaluated.
4. (10%) Correctness check of the infix expression.

Hint:

1. 在處理輸入讀取infix expression時，應該使用 getline(cin, infix)

而不是 cin >> infix.

如果使用第二種方法，那麼 3\*4 + 5會只讀到第一個空字元之前的3\*4。

1. 此infix expression首先必須tokenize，也就是35+27必須分為35(operand)， ‘+’(operator)，和27(operand)三個tokens，tokenize的過程可以從網路上找到。
2. 本程式主要是要處理operator的優先權問題，以及’(‘和’)’的處理方法，另外配合課本所提供evaluate和infix2postfix程式中有須要針對結束符號’#’的優先權做額外處理。請特別注意此一符號如何解決算術的程式流程控制問題。