3460:316 Data Structures **Spring 2018**
Dr. C.-C. Chan
Project #1 (40pts) **Due 2/14/2018**

Problem: Write a C++ program to convert infix to postfix expression and evaluate the resulting postfix expression by using Stack ADT.

- An infix expression may have nested parentheses.
- Operator tokens that may appear in an expression are '+', '−', '*', '/', '(', and ')' with orde of precedence as defined in C++.
- Operand tokens are assumed to be either integer literals or single characters in 'a' .. 'z' or 'A' .. 'Z'.
- Each input expression is ended with the character '#'.
- Here are two examples of input expressions: "a + b #" and "a + 100 * b #"
- Expressions are assumed to be in correct forms.
- Use a list to store input expression tokens.

Inputs/Outputs:

1. Take interactive inputs from the keyboard. Your program should start with a menu of th following items:
   (1) Convert Infix to Postfix
   (2) Evaluate Expression
   (3) Exit
2. Menu item (1): ask the user to enter an infix expression. Your program will print the infix expression and the converted postfix expression. Note: input expressions may hav mixed operands of letter and integer literals. E.g.: the expression a + 100 * b # is a valid input.
3. Menu item (2): will take the input infix expression entered in menu (1) by converting it to postfix first. Your program needs to ask the user to enter integer values for operands which are letter tokens, then evaluate the expression.

Outputs: print all results to screen.
Testing: create your testing expressions.

Submissions:

- Name your source files as **uid_316p1.cpp** and **uid_316p1.h** where uid is the last four digi of your UA ID.
- Submit your source files thru BrightSpace.
- The project is due by midnight of the due date.