EE441 HOMEWORK 2

Queue and Stack based Calculator

Due: December 4, 2017, 23:55

Ferhat Gölbol	**D-227**	**ferhatg@metu.edu.tr**
Kamil Sert	**D-215**	**ksert@metu.edu.tr**
Aycan Doğa Hakyemez	**ARC-201**	**hdoga@metu.edu.tr**

***You may ask HW#2 related questions to ksert@metu.edu.tr**

Introduction

In this homework you are asked to implement a user interface for the calculator using stack and queue data structures. Your calculator must support the following operators:

- Addition: +
- Subtraction: -
- Multiplication: *
- Division: /
- Modulo: %
- Exponent: ^

The input will be in the form of infix expressions. For evaluation of the expression; if there is an innermost expression, i.e. expression in parenthesis, "(….), [..(….)..], begin with the calculation of it. Until there are no innermost expressions do the previous calculation. If there is no innermost expression calculate the final result.

Example:

Infix expression:	[(4 + 6 ÷2) + 1] * 3
Evaluation step 1:	(4 + 6 ÷ 2) = 7
current expression :	[7 + 1]* 3
Evaluation step 2:	[7 + 1] = 8
current expression :	8* 3
Evaluation step 3:	
result = 24	

Calculator must work with all integers (for example 7/2=3). Expressions entered can be invalid, so you have to check it before calculating the result (you can simply control if parentheses are matching, if operators are valid, and if operands contain only numerals). For calculation step you can only use at most 1 stack and any number of queues. Calculator design must be your original work.

Implement the queue and the stack in separate classes with all required methods. Also you have to consider dynamic memory management for this homework. Make sure that there is no memory leak in your program.

* Using Postfix Notation or Reverse Polish Notation (RPN) are not allowed!

User Interface

- Functions:
  - enterExpression (asks for the Expression from user, and writes to the memory)
  - checkExpression (checks whether the form of Expression is valid or not)
  - evaluateExpression (calculates the result by evaluating the expression and returns the result)
  - resetCalculator(deletes all data structures)

The program should ask the user for his/her input by offering different options as a menu:

- Enter an Expression
- Check the Expression
- Calculate
- Reset
- Exit

Remarks

- You should check for illogical input (e.g. characters in the operation or selecting Calculate before entering an Expression) and give appropriate warnings.

Example

```
Queue and Stack Based Calculator

MENU
1-) Enter an expression
2-) Check the expression
3-) Calculate
4-) Reset
5-) Exit

Enter your option: 1
Enter an expression: [(4+6/2)+1]*3

Enter your option: 2
Valid expression!

Enter your option: 3
Result of the expression is 24.

Enter your option: 4
Reset is successful!

Enter your option: 1
Enter an expression: 23/2[9*(2+1]

Enter your option: 2
Invalid expression!

Enter your option: 5

Process returned 0 (0x0)   execution time : 0.030 s
Press any key to continue.
```

Submission

- Use Code::Blocks IDE and choose GNU GCC Compiler while creating your project. Name your project as "e1XXXXXX_HW2" where Xs are the digits of your student ID number. Send the whole project folder compressed in a rar or zip file. Name your submission as e1XXXXXX_ee441_hw2.rar. You will not get full credit if you fail to submit your project folder as required.
- Your C++ program should follow object oriented principles, including proper class and method usage and should be correctly structured including private and public components. Your work will be graded on its correctness, efficiency and clarity as a whole.
- You should insert comments in your source code at appropriate places without including any unnecessary detail.
- Late submissions are welcome, but are penalized according to the following policy:
    - 1 day late submission: HW will be evaluated out of 70.
    - 2 days late submission: HW will be evaluated out of 50.
    - 3 days late submission: HW will be evaluated out of 30.
    - 4 or more days late submission: HW will NOT be evaluated.
- It is **not** allowed to prepare homework as groups. METU honor code is essential.
- Any involvement in any cheating will be graded 0.