Linked List – Stack / Queue

Due Date : Monday October 21st , 2019

no later than 12:45 pm

Write a C++ program that will simulates a ADT stack / queue using a linked List ADT with actual pointers. The program displays a menu on the screen allowing the user to :

A. Insert new element in the stack (LIFO).
B. Remove an element from the stack (LIFO).
C. Insert new element in the queue (FIFO).
D. Insert an element at specific location in the queue.
E. Remove an element from the queue (FIFO).
F. Remove a specific element from the queue.
G. Find Min element in the queue, Max element in the queue, and the Average of the queue.
H. Delete the entire stack.
I. Delete the entire queue.
X. Exit the program.

The entire list must be displayed After completion of selecting A, or B , or C , Or D or E or F or G. The program must continue running until the letter X is entered.

Notes :

- Only capital letters are accepted.
- You design your own output format. However, The program / output must be easy to follow.
- Validation must be done with the repeat process. Only **capital** letters are accepted. Everything else should be rejected
- Must use functions to handle each selection (A – I)
- Must place the following as a footer / signature line before terminating the program.

This LL / Stack & Queue Program is Implemented By :
Husain Gholoom - October 21st , 2019

Style Guidelines:

At the beginning of your program (and before the #include statement), include the following :

Header comments (file documentation block) should be at the top of each file and should contain: Author / s, Due Date, Assignment Number, Course number and section, Instructor, and a brief description of the purpose of the code in the file. For example :

```
//    Serial  Number  / s  :        xxxxxxxxx
//
//    Author  : (Your name  here!!)
//
//    Programming Assignment Number  4
//
//    Fall   2019   -   CS 3358    -    Your Section  Number
//
//    Due Date  :
//
//    Instructor:    Husain Gholoom.
//
//     <Brief description of the  purpose of the program>
```

Variable names :

- Must be meaningful.
- The initial letter should be lowercase, following words should be capitalized, no other caps or punctuation (i.e. `weightInPounds`).
- Each variable must be declared on a separate line with a descriptive comment.

Named constants :

- Use for most numeric literals.
- All capitals with underscores (i.e. `TX_STATE_SALES_TAX`)
- Should occur at top of function, or global (only if necessary)

Line length of source code should be no longer than 80 characters (no wrapping of lines).

Indentation :

- Use 2-4 spaces (but be consistent throughout your program).
- Indent blocks, within blocks, etc.
- Use blank lines to separate sections.

Comments for variables :

All variable definitions should be commented as follows:

```
int  gender;    // integer value for the gender,
              // 1 = Male , 2 = Female ,
```

Rules :

1. Your program **must compile** and run using latest version of codeblocks under windows.
2. The entire program must be **documented according the style above . See the website for the sample programming style program.**
3. You must use the appropriate libraries in writing this program, however, you are **not allowed** to use libraries that support the functionality of linked lists such as list or forward_list. You are also not allowed to use libraries that supports queue. Finally , are also not allowed to use recursion.
4. **Your program must use a class definition f along with appropriate member functions when applicable. You are going to need to use 2 classes, one the stack and one the queue.**

```
struct Node
    {
       int data;
       Node *link;
    };
```

5. You must name your program as :

   - **PG4_F2019_3358_1_LastName_FirstName.cpp (not .cbp) or**

   Where LastName is your Last Name and FirstName is your First Name.

6. **No late assignments will be accepted.** **DO NOT** send your assignment solution via email .

   Use TRACS To upload electronic version of your program

7. You must **also** turn in hard copy of your source code no later than the starting of class time on the due date . should the hard copy consist of more than one page , then , the hard copy must be **stapled**. if you are unable to turn in a printout during class, you can take the program to the computer science department and hand it to the front desk personal (Comal 211) before the deadline. Make sure that the front office stamps the program. Make sure that include the date and time. Finally ,make sure that they place the program in my mailbox.

   DO NOT slide your program under my office door – It will **NOT** be accepted

The following points will be deducted if :

- Incorrect file format such as uploading .cbp instead of .cpp , missing electronic copy , missing the hardcopy using .h and .cpp files , compilation Errors , Using list / forward list libraries, Using arrays or global arrays , vectors or global vectors , not using classes , using recursion … etc **(- 10 points)**

- Other **(at least 1 point each)** :

  - Logical Errors
  - Unable to read the source code due to unclear printing
  - Incorrect program file name.
  - More than one hardcopy per group or Hard copy is not stapled.
  - Not using at least 9 functions.
  - Incorrect Style **such as but not limited to** missing comments or program documentations , missing serial number , missing section number, missing function prototypes if functions are used, missing signature line … etc