Programming Assignment 4

Stacks – Queues

Due Date : Section 0 - Wednesday March 28th , 2018 - No Later than 2:15 pm

Section 1 - Wednesday March 28th , 2018 - No Later than 3:45 pm.

Section 2 - Wednesday March 28th , 2018 - No Later than 5:15 pm.

For this assignment you will implement stacks and queues ADT. A stack or a queue can be used to recognize a certain types of patterns.

Consider the pattern STRING1#STRING2 where neither STRING1 and STRING2 contains "#".

Use stack ADT to determine whether or not the two strings are the same. STRING1 and STRING2 are the same if they have the same number of elements and the elements at the corresponding positions are the same.

Use queue ADT to determine whether or not STRING2 is reversed of STRING1.

The program displays a menu on the screen allowing the user to enter the elements of stack or queue. The program terminates when the user enters 9.

Note :

- **Size of stack and queue is determined by the number of the elements entered.**
- **You are not allowed to use stack / queue libraries. You must implement all stack / queue functions and operations . Use only one .cpp to define and implement templates and the driving program.**
- **You are not allowed to use string manipulation operations .**

Validations:

The program accepts only integer data type when selecting from the menu (1 , 2 , or 9). Everything else should be rejected with an invalid option message.

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 :

```
//       Roster Number / s  :       xxxxxxxxx
//
//       Author / s : (Your name  here!!)
//       Due Date  :
//       Programming Assignment Number  4
//
//       Spring   2018   -   CS 3358   -   Your Section  Number
//
//       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 ,
```

Sample Run :

```
*** Welcome to My stack / Queue Program ***

The function of this program is to :

     1. Use stack to determine whether or not
        two strings are the same.

     2. Use queue to determine whether or not
        STRING2 is the reversed of STRING1.

Select from the following menu
     1.   Enter Stack Values.
     2.   Enter Queue Values.
     9.   Terminate the program.    1

Enter Stack Values :   5+(12+7)#5+(12+7)

Strings are identical


Select from the following menu
     1.   Enter Stack Values.
     2.   Enter Queue Values.
     9.   Terminate the program.    1

Enter Stack Values :  5+(12+7]#5+(12+7)

Strings are  not identical


Select from the following menu
     1.   Enter Stack Values.
     2.   Enter Queue Values.
     9.   Terminate the program.    1

Enter Stack Values :  5+[12+7)#5+[12+7

Strings are  not identical
```

```
Select from the following menu
     1.   Enter Stack Values.
     2.   Enter Queue Values.
     9.   Terminate the program.    2

Enter Queue Values :   1234#4321

STRING2 is reversed of STRING1.


Select from the following menu
     1.   Enter Stack Values.
     2.   Enter Queue Values.
     9.   Terminate the program.    2

Enter Queue Values :   1{[(#([{1

STRING2 is reversed of STRING1.



Select from the following menu
     1.   Enter Stack Values.
     2.   Enter Queue Values.
     9.   Terminate the program.    2

Enter Queue Values :   1{[(#([{2

STRING2 is not reversed of STRING1.
```

```
Select from the following menu
     1.   Enter Stack Values.
     2.   Enter Queue Values.
     9.   Terminate the program.    3

Invalid Option


Select from the following menu
     1.   Enter Stack Values.
     2.   Enter Queue Values.
     9.   Terminate the program.    X

Invalid Option


Select from the following menu
     1.   Enter Stack Value.
     2.   Enter Queue Values.
     9.   Terminate the program.    9



*** End of the program.  ***
*** Written by Husain Gholoom ***
*** March 28 – 2018  ***
```

Rules :

1. Your program **must compile** and run. The program will be tested using the **latest** version of Codeblocks for windows.
2. **You re not allowed to use stack and queue libraries . You Must implement all the functions of stacks and queues. You are also not allowed to use static arrays , vectors or string manipulation operations , or tempalte .**
3. Your program must be properly **documented according the style above . See the website for the sample programming style program.**
4. **You must use repetitions , control structures , and switch statements.**
5. Must properly format the output by use the appropriate library. See the output below . Also , **Replace my first / last name with your own first / last name.**
6. You must name your program as :

   - **3358_0_LastName_FirstName_PG4.cpp (Section 260)**
   - **3358_1_LastName_FirstName_PG4.cpp (Section 261)**
   - **3358_2_LastName_FirstName_PG4.cpp (Section 262)**

   Where LastName is your Last Name and FirstName is your First Name. For example , the file name should look something like : 3358_1_Gholoom_Husain_PG4.cpp (not .cbp)

7. Every one must upload the electronic version of the program no later than the starting of class time on the due date. **No late assignments will be accepted.** **DO NOT** send your assignment solution via email. **Group members must upload identical copy of the assignment.**

   Use TRACS To upload electronic version of your program

8. 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. **Only one copy per group.**

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

The following points will be deducted if :

- Compilation errors , incorrect file format such as uploading .cbp instead of .cpp , missing electronic copy , missing the hardcopy , different copies of the assignment per group , using .h and .cpp files , using vectors , using stack / queue libraries **,** string manipulations, using static arrays or global arrays , using vectors or global vectors , not using class … etc **(- 10 points)**

- Logical Errors **(at least - 1.25 points)**

- Other **(at least -1.50 points)** if **any** of the following takes a place :

  - Unable to read the source code due to unclear printing or print layout is not portrait.
  - Incorrect Output format.
  - Incorrect program file name.
  - Hard copy is not stapled.
  - Incorrect Style **such as but not limited to** Missing Header comments , signature line , comments or program documentations , missing roster number , section number , , not using switch , using template … etc