Algorithms and data structures

Labwork 3 - Stacks and Queues

Please use Google classroom for labwork submission; join the class by the following link: https://classroom.google.com/c/NTQ1NjMwOTMwNzU0?cjc=gqtgfva

There are 6 Labworks in this course. After each lab work session:

- You need to complete one of the given exercises and upload your files to the assignment "Labwork No. - Group No. - Version 1 (Attendance)". Submission must be done within 30 mins after the lab; otherwise, it will be considered a late submission.
- You will have one week (or 7 days) to complete the remaining exercises and upload your files to the assignment "Labwork No. - Group No. - Version 2 (Complete)"
- Compress all code source files in a zip file and rename it as FULLNAME-ID-Lab#no.zip (e.g NguyenVanA-070-Lab1.zip). Save your files according to the exercise number i.e Ex1.cpp, Ex2.c, etc. Incorrect filenames will result in no score for the respective exercises.
- Only code source files (.c or .cpp) should be in the zip files. Other files (.exe, .o) MUST be removed from the zip file.
- Copy/Paste from any source is not tolerated. The penalty will be applied for a late submission.

NOTE: You must follow the guide. Incorrect zip file names, zip files containing other files (.exe), and copy/pasting lead to heavy penalties.

Exercise 1: 12 points
Suppose we would like to implement a queue system for a commercial website. Given that the website offers three items for sale (or a list of items), each item has a limited quantity in stock. Implement a queue of customers, each customer can buy only a few products among the available three items.

- Determine the item name, their quantity in stock;
- Specify a queue of n customers; each customer can buy k products from one item (k is different for each customer);
- Customers take turns to enter (enqueue) and leave (dequeue) the queue according to the FIFO order to purchase wanted products;

- If a customer successfully purchases products, then display a message and reduce the number of products in stock; otherwise, if the item has been run out, display a warning message.

Implement the above problems in C/C++ using a Queue data structure. Write the main function for testing all written functions (init(), display(), enqueue(), dequeue(),...). .

Exercise 2: 8 points
In this problem, we try to implement the undo/redo mechanism using two stacks. Choose one method to implement a Stack data structure (Array-based Stack or Stack using Linked List) to implement the above problem.

- String of characters are stored in the Stack data structure;
- If a character is popped from Undo stack, then it is immediately pushed into Redo stack; and vice-versa;

Write the main function for testing all written functions (init(), display(), push(), pop(),...).