CS 2401 Spring 2022

Lab 9: Stacks and Queues

Due: Friday 3/18 at 11:59 PM

Part 1:

For this lab you are to use the STL stack which you can obtain by putting `#include<stack>` at the top of your program and having the std namespace open.

Write a program that evaluates an arithmetic expression in postfix (RPN) notation. The basic algorithm is to use a single stack of integers. If, when parsing the input you encounter a number, you push the number onto the stack. If you encounter an operator (+, -, *, / are the only ones we are working with), you apply the operator to the top two elements on the stack. The result is then pushed back onto the stack. At the end of the whole operation there should be a single number on the stack which is the value of the expression (the answer).

Some important hints:

- In the files `calc_useful.h` and `calc_useful.cc` I have provided a couple of functions that make this whole lab a lot easier.
- I have also provided enough of the main to show you how to do the input for the expression. It is called `stackmain.cc`
- This lab does not involve writing a class – only completing the necessary stack operations in the main

The calculator should be embedded in a loop that runs until the user chooses to quit.

Use your calculator to evaluate:

```
12 679 5 + 4 * 8 / +  (Answer = 354)   Answer:((679+5)*4)/8+12=354

34 6 + 16 * 18 – (Answer = 622)   Answer:((34+6)*16)-18=622

12 6 – 8 + 7 * + (Should produce an error)

56 78 84 + 33 *  (Should produce an error)
```

Plus: Two problems that you make up on your own. These should each have a minimum of four operands.

Run a script of the above interaction including the two problems that you made up on your own.

Part 2:

For this assignment you are to use the STL queue and priority queue classes. You can do this with `#include <queue>`.

Create a **class** for a single chore. This class will have two private attributes, a string describing what the chore is and a priority number (higher numbers meaning the item is more urgent). Overload the < operator so that it returns whatever the comparison of the priority levels is. Your class should have input and output functions as well, allowing the chore and its priority level to be input from the keyboard and output to the screen.

In the main, declare an STL queue of type chore. (*Note*: You are declaring a queue of chores, **not** pointers to chores). You will also need a temp chore to handle input. Now start a little loop and type in five chores, along with their priority level. This will involve calling `temp.input();` (OR `cin>>temp;` depending on how you've done your chore input), and then `q.push(temp);` Note that the queue only allows you to push to one end. Once you have finished your input loop, have a loop that outputs all the chores in the queue.

```
cout << q.front();  OR q.front().output();  //depending on how you've
done your chore output.
q.pop(); // removes that item from the queue.
```

Run this program while doing a script file (or take a screenshot of the output)

Now change your main so that you are declaring a `priority_queue` of type chore. <u>You will also have to change "front" to "top" in the output</u>. Now compile this version.

Run this program while doing a script file (or take a screenshot of the output). Input the same chores that you did for the non-priority queue version of the program.

When you compare the two script files you should see a different ordering of the chores, with the second being ordered by priority.

Submit a copy of your final source code files along with all script files or screenshots according to the instructions of your TA.