CSci 1113: Introduction to C/C++
Programming for Scientists and Engineers
Homework 5
Spring 2020

Due Date: Friday, March 27, 2020 before 11:55pm.

Instructions: This is an individual homework assignment. There are two problem worth 20 points each. Solve the problem below by yourself (unlike the labs, where you work collaboratively), and submit the solution as a C++ source code file. Here are a few more important details:

1. Unlike the computer lab exercises, this **is not** a collaborative assignment. You must design, implement, and test the solution to each problem on your own without the assistance of anyone other than the course instructor or TAs. In addition, you may not include solutions or portions of solutions obtained from any source other than those provided in class: examples from the textbook, lectures, or code you and your partner write to solve lab problems. Otherwise obtaining or providing solutions to any homework problems for this class is considered academic misconduct. See the "collaboration rules" file on the class website page for more details, and ask the instructor if you have questions.

2. Because all homework assignments are submitted and tested electronically, the following are important:
   - You follow the naming conventions mentioned at the end of the problems.
   - You submit the correct file(s) on gradescope (https://www.gradescope.com/) by the due deadline.
   - You follow the example input and output formats exactly given in each problem description.
   - **Regardless of how or where you develop your solutions, your programs compile and execute on gradescope computers (which run Linux/Ubuntu operating system like the cselabs machines).**

3. The problem descriptions will usually show at least one test case and the resulting correct output. However, you should test your program on other test cases (that you make up) as well. Making up good test cases is a valuable programming skill, and is part of ensuring your code solution is correct.

Problem A: Saving (20 points)
Both parts of this homework deal the the tictactoe.cpp posted on the class webpage:
http://www-users.cselabs.umn.edu/classes/Spring-2020/csci1113/assignments/tictactoe.cpp

For part A of the homework, you need to modify the existing code to enable two features (before the game ends):
(1) You can enter 'q' to quite the game.
(2) You can enter 's' to save the game.

If you quit, it should simply say "Goodbye." then end the program.

If you save the game, you should create a file "save.txt" in the current folder that holds the information about the current board. This file should be formatted:

X: [list of numbers where X is, separated by spaces]
O: [list of numbers where O is, separated by spaces]

So for example, the board:

```
X|O|O
-----
4|X|6
-----
7|8|9
```

Should correspond to a “save.txt” file:

```
X: 1 5
O: 2 3
```

(Note: the numbers do **not** need to be in any particular order in the file.)

Example 1 (user input is underlined):

```
1|2|3
-----
4|5|6
-----
7|8|9
What number do you with to play at? (or (s)ave or (q)uit)
s

(many newlines)

1|2|3
-----
4|5|6
-----
7|8|9
What number do you with to play at? (or (s)ave or (q)uit)
q
Goodbye.
```

Resulting “save.txt”

```
X:
O:
```

Example 2 (user input is underlined):

```
1|2|3
-----
```

```
4|5|6
-----
7|8|9

What number do you with to play at? (or (s)ave or (q)uit)
1
```

(many newlines)

```
X|O|3
-----
4|5|6
-----
7|8|9

What number do you with to play at? (or (s)ave or (q)uit)
5
```

(many newlines)

```
X|O|O
-----
4|X|6
-----
7|8|9

What number do you with to play at? (or (s)ave or (q)uit)
4
```

(many newlines)

```
X|O|O
-----
X|X|O
-----
7|8|9

What number do you with to play at? (or (s)ave or (q)uit)
s
```

(many newlines)

```
X|O|O
-----
X|X|O
-----
7|8|9
```

```
What number do you with to play at? (or (s)ave or (q)uit)
9
```

(many newlines)

```
X|O|O
-----
X|X|O
-----
7|8|X
Game over!  You win!
```

Resulting "save.txt"

```
X: 1 4 5
O: 2 3 6
```

When you are done, name the source code file hw5A.cpp. Then log into gradescope and upload your file for the "Homework 5A" submission. **If you name your file incorrectly it will be unable to compile and run your code, so you will fail all test cases**. You may submit cpp files as many times as you want until the deadline to try and fix the code if you fail a test case. Following rigorous naming conventions and using test cases are something computer programmers often must do in "real life" programming, and so submitting your program with the correct name and functionality is part of doing this assignment correctly.

Problem B: Loading (20 points)
Again, use the tictactoe.cpp file as a starting point. Also again, you need to implement two features (only one "new" one):

(1) You can enter 'q' to quite the game.
(2) You can enter 'l' to load the game from "save.txt". If the "save.txt" file exists the indicated spots in the file should be filled with the appropriate marker, while all others are blank (and assume it is your turn to move). If there is no save file, do not change the current game.

Note: You should be able to load **<u>any</u>** file, even if it is nonsensical if it is in the proper format. For example, you should be able to load:

```
X:
O: 1 2 5 4
```

... and end up in a board:

```
O|O|3
-----
O|O|6
-----
7|8|9
```

You do not need to handle any case where the board is already won, however. You can also assume the

same number will not appear on the first and second line.

Example 1 (user input underlined, first input is a lower case ‘L’):

```
1|2|3
-----
4|5|6
-----
7|8|9
What number do you with to play at? (or (l)oad or (q)uit)
l
```

(many newlines)

```
X|O|X
-----
O|5|O
-----
X|O|X
What number do you with to play at? (or (l)oad or (q)uit)
5
```

(many newlines)

```
X|O|X
-----
O|X|O
-----
X|O|X
Game over!  You win!
```

The “save.txt” file before running

```
X: 1 3 7 9
O: 2 4 6 8
```

When you are done, name the source code file hw5B.cpp. Then log into gradescope and upload your file for the “Homework 5B” submission. **If you name your file incorrectly it will be unable to compile and run your code, so you will fail all test cases**. You may submit cpp files as many times as you want until the deadline to try and fix the code if you fail a test case. Following rigorous naming conventions and using test cases are something computer programmers often must do in “real life” programming, and so submitting your program with the correct name and functionality is part of doing this assignment correctly.