CSci 1113: Introduction to C/C++
Programming for Scientists and Engineers
Homework 4
Spring 2018

Due Date: Thursday, March 8, 2018 before 11:55pm.

Instructions: This is an individual homework assignment. There are two problems 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.

2. Because all homework assignments are submitted and tested electronically, the following are important:
   - You follow any naming conventions mentioned in the homework instructions.
   - You submit the correct file(s) through Moodle 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 cselabs computers running the Linux operating system.**

3. 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.

General comments:
Both parts of this homework relate to the Connect-4 code posted on the website. The general purpose of Connect-4 is to be the first person to get 4-in-a-row. This game has historically been played where you slid chips into columns, so the play is always the lowest available row in the column. See this for more details if you are unfamiliar with the rules:

https://en.wikipedia.org/wiki/Connect_Four

For both part A and B of this homework ***you should not add any couts in your final submission***. You may add temporary couts to help debug, but please ensure these are all removed when you do the final submission. Also for both parts, I do not care what "message" you tell the user for the modifications (do whatever you want for this). Also please do not put '\n' characters in the "message" variable.

Problem A: Saving a game (20 points)
Add functionality so the user can type 's' to save the game. When this "save" option is selected, you should put the board exactly as it is shown into the file "save.txt". If "save.txt" already exists, override it with the current board.

Note: like in the example, you should still be able to play the game normally after saving.

Example 1 (user input is underlined, but many newlines are skipped):
. . . . . .
. . . . . .
. . . . . .

```
.......
.......
.......
1234567

Which column do you wish to play in? Or (s)ave/(l)oad?
1

.......
.......
.......
.......
.......
X......
1234567

Which column do you wish to play in? Or (s)ave/(l)oad?
2

.......
.......
.......
.......
.......
XO.....
1234567

Which column do you wish to play in? Or (s)ave/(l)oad?
1

.......
.......
.......
.......
X......
XO.....
1234567

Which column do you wish to play in? Or (s)ave/(l)oad?
2

.......
.......
.......
.......
XO.....
XO.....
```

```
1234567

Which column do you wish to play in? Or (s)ave/(l)oad?
1

.......
.......
.......
X......
XO.....
XO.....
1234567

Which column do you wish to play in? Or (s)ave/(l)oad?
2

.......
.......
.......
XO.....
XO.....
XO.....
1234567

Which column do you wish to play in? Or (s)ave/(l)oad?
s

.......
.......
.......
XO.....
XO.....
XO.....
1234567
Game saved.
Which column do you wish to play in? Or (s)ave/(l)oad?
1

.......
.......
X......
XO.....
XO.....
XO.....
1234567

The mighty Xs reign supreme!
```

Inside “save.txt” after above example:

```
.......
.......
.......
XO.....
XO.....
XO.....
```

When you are done, name the source code file <username>_4A.cpp. Here you replace <username> with your U of M email address; for example, if your email address is smithx1234@umn.edu, your file should be named smithx1234_4A.cpp. Then submit your program using the HW 4 Problem A submission link in Moodle.

Problem B: Loading games (20 points)
Add functionality to either the base connectFour.cpp or your answer to part A to press ‘l’ to load a game from “save.txt”. If this file does not exist, you should not change the current board. Otherwise, you should change the board and let the user play the saved board normally.

Note: You may assume the the “save.txt” is in a valid board for the game.

Example 1 (user input is underlined, note that the game would continue but I do not show more):

```
.......
.......
.......
.......
.......
.......
1234567

Which column do you wish to play in? Or (s)ave/(l)oad?
6

.......
.......
.......
.......
.......
.....X.
1234567

Which column do you wish to play in? Or (s)ave/(l)oad?
1

.......
.......
.......
```

```
.......
.......
.....X.
1234567
No save file
Which column do you wish to play in? Or (s)ave/(l)oad?
```

... and program would continue

Example 2 (user input is underlined, this is with a “save.txt” as shown in Example 1 of part A):

```
.......
.......
.......
.......
.......
.......
1234567
Which column do you wish to play in? Or (s)ave/(l)oad?
6

.......
.......
.......
.......
.......
.....X.
1234567
Which column do you wish to play in? Or (s)ave/(l)oad?
1

.......
.......
.......
XO.....
XO.....
XO.....
1234567
Game loaded
Which column do you wish to play in? Or (s)ave/(l)oad?
1

.......
.......
X......
XO.....
XO.....
XO.....
```

```
1234567

The mighty Xs reign supreme!
```

When you are done, name the source code file <username>_4B.cpp. Here you replace <username> with your U of M email address; for example, if your email address is smithx1234@umn.edu, your file should be named smithx1234_4B.cpp. Then submit your program using the HW 4 Problem B submission link in Moodle.