Assignment #5  
Practice Using Arrays  
Due: Sunday, 11/22/15, 11:59pm

Grading: EVERY assignment in this course is graded by demoing your work for 10 minutes with a TA. You are required to meet with a TA within one week from the due date to demo. You are penalized for failure to see a TA within the week or missing a scheduled appointment. In either case, if you are within 1 day (24 hours) of the deadline, you lose 10 points. If you are within 7 days (1 week) of the deadline, then you lose 25 points, anything outside of a week from the deadline to demo is an automatic 50 point deduction. Your job is to convince the TA that your program works correctly, i.e. show your TA how to use/break your program.

(80 pts) Problem Statement: Write a C++ program that plays the game of connect four. To see how to play the traditional 2-person game, visit page 2 of these instructions: http://www.hasbro.com/common/documents/dad2614d1c4311ddbd0b0800200c9a66/1EF6874419B9F36910222EB9858E8CB8.pdf

Connect Four has a 6 x 7 board, and the goal of the game is to connect four pieces together horizontally, diagonally, or vertically. You need to make sure you handle inappropriate data, i.e. non-positive integers, and ask the user again for the correct input. You must continue to play the game, until the user no longer wants to play.

2-player Example:
./connect_four

Player one, do you want red or yellow (r or y)? r

| | | | | | | |
|-------------------------|
| | | | | | | |
|-------------------------|
| | | | | | | |
|-------------------------|
| | | | | | | |
|-------------------------|
| | | | | | | |
|-------------------------|

Player 1, what column do you want to put your piece? 4
Player 2, what column do you want to put your piece? 5

Player 1, what column do you want to put your piece? 4

Player 2, what column do you want to put your piece?...

Do you want to play again (0-no, 1-yes)? 0
A requirement for this program is that your functions are 10 lines or less, and you must not have any global variables.

**Command-line argument**

In addition to the 2-player game, you will need to support a 1-player game! If the user runs the program supplying a `-a` option, then you will run the game as 1-player connect four, where the other player is the computer. You can make this as easy or as hard as you want. The computer can just guess a random column or be very smart, this is up to you.

**1-player Example:**

```
./connect_four -a

Player one, do you want red or yellow (r or y)? r

| | | | | | | |
------------------------
| | | | | | | |
------------------------
| | | | | | | |
------------------------
| | | | | | | |
------------------------
| | | | | | | |
------------------------
| | | | | | | |
------------------------
| | | | | | | |
------------------------

Player 1, what column do you want to put your piece? 4

| | | | | | | |
------------------------
| | | | | | | |
------------------------
| | | | | | | |
------------------------
| | | | | | | |
------------------------
| | | | | | | |
------------------------
```
Player 1, what column do you want to put your piece? 4

Player 1, what column do you want to put your piece? 4

Do you want to play again (0-no, 1-yes)? 0
Your program must be able to:

- Print an error message and recover, when the player doesn’t supply a valid column on the board. This includes selecting a column that has filled all rows.
- Print an error message when the user enters an invalid option as a command line argument. You do not need to recover from this.
- Play the game correctly based on how many pieces to connect or when any user fails to win by the board completely filling.
- Continue to play until the user selects no.

(10 pts Extra Credit) Connect Any
Support a board that can be any size, as well as any number of pieces to connect! You need to ask the user for the dimensions of the board to create a dynamic board, and ask the user for the number of pieces to connect. For example, you could have a 4x4 board with 2 pieces to connect or a 10x10 with 5 pieces to connect. The game should still play correctly, and you should check that their board will support the number of pieces they want to connect😊

(10 pts) Program Style/Comments
In your implementation, make sure that you include a program header in your program, in addition to proper indentation/spacing and other comments! Below is an example header to include. Make sure you review the style guidelines for this class, and begin trying to follow them, i.e. don’t align everything on the left or put everything on one line!
http://classes.engr.oregonstate.edu/eecs/fall2015/cs161-001/161_style_guideline.pdf

/***************************************************************************/
** Program: connect_four.cpp
** Author: Your Name
** Date: 11/05/2015
** Description:
** Input:
** Output:
***************************************************************************/

(10 pts) Testing
Report any checking/self-reflection you did while solving the problem. For instance, how did you make sense of the output from the implementation? This includes things such as using a calculator to make sure the output is correct, testing to make sure your code executes correctly and behaves the way you expect under specific circumstances, using external sources of information such as the internet to make sense of the results, etc. In addition, you will provide us a test plan!

You will be graded on how thorough your test plan is. Make sure you think about input you hope works and input that won’t work. Your program doesn’t have to handle input that doesn’t work!!!
Please see the example of a test plan: Polyta_template.docx

Electronically submit your C++ program (.cpp file, not your executable!!!) and test plan, as a pdf, by the assignment due date, using TEACH.

**NOTE: ** The easiest way to upload your program from ENGR to TEACH is to map a network drive to your home directory on ENGR. Mac or Windows, See: http://engineering.oregonstate.edu/computing/fileaccess/