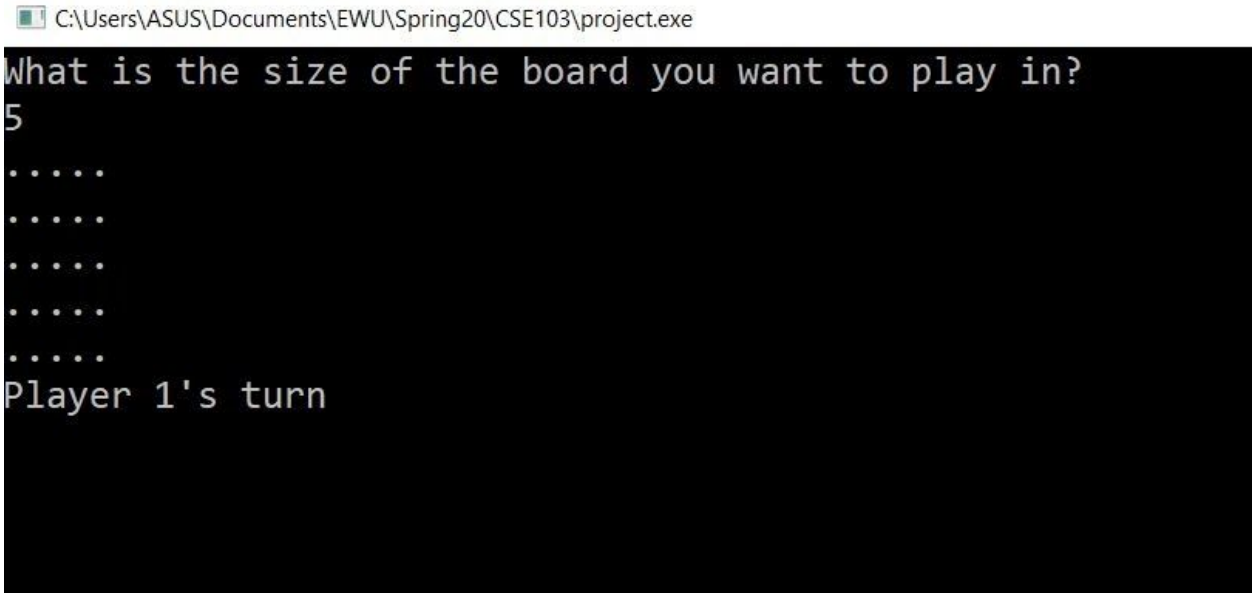
Tic Tac Toe

Submission Deadline: 24th September, 2020, 11:59 pm
How to submit: Rename your c file to your_student_id.c (e.g. if your student id is 2020-1-60-001, your file should be renamed to 2020-1-60-001.c) and submit your code in the google classroom assignment.

Project Description:
I guess all of you may know the game of tic tac toe (if not, you can play the game using this link) by now. The game is typically played in a 3*3 board, which is initially empty. Two players alternate their turns to fill up the board. The first player fills any one of the empty cells with an 'X' in the next turn, the second player fills any one of the empty cells with an 'O'. The players alternate their turns till somebody wins, or all the cells are filled up before anyone can win (in which case it is a draw). If all the cells of any one of the rows or the columns or either of the two diagonals is filled up with 'X'-s, player 1 wins. If they are filled up with 'O'-s. player 2 wins. [Seriously, play the game, it is much easier to understand that way].

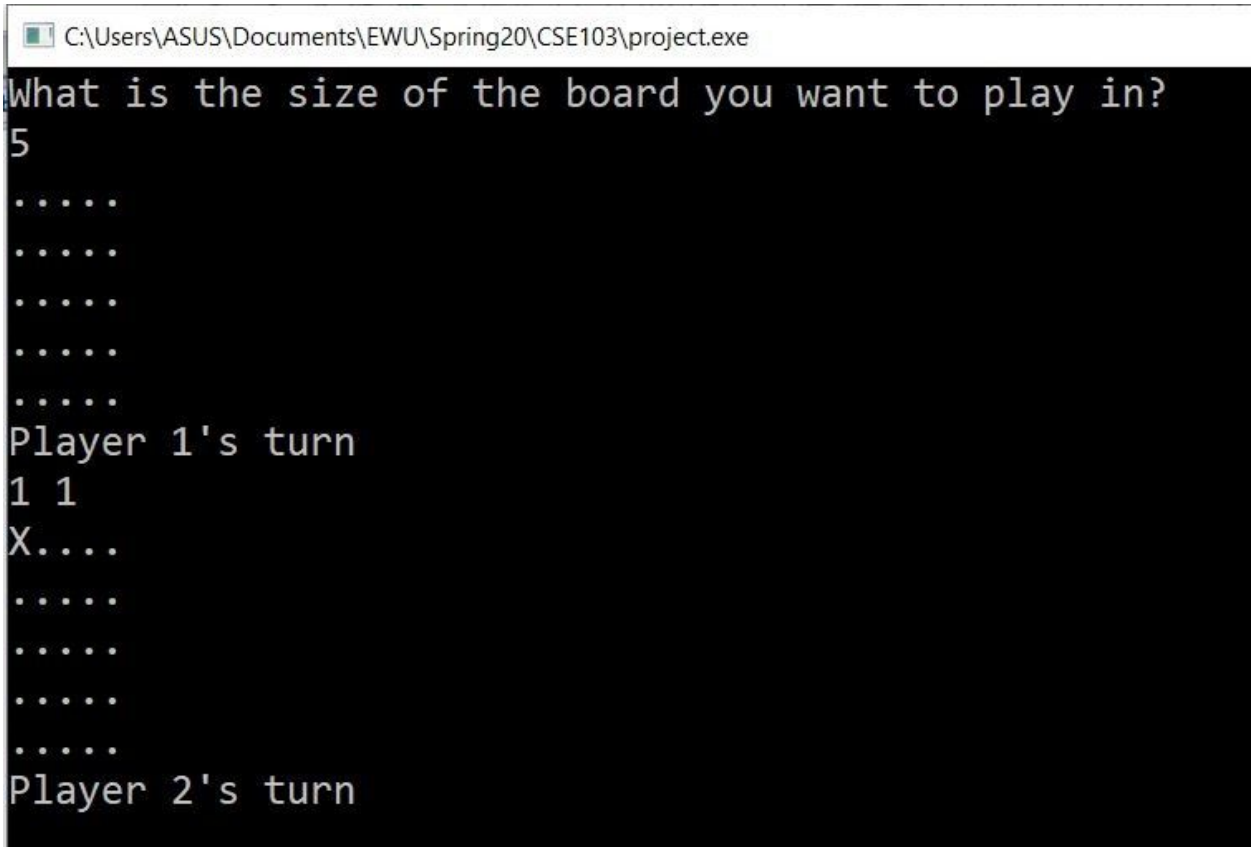
In this project, you will be tasked with writing a program that allows us to play the game. However, instead of always playing on the traditional 3*3 board, the game will be played on an n*n board.

At the beginning, your program will ask the user about his preferred board size, (i.e. the value of n). In the example shown below, the user has entered 5 as his preferred board size. You can assume that the user will never prefer a board with size greater than 15.



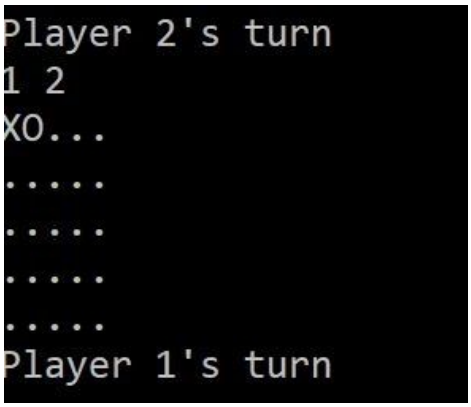
Accordingly, the program prints a 5*5 board, which is initially empty (use '.'-s to specify the empty cells). The program will then mention whose turn it is, starting with player 1, and then to player 2, and the turns will alternate from there on.

When it's player 1's turn, the first player will enter two numbers, specifying the row and column number of the cell in which he wants to make his move. Upon receiving the input, your program will then draw the board again, this time with an 'X' in the position specified by the first player. In the following example, the player inputs "1 1", and the board is redrawn accordingly with an X in the first column of the first row.

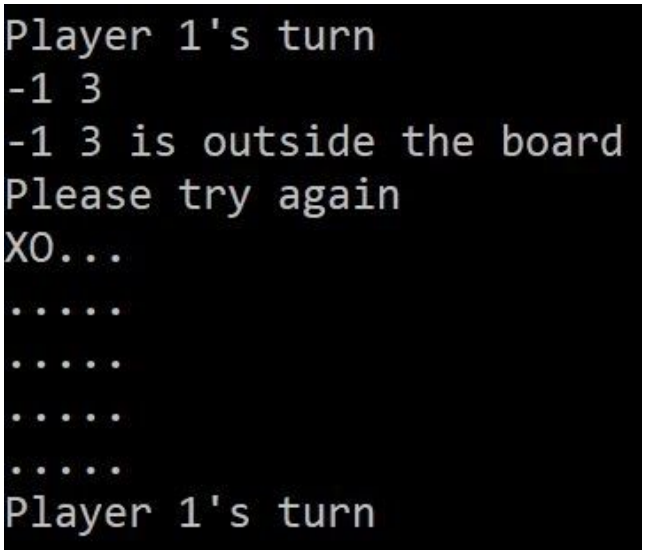


In a perfect world, the players' moves will always be valid. But the world is not perfect and you can never place any assumptions on what the user's moves might be. This can be shown in the following example, where player 2 enters a position which is already filled up. Your program should be able to prevent the user from being able to do this. It should display an appropriate error message, and prompt the same player to try again.

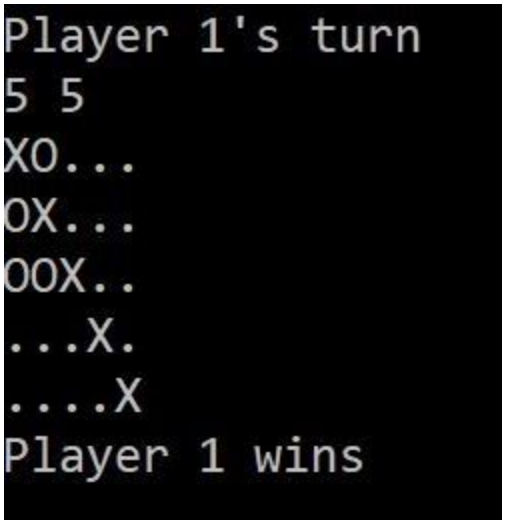
Once player 2 inputs a valid move, the turn will then go back to player 1. This alternating of turns will continue until the game is won by either player or drawn.



Again, user inputs can be anything, and player 1 may then even enter a row or column that does not exist at all, Your program should be able to handle this as well, prompting player 1 to try again and printing an appropriate error message. An example is shown in the following:



In this way, players 1 and 2 will keep playing the game, and the game will end if and only if the game is won by either player, or drawn. When the game finally ends, your program should output who won or in case of a draw, simply print “It’s a draw” and then return. An example is shown below:



Here, player 1 won by managing to fill up the entire diagonal with ‘X’-s. I guess we can all agree player 2 played very poorly here.

Good luck! Give your best effort, and remember that while honest efforts are always appreciated, any form of cheating, if detected, will get you a 0 in your project marks.