PROLOGUE

FOR ALL ASSIGNMENTS

- **Attach a *prologue* for all assignments.**
- **Use sample *prologue* sheet in the course material, customize it for every assignment.**
- ***Prologue* makes it easy to separate assignments for grading purpose.**

EXERCISE 6

AFTER CHAPTER – 10

PROBLEM

Implement the game tic-tac-toe as described below. Show the input and output of the game and finally the result. You may use a 4 X 4 array and you are free to use the algorithm described in the course material. Show the initial board, the moves of each player, and the result of the game including the final board.

Tic-Tac-Toe Game Strategy:

The first player can win if the second player does not use a perfect defense strategy. The second player can rarely win unless playing against an inexperienced opponent or unless the opponent makes a mistake.

In general, players at their turn must 1) first check to see if they can win during this turn. That is, if they can complete a row, column or diagonal with three of their symbols 2) Next check to see if the opponent has two symbols in a row, column, or diagonal that would allow the opponent to win at their next turn.

Many consider the center cell the best first move choice for either player. The other player can be forced to defend early game plays made by the holder of the center cell, usually resulting in a tie.

There are many variations of the game but **Tic-Tac-Toe** is played on an *N*-by-*N* grid. Two players play, one can place *X*'s on the board, the other O's.

EXERCISE 6

The player who obtains *N X*'s (or *N O*'s) in a row – vertical, horizontal, or diagonal – wins.

It should first ask player "***one***" to make a move, asking for the row and column of where the "***X***" is to be placed. Then it should do the same for player "***two***", placing an "***O***" in the appropriate row and column.

It should alternately ask the players, checking after each move if a player has won (or if the game is over because there's no more room let to place a piece).

Print the initial board and result of the game in a grid.

Initial setting of the grid:

Rows	Col 0	Col 1	Col 2	Col 3
Row0	--	--	--	--
Row1	--	--	--	--
Row2	--	--	--	--
Row3	--	--	--	--

Possible Final setting of the grid:

Rows	Col 0	Col 1	Col 2	Col 3
Row0	X	O	X	O
Row1	X	O	X	X
Row2	O	X	X	O
Row3	X	O	O	O

EXERCISE 6

The player who obtains *N X*'s (or *N O*'s) in a row – vertical, horizontal, or diagonal – wins.

It should first ask player "***one***" to make a move, asking for the row and column of where the "***X***" is to be placed. Then it should do the same for player "***two***", placing an "***O***" in the appropriate row and column.

It should alternately ask the players, checking after each move if a player has won (or if the game is over because there's no more room let to place a piece).

DELIVERABLES

Write the prolog and fill up all information for this exercise as given in the sample. Submit the source code, input and the output. The program is expected to be well commented. Place your program as soft copy on assigned shared drive for students of this course.

DUE DATES

Assignments are due on the following week after completing the chapter discussion.