CSC 15 project#4
Due Date: December 9th by midnight

You must follow all the rules to get the full credit such as comments, naming conventions, indentation,

Problem: You are to create a 2 player tic-tac-toe game.
It will be two human players playing.(i.e. It is not a human playing against the computer.)

Your program will consist of a class called ***TicTacToe*** that will encapsulating the concept of a tic-tac-toe game, and a client application class that will control the play of the game using the methods from the *TicTacToe* class..

See the sample output for the interactive play of the game on the last page of this document.

Following is the UML diagram for the TicTacToe class:

TicTacToe
+ PLAYER_X: char = ‘X’ + PLAYER_O: char = ‘O’ - xName: String - oName: String - board: char[] - moveCount: int

+ TicTacToe(playerXName: String, playerOName: String)
+ initializeBoard( )
+ takeATurn(keyboard: Scanner, player: char): boolean
+ isWinner(char player): boolean
+ displayBoard( )
+ displayResults( )
- validMove(int move): boolean

- The **constructor** will initialize the player names to the names passed to the parameters. It will instantiate the board array, and then it will call initializeBoard to fill the board array and initialize moveCount.

- ***initializeBoard*** will initialize the moveCount to 0 and fill the board array by putting the character ‘1’ into board[1], the character ‘2’ into board[2], etc…
  This method will be called to reset the board and moveCount every time a new game is started.

.
- ***takeATurn*** will execute one complete turn for the ***player*** passed to the method. The prerequisites for this method are:
    - there is not yet a winner
    - there is still a place on the board to move
    - ***player*** is either ***PLAYER_X*** or ***PLAYER_O***

  This method will determine what name to user for the player in the prompts to the user.
  It will ask the user where he/she wishes to play and then if the user’s response is a valid move it will place the player’s ‘X’ or ‘O’ in that space on the board. If it is not a valid move (use the ***validMove*** method), it will let the user know and again ask the user where he/she wishes to play. The method will not end until the user has selected a valid move and the move has been recorded in the board array. When the move is complete, the move count will be incremented.

- ***isWinner*** will return true if ***player*** has won the game, false otherwise.
  Prerequisite for this method is that *player* is either ***PLAYER_X*** or ***PLAYER_O***.
  To determine if the ***player*** is a winner the method will use a long boolean expression to look at all the possible combinations of ways to win. i.e. are all the characters in the first row equal to *player,* **or** are all the characters in the second row equal to *player,* **or** are all the characters in the third row equal to *player* etc….

- ***dislayBoard*** displays to the console, the current state of the ***board*** array in the format shown in the output on the last page of this document.

- ***displayResults*** will use the *isWinner* method to determine who the winner of the game is.
  The prerequisite for this method is that the game is over. i.e. either there is a winner or else all the spaces on the board have been played and there is no winner, i.e.”the cat won”.
  The method will display an appropriate method on the console that will either congratulate the winner, or else declare “The game is a tie. – The CAT won.”.

- ***validMove*** will return true if move is in the proper range 1..9 and the board at that position is not an‘O’ or ‘X’. Otherwise it will return false.

Class: ***PlayTicTacToe***

The Tic Tac Toe client Application class is the driver program that contains the main method.

The main method will get the names of the players from the user and use those names to instantiate an instance of the **TicTacToe** class called ***game***.

The main method will have one big loop that will execute as long as the client says "yes" he/she whishes to play another game of tic tac toe.

Inside the loop (every time the game is played):

- The game will be initialized

- (While the last player to take a turn is not a winner and the players have not taken 9 turns)
  - The player is switched (i.e. if the player is PLAYER_X, the player will become PLAYER_O, etc…)
  - The player ***takes A Turn***.
  - number of turns taken is incremented.

- The results will be displayed

- The players will be asked if they wish to play the game again.

The console display from playing the game one time should look something like the following. The bold entries are the user's inputs.

```
What is the Name of the player who will play X's? Mom
What is the Name of the player who will play O's? Dad
 1 | 2 | 3
 ---------
 4 | 5 | 6
 ---------
 7 | 8 | 9
Mom Where would you like to play? 5
 1 | 2 | 3
 ---------
 4 | X | 6
 ---------
 7 | 8 | 9
Dad Where would you like to play? 4
 1 | 2 | 3
 ---------
 O | X | 6
 ---------
 7 | 8 | 9
Mom Where would you like to play? 0
0 is not a valid move.
Mom Where would you like to play? 2
 1 | X | 3
 ---------
 O | X | 6
 ---------
 7 | 8 | 9
Dad Where would you like to play? 5
5 is not a valid move.
Dad Where would you like to play? 8
 1 | X | 3
 ---------
 O | X | 6
 ---------
 7 | O | 9
Mom Where would you like to play? 3
 1 | X | X
 ---------
 O | X | 6
 ---------
 7 | O | 9
Dad Where would you like to play? 1
 O | X | X
 ---------
 O | X | 6
 ---------
 7 | O | 9
Mom Where would you like to play? 7
 O | X | X
 ---------
 O | X | 6
 ---------
 X | O | 9

CONGRATULATIONS Mom YOU WON!

Do you want to play again (Y/N)? n
Thank-you for playing Tic Tac Toe.
 HAVE A NICE DAY!
```