CPE 101 (Spring 2018)
Project 3
Tic-Tac-Toe Simulator

Your code must pass all of the tests to get any credit. Your grade will be based on the date which your code passes all the tests. To get 100% credit for this project, you must submit in PolyLearn by 11:00PM on 5/9/18. Code that does not pass all of our tests will not get any credit.

Monday (5/9) - 100% @ 11:00PM
Tuesday (5/10) - 80% @ 11:00PM
Wednesday (5/11) - 60% @ 11:00PM

Objectives:

- Practice on List
- More practice on writing loops in python.
- More practice on writing and calling functions that you have written.
- Practice testing your code.
- To appreciate the benefits of modular development.

Required File Header:

All students are required to have the following header comment at the top of their source files. Note that the stuff to the right of the colon in red is information for an imaginary example student. (Please put YOUR appropriate information. Also, your header comment is not expected to have red text.) All program files require this header.

```
# Project 3 - Tic-Tac-Toe Simulator
# Name: Your Name
# Instructor: S. Einakian
# Section: Your Section
```

Resources:

- Your instructor, via office hours, email, etc.
- Free tutoring Sunday-Thursday, 7-9 PM, 14-302
- Your TA

Program Description:

For this program, you will be writing a simulation for the game, Tic-Tac-Toe. Two people play Tic Tac Toe with paper and pencil. One player is X and the other player is O. Players take turns placing their X or O. If a player gets three of their marks on the board in a row, column or one of the two diagonals, they win. When the board fills up with neither player winning, the game ends in a draw.

This program has two parts:

1. Two Players
2. Play with Computer

Part One: Two Players

- Take user input (for two players)
- Store user inputs into a 3x3 Tic-Tac-Toe board
- Check for validity of inputs (no overriding, proper cell value, etc.)
- Check for a win every turn
- Print the board every turn

Part Two: With Computer

- Randomly pick a player (you or computer)
- Take input (for one players)
- Computer or you will generate the next
- Store user inputs into a 3x3 Tic-Tac-Toe board
- Check for validity of inputs (no overriding, proper cell value, etc.)
- Check for a win every turn
- Print the board every turn

Getting Player Input:

- Each player will input a number (1-9) which corresponds to the following cells.

1	2	3
4	5	6
7	8	9

- Location 0 in list corresponds with number 1
- Players' input will be stored in a list.
- Players will take turns.

Checking For a Win:

- A player wins if there is a sequence of 3 of the same letter in the same row, column, or diagonal.

Printing Output:

- You should use for loops to output a 3x3 board.

General Notes:

- You should implement your functions in a file, **tictactoeFuncs.py**
- Checking for a win requires multiple operations.

Functions:

1. Welcome
    a. Welcomes user to the game and explain the game rules.
    b. You will provide your own Welcome information
    c. Ask user to decide if playing with computer (pass 1) or another player (pass 2)
    d. No parameter to pass
    e. Return 1 if play with computer and 2 if plays with another player

2. createBoard
    a. You will use a list to store the data
    b. List has 9 items starting from index 0 to 9 which holds the values (X or O)
    c. This function has no parameter to pass and it return a list
    d. Your game board at first should look like the following:

```
1  |  2  |  3
---------------
4  |  5  |  6
---------------
7  |  8  |  9
```

3. printBoard
    a. The printBoard function will print the game. Remember that the board is represented as a list of nine strings, where the string at index 0 is the mark on space 1 on the Tic Tac Toe board, and so on.
    b. Passing a list as the board to the function.
    c. The first call to the print will show an empty board game:

```
   |   |
---------------
   |   |
---------------
   |   |
```

4. pickLetter
    a. This function ask user to pick ‘X’ or ‘O’
    b. No parameter to pass
    c. This will return the letter of player’s choice
    d. If user enters the wrong letter you need to ask user to reenter the letter
    e. This function return letter

5. getInput
    a. This function takes the letter and board as parameters

b. Ask user to enter the letter of his/her choice in the one of the locations (1-9) on the board
    i. Where do you like to place your letter (pick in range of 1-9)
    ii. If user repeat the location or pass value outside range (1-9) for placing the letter:
        1. Send a message:
            a. Invalid move! Location is already taken. Please try again.
c. This function will return the board after entering the letter

6. checkRows
    a. takes the board as parameter
    b. returns True and the letter if a row is filled with the same Letter

7. checkCols
    a. takes the board as parameter
    b. returns True and the letter if a column is filled with the same Letter

8. checkDiags
    a. takes the board as parameter
    b. returns True and the letter if any of diagonals are filled with the same Letter

9. boardFull
    a. takes the board as parameter
    b. returns True if the board is full

10. checkWin
    a. takes the board
    b. if any of checking (rows, columns, diagonals) returns True
        i. announce the winner
    c. if board is filled and no winner announce draw!

11. turnCount

    Use this function to show, which turns the game is.

    Example:

    ```
    It's Player 2's (O' )turn!
    ```

    a. takes count of turns
    b. increment counter
    c. return counter

Resources:

You can download the following files from PolyLearn:

- testFunctions.py, at least 2 test cases for each function
- tictactoeFuncs.py
- incol (test file for two player)
- inrow (test file for two player)
- india (test file for two player)

For testing your program with these files:
python3 tictactoe.py < incol
python3 tictactoe.py < inrow
python3 tictactoe.py < india

Note:

You can add any extra functions you need. You have freedom on how you format your messages. Make sure to follow the steps. The sample screenshot is given.
USE EXACTLY THE SAME FUNCTIONS' NAME.

Submission:

- testFunctions.py (test all your functions- at least two tests for each function)
- tictactoe.py
- tictactoeFuncs.py

Rubric:

1) tictactoeFuncs.py (implement all functions) 5 Points
2) tictactoe.py (call all necessary functions) 5 Points
3) testFunctions.py (two test for each function) 10 Points
4) pass 7 tests 70 Points
5) comments 10 Points