CSCI 1411: Fundamentals of Computing
Lab 12
Due Date: November 17, 2023

Name: Brandon Perez

Goals:

- Using nested lists (2D lists).
- Using nested loops to iterate over a nested list.
- Using f string to format the output.
- Functions

Development Environment: IDLE

Deliverables:

1. This completed document with required screen shots.
2. Python file created for the lab. Name the file using the following format: `lastnameLab12.py`.

How to take a **screen shot**:

- For a Windows 10: Use Snipping Tool to copy and press CTRL + V to paste screen shot.
- For Mac: Press Shift + Command (⌘) + 4 to copy and press Command (⌘) + V to paste screen shot.

Tic Tac Toe

> Tic-tac-toe (American English), noughts and crosses (Commonwealth English), or Xs and Os (Canadian or Irish English) is a paper-and-pencil game for two players who take turns marking the spaces in a three-by-three grid with X or O. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row is the winner. It is a solved game, with a forced draw assuming best play from both players. (https://en.wikipedia.org/wiki/Tic-tac-toe).

In this lab you will write a that will let two players play the game of Tic Tac Toe. You will write functions to perform various sub tasks (checking rows, checking column, checking diagonal, checking for empty cells, and printing grid). Grid will be represented with a nested (2D) list. Outer list will contain three inner list (each representing a row of the grid). Each inner grid will contain three elements (one for each column). Elements of the inner list can be an 'x', 'o' or blank (' '). Make sure that you read this lab handout carefully before starting to write your code. You will implement the following functions:

1. reset_grid (grid): reset the grid by changing all the elements to blank (' ').
2. check_row (grid, row_num): returns True if all the elements in the given row are x or all the elements in the given row are o, otherwise it will return False.
3. check_column (grid, col_num): returns True if all the elements in the given column are x or all the elements in the given column are o, otherwise it will return False.

4. check_diagonal (grid, diagonal): returns True if all the elements in the given diagonal are x or all the elements in the diagonal are o, otherwise it will return False. Note that the diagonal d1 includes elements [0][0], [1][1] and [2][2] and the diagonal d2 includes elements [0][2], [1][1] and [2][0].
5. place_entry (grid, row_num, col_num, ch): places the ch into the location given by row_num and col_num). ch will be either x or o. It will check to make sure that the cell given by row_num and col_num is blank. If it is blank then it will place the ch in the given location and return True. If the cell is not empty then it will return False.
6. playable (grid): check to see if there are any empty (blank) elements. It will use nested loops to check each element in the nested list. If any of the element is empty (blank) then it will return True, otherwise it will return False.
7. print_grid (grid): prints grid using f string.

main function will use the above set of functions to perform various task. Outline of the main function is as follows:

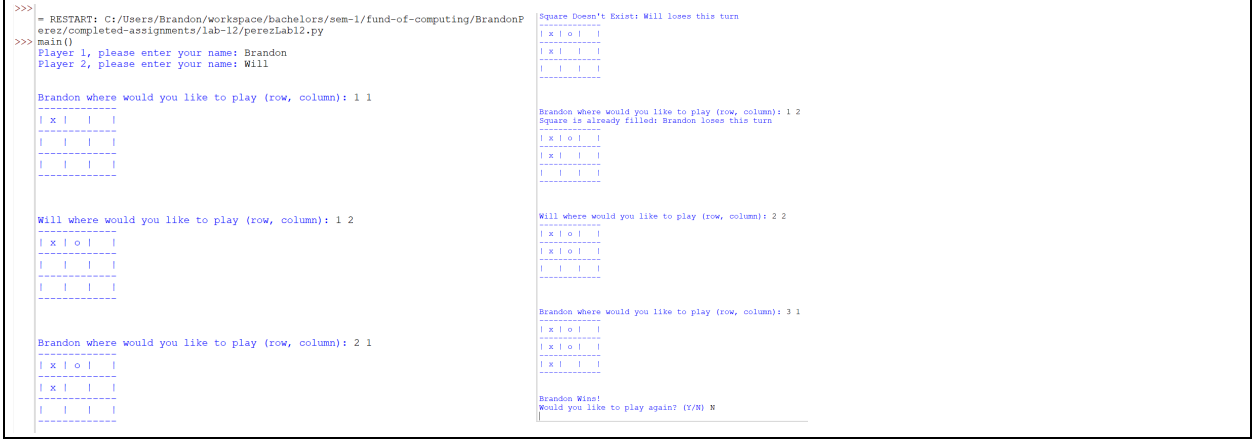
1. Create a nested lists as follows:
   ```
   grid = [[' ', ' ', ' '], [' ', ' ', ' '], [' ', ' ', ' ']]
   ```
2. Ask player 1 to enter their name. Player 1 will get x and will go first.
3. Ask player 2 to enter their name. Player 2 will get o and will go second.
4. Loop which iterates till game is over. Game will end when one of the players wins or there are no more empty cells in the grid.
   a. Ask player one to enter the location which they want to mark the location. The player will enter two integers as x y; where x is the row number and y is the column number. Example: 2, 3. Note that players will count rows and columns as 1, 2, and 3 but our program will use 0, 1, and 2 for row number and column number.
      i. Check to make sure that the entered row number and column numbers are 1, 2, or 3. If they are not then print an error message and player 1 loses their turn.
      ii. Use the place_entery function to place an x in the selected location. If place_entry returns a False then print an error message and player 1 loses their turn.
      iii. Print the grid.
      iv. Use check_row function to check the selected row for completion. If the check_row function returns True then print the message that Player 1 won the game and this will end the game.
      v. Use check_colum function to check the selected column for completion. If the check_row function returns True then print the message that Player 1 won the game and this will end the game.
      vi. If the player placed the mark on the diagonal (d1 or d2) the use the check_diagonal function to check for completion. If the check_diagonal function returns True then print the message that Player 1 won the game and this will end the game.
      vii. Use playable function to check if there are any empty cells. If playable function returns False then there are no empty cells. Print the message that game is draw and end the game.

b. Ask player two to enter the location which they want to mark the location. The player will enter two integers as x y; where x is the row number and y is the column number. Example: 2, 3. Note that players will count rows and columns as 1, 2, and 3 but our program will use 0, 1, and 2 for row number and column number.
   i. Check to make sure that the entered row number and column numbers are 1, 2, or 3. If they are not then print an error message and player 2 loses their turn.
   ii. Use the place_entery function to place an o in the selected location. If place_entry returns a False then print an error message and player 2 loses their turn.
   iii. Print the grid.
   iv. Use check_row function to check the selected row for completion. If the check_row function returns True then print the message that Player 2 won the game and this will end the game.
   v. Use check_colum function to check the selected column for completion. If the check_row function returns True then print the message that Player 2 won the game and this will end the game.
   vi. If the player placed the mark on the diagonal (d1 or d2) the use the check_diagonal function to check for completion. If the check_diagonal function returns True then print the message that Player 2 won the game and this will end the game.
   vii. Use playable function to check if there are any empty cells. If playable function returns False then there are no empty cells. Print the message that game is draw and end the game.
5. If the game is not over then go back to the top of the loop (Step 4).
6. If the game is over then ask the players if thee want to play again. If they answer yes then use the reset_grid function to reset the loop and go back to step 2.

Note: You can implement step a and b using a single function if you want to simplify your main function.

Once you are satisfied that your program is working correctly then play the game with someone, take a screen shot of your results and paste it in the box below:

Screen Shot 1



Sample I/O

```
Player 1 enter your name: David
Player 2 enter your name: Mary
David Enter location to be marked (row, col): 1 1
-------------
| x |   |   |
-------------
|   |   |   |
-------------
|   |   |   |
-------------
Mary Enter location to be marked (row, col): 2 2
-------------
| x |   |   |
-------------
|   | o |   |
-------------
|   |   |   |
-------------
```

```
David Enter location to be marked (row, col): 4 3
Error: Your row or column number is out of bound
Mary Enter location to be marked (row, col): 1 2
-------------
| x | o |   |
-------------
|   | o |   |
-------------
|   |   |   |
-------------
David Enter location to be marked (row, col): 1 2
That location is already taken
Mary Enter location to be marked (row, col):
```

```
-------------
| x | o |   |
-------------
|   | o |   |
-------------
|   | o |   |
-------------
Mary You won the game
Do you want to play again (y or n)?
```

Add the following comment block at the top of your file. Also make sure that you have block of comments after start each of the functions (see handout for lab 11).

```
"""
Name:
Class: CSCI 1411-00X
Due Date:
Description:
Status:
"""
```

Rubric for Lab 12:

Criteria	Rating
Screen shot 1	Screen shot included – 5 points No screen shot included – 0 points
reset_grid function	reset_grid sets all elements to blank – 5 points No implementation of reset_grid function or it does not work as expected – 0 points
check_row function	Function works as expected – 5 points No implementation of function or does not work as expected – 0 points
check_column function	Function works as expected – 5 points No implementation of function or does not work as expected – 0 points
check_diagonal function	Function works as expected – 5 points No implementation of function or does not work as expected – 0 points
place_entry function	Function works as expected – 5 points No implementation of function or does not work as expected – 0 points
playable function	Function works as expected – 5 points No implementation of function or does not work as expected – 0 points
print_grid function	Prints using the f function (see sample I/O) – 10 points Prints but is not formatted correctly – 5 points No implementation of function or does not work as expected – 0 points
main function	Uses the functions to perform various tasks – 10 points Does not use functions to perform various tasks – 0 points
main function	main function implements all the required steps – 10 points main function does not implement all or some of the required steps – 0 points
main function (bonus points)	main function uses a single function to implements steps in a and b. They will get 5 bonus points (not included in total)
Total	65