**Due: week 10, beginning of class**

**Instruction:**

* Demonstrate your work to your instructor during the class
* Submit your source codes to Moodle
* Your source codes must contain your StudentId and Name

Suppose a tic-tac-toe game is represented by a 3x3 array, where *0* means a empty slot, *1* and *2* represent the pieces 'O' and 'X' respectively.

For example, here is a possible games with 4 pieces on board representing the game on the right

|  |  |  |
| --- | --- | --- |
|  | *O* | *O* |
| *X* | *X* |  |
|  |  |  |

x = np.array [[0, 1, 1],

[2, 2, 0],

[0, 0, 0]]

Write a function win(state) in python to check whether one of the side has won the game, where state is a 3x3 array representing a board as defined above

The function should return the side (1 or 2) that has won, or 0 if no one is winning yet.

**Part of the file (TicTacToeWhoWinStu.py) is provided. Please use it to do your work.**

**Bonus:** Complete the provided program TicTacToeMiniMax.py by adding the function you have written above. Experiment by varing the variable MAX\_PLY from 1 to 6. Note the difference in computer response time and the quality of play by the AI,

**Example Test Cases**

x = np.array ([[0, 1, 1],[2, 2, 0],[0, 0, 0]])

print (win(x))

*0*

x = np.array ([[0, 1, 1],

[2, 2, 2],

[0, 0, 1]])

print (win(x))

*2*

x = np.array ([[2, 1, 1],

[2, 2, 1],

[1, 2, 1]])

print (win(x))

*1*

x = np.array ([[2, 1, 1],

[0 ,2, 1],

[0, 0, 2]])

print (win(x))

*2*

x = np.array ([[2, 1, 1],

[1 ,1, 2],

[1, 2, 2]])

print (win(x))

*1*

x = np.array ([[2, 1, 1],

[1 ,2, 2],

[1, 2, 1]])

print (win(x))