Name: Chandan A.M.
1BM18CS025

AI Lab.

2) Implement Tic-Tac-Toe with AI vs AI:

```
def create_board():
    return ( array([[0,0,0],[0,0,0],[0,0,0]]) )

def random_place(board, player):
    selection = possible(board)
    curr_loc = random.choice(selection)
    board[curr_loc] = player
    return board

def row_win(board, player):
    for x in range(len(board)):
        win = True
        for y in range(len(board)):
            if board[x,y] != player
            win = False
            continue
    if win == True
        return win
    else
        return win
```

Chand

Chandan A.M.
1BM18CS025

AI Lab

```
def col_win (board, player)
    for x in range (len(board)):
        win = True
    for y in range (len(board)):
        if board[y][x] != player:
            win = False
            continue
        if win == True:
            return win

    return win

def evaluate (board):
    winner = 0
    for player in [1, 2]:
        if (row_win (board, player) or
            col_win (board, player) or
            diag_win (board, player)):
                winner = player

    if np.all (board != 0) and winner == 0:
        winner = -1

    return winner.
```

Chandan A.M.
CS025

AI Lab.

```
def diag_win ( board, player ):
    win = True
    y = 0
    for x in range( len(board) ):
        if board [x, x] != player :
            win = False
    if (win):
        return win
    win = True
    if win :
        for x in range (len(board)):
            y = len(board) - (-x
            if board [x, y] != player:
                win = False

    return win.

def play_game():
    board, winner, counter = create_board(), 0, 1
    print (board)
    sleep(2)
    while winner == 0 :
        for player in [1, 2]:
```

```
board = random_place(board, player)
print("Board after " + str(counter) + " move")
print(board)
sleep(2)
counter += 1
winner = evaluate(board)
if winner != 0:
    break
return(winner)
```