

Lab program 1

1. Implement Tic-Tac-Toe game

```
board = [' ' * 9 for x in range(10)]
```

```
def insertLetter(letter, pos):  
    board[pos] = letter
```

```
def spaceIsFree(pos):  
    return board[pos] == ' '
```

```
def printBoard(board):  
    print(' | | |')  
    print(' ' + board[1] + ' | ' + board[2] + ' | ' + board[3])  
    print(' | | |')  
    print('---')  
    print(' | | |')  
    print(' ' + board[4] + ' | ' + board[5] + ' | ' + board[6])  
    print(' | | |')  
    print('---')  
    print(' | | |')  
    print(' ' + board[7] + ' | ' + board[8] + ' | ' + board[9])  
    print(' | | |')
```

```
def isWinner(bo, le):  
    return (bo[7] == le and bo[8] == le and bo[9] == le)  
    or (bo[4] == le and bo[5] == le and bo[6] == le)  
    or (bo[1] == le and bo[2] == le and
```

```

b1 = 10 and b2 = 10 and
b3 = 10 and b4 = 10 and

```

```

b1 = 10 and b2 = 10 and b3 = 10 and b4 = 10 and
b5 = 10 and b6 = 10 and b7 = 10 and b8 = 10 and

```

```

b1 = 10 and b2 = 10 and b3 = 10 and b4 = 10 and
b5 = 10 and b6 = 10 and b7 = 10 and b8 = 10 and

```

```

b1 = 10 and b2 = 10 and b3 = 10 and b4 = 10 and
b5 = 10 and b6 = 10 and b7 = 10 and b8 = 10 and

```

```

b1 = 10 and b2 = 10 and b3 = 10 and b4 = 10 and
b5 = 10 and b6 = 10 and b7 = 10 and b8 = 10 and

```

```

def player_move():

```

```

    sum = 0

```

```

    while sum < 10:

```

```

        move = input("Please select a position
        to place an 'X' (1-9): ")

```

```

        try:

```

```

            move = int(move)

```

```

            if move > 0 and move < 10:

```

```

                if space_is_free(move):

```

```

                    sum = False

```

```

                    insert_at(move, 'X')

```

```

            else:

```

```

                print("Sorry this space is
                occupied.")

```

```

            else:

```

```

                print("Please type a number
                within the range.")

```

```

        except:

```

```

            print("Please type a
            number.")

```

```
def compMove():  
    possibleMoves = [x for x, letter in  
        enumerate(board) if letter == ' ' and x != 0]  
    move = 0
```

```
    for let in ['O', 'X']:  
        for i in possibleMoves:  
            boardCopy = board[:]  
            boardCopy[i] = let  
            if isWinner(boardCopy, let):  
                move = i  
    return move
```

```
cornersOpen = []  
for i in possibleMoves:  
    if i in [1, 3, 7, 9]:  
        cornersOpen.append(i)
```

```
if len(cornersOpen) > 0:  
    move = selectRandom(cornersOpen)  
    return move
```

```
if 5 in possibleMoves:  
    move = 5  
    return move
```

```
edgesOpen = []  
for i in possibleMoves:  
    if i in [2, 4, 6, 8]:  
        edgesOpen.append(i)
```

```
if len(edgesOpen) > 0:
    move = SelectRandom(edgesOpen)
```

```
return move
```

```
def selectRandom(li):
    import random
    ln = len(li)
    r = random.randrange(0, ln)
    return li[r]
```

```
def isBoardFull(board):
    if board.count(' ') > 1:
        return False
    else:
        return True
```

Algorithm

1: Initialize the board

```
board = [' ' for _ in range(10)]
```

2: Insert a letter ('X' or 'O') into the Board

```
def insertLetter(letter, pos):
```

3: Check if space on the Board is free:

```
def spaceIsFree(pos):
    return board[pos] == ' '
```

4: print the Board


```
def printBoard(board):
```

5: check for a winner.

```
def isWinner(b, l):
```

6: player's move

```
def playerMove():
```

Ask the player to choose a position
to place their 'X' on the board.

7: Computer's move.

Implement the logic for the computer's
move
def compMove

output

Player 'X', choose your position (1-9): 5

```

  |   |
  |   X   |
  |   |   |

```

Computer's turn

```

  |   |
  O  |  X  |
  |   |   |

```

Player X, choose your position (1-9): 1

```

X  |   |
  O  |  X  |
  |   |   |

```

Computer's turn

X	O	
---	---	--

O	X	
---	---	--

		1
--	--	---

Player 'X', choose your position (1-9) : 9

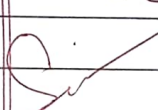
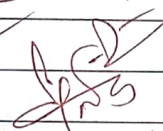
X	O	
---	---	--

O	X	
---	---	--

		X
--	--	---

Kaggle - don

Congratulations! You win!



 17-11-22