WEEK 2

TIC TAC TOE GAME

import random

# Initialize the game board

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

def insertLetter(letter, pos):

    global board

    board[pos] = letter

def spaceIsFree(pos):

    return board[pos] == ' '

def printBoard(board):

    print(' ' + board[1] + ' | ' + board[2] + ' | ' + board[3])

    print('-----------')

    print(' ' + board[4] + ' | ' + board[5] + ' | ' + board[6])

    print('-----------')

    print(' ' + board[7] + ' | ' + board[8] + ' | ' + board[9])

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 bo[3] == le) or

        (bo[1] == le and bo[4] == le and bo[7] == le) or

        (bo[2] == le and bo[5] == le and bo[8] == le) or

        (bo[3] == le and bo[6] == le and bo[9] == le) or

        (bo[1] == le and bo[5] == le and bo[9] == le) or

        (bo[3] == le and bo[5] == le and bo[7] == le)

    )

def playerMove():

    global board

    run = True

    while run:

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

        try:

            move = int(move)

            if 1 <= move <= 9:

                if spaceIsFree(move):

                    run = False

                    insertLetter('X', move)

                else:

                    print('Sorry, this space is occupied!')

            else:

                print('Please type a number within the range!')

        except ValueError:

            print('Please type a number!')

def compMove():

    global board

    possibleMoves = [x for x, letter in enumerate(board) if letter == ' ' and x != 0]

    for let in ['O', 'X']:

        for i in possibleMoves:

            boardCopy = board[:]

            boardCopy[i] = let

            if isWinner(boardCopy, let):

                return i

    cornersOpen = [i for i in possibleMoves if i in [1, 3, 7, 9]]

    if cornersOpen:

        return selectRandom(cornersOpen)

    if 5 in possibleMoves:

        return 5

    edgesOpen = [i for i in possibleMoves if i in [2, 4, 6, 8]]

    if edgesOpen:

        return selectRandom(edgesOpen)

    return None  # Indicates a tie

def selectRandom(li):

    ln = len(li)

    r = random.randrange(ln)

    return li[r]

def isBoardFull(board):

    return board.count(' ') <= 1

def main():

    global board

    print('Welcome to Tic Tac Toe!')

    printBoard(board)

    while not isBoardFull(board):

        if not isWinner(board, 'O'):

            playerMove()

            printBoard(board)

        else:

            print('Sorry, O\'s won this time!')

            break

        if not isWinner(board, 'X'):

            move = compMove()

            if move is None:

                print('Tie Game!')

            else:

                insertLetter('O', move)

                print('Computer placed an \'O\' in position', move, ':')

                printBoard(board)

        else:

            print('X\'s won this time! Good Job!')

            break

    if isBoardFull(board):

        print('Tie Game!')

    while True:

        answer = input('Do you want to play again? (Y/N)')

        if answer.lower() == 'y' or answer.lower() == 'yes':

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

            print('-----------------------------------')

            main()

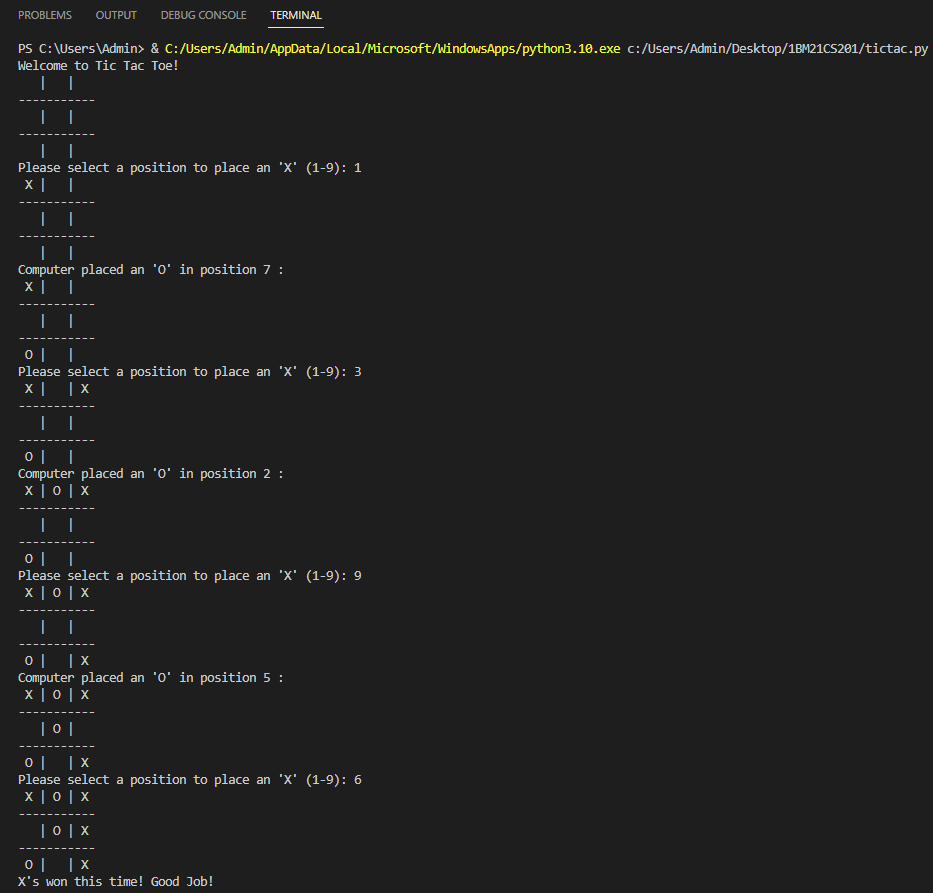
        else:

            break

# Run the game

main()

OUTPUT:

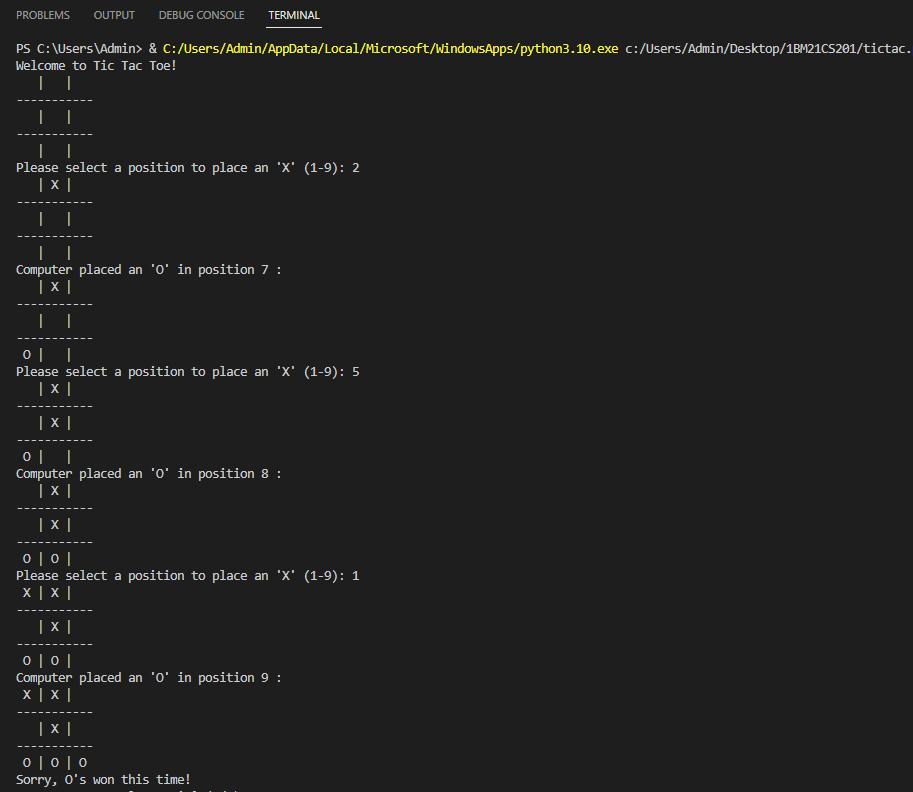


PROBABILITY OF ME WINNING IS ONLY 2 ATTEMPTS AS SHOWN ABOVE.

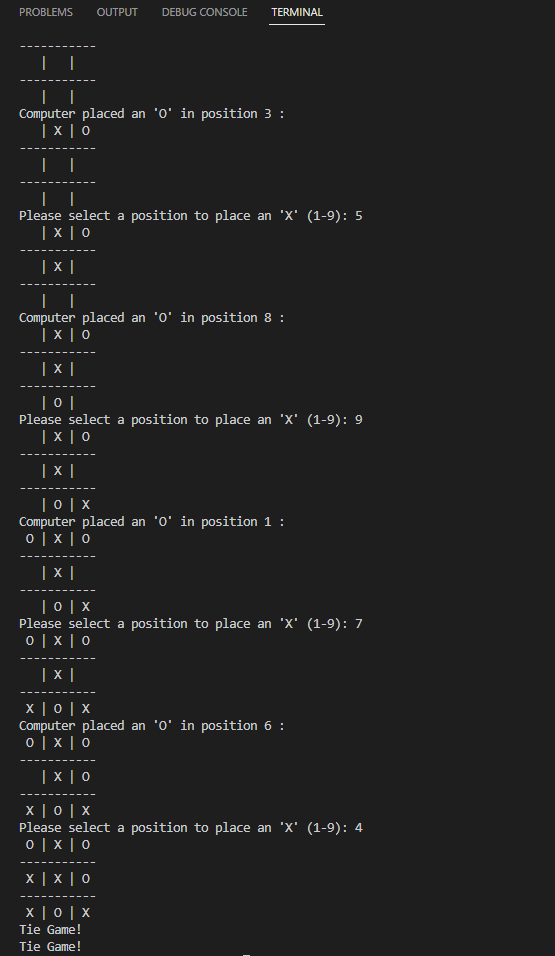
PLACE ALL OF THEM AT CORNERS INITIALLY LATER PLACE 1 OF THEM EITRHER AT 4TH 5TH OR 6TH POSITIONS.

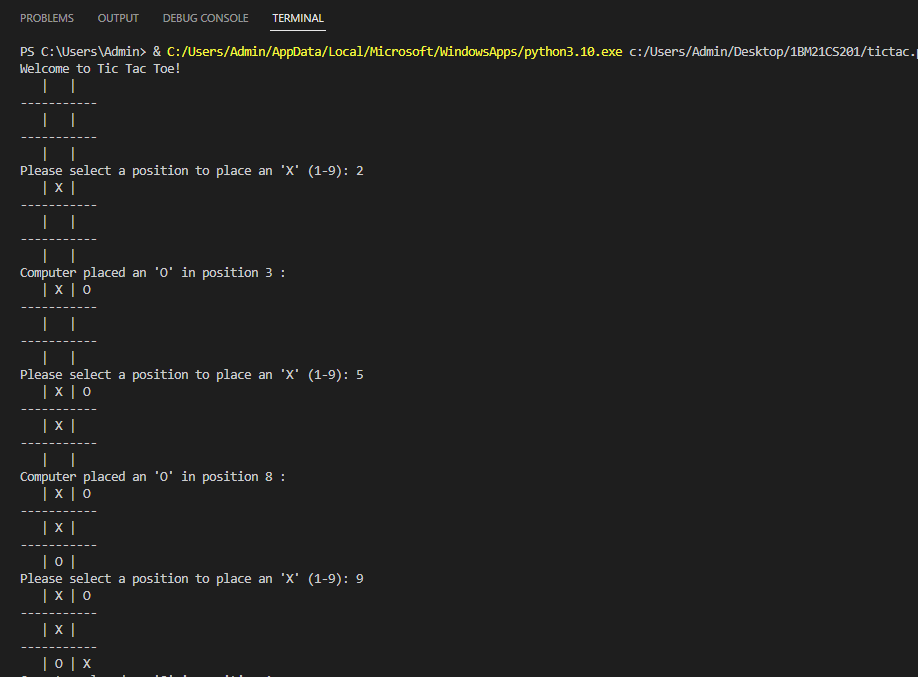
THAT’S HOW I CAN WIN.

REMAINING ONES ARE EITHER A TIE OR THE MACHINE WINNING.



COMPUTER WINNING





TIE CASE