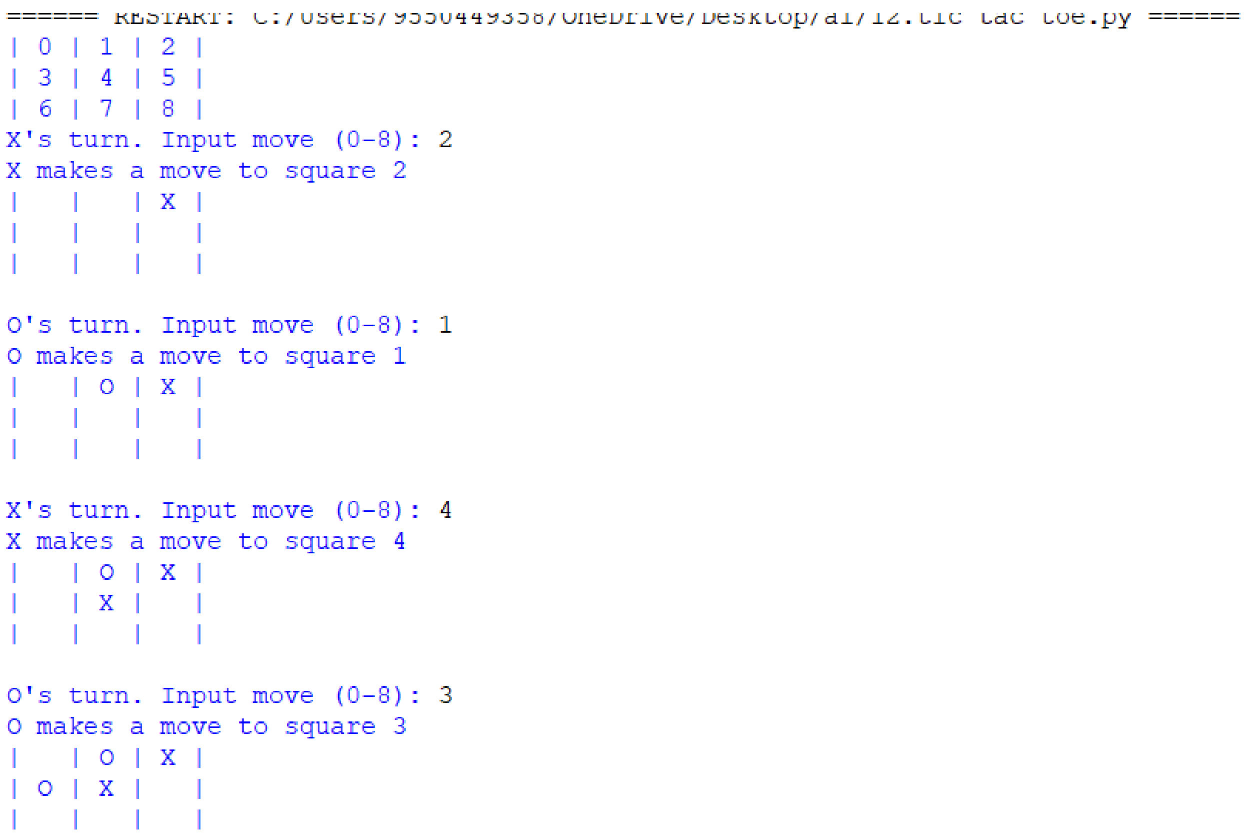
Date 02/03/2024

12.Aim: Write the python program for Tic Tac Toe game

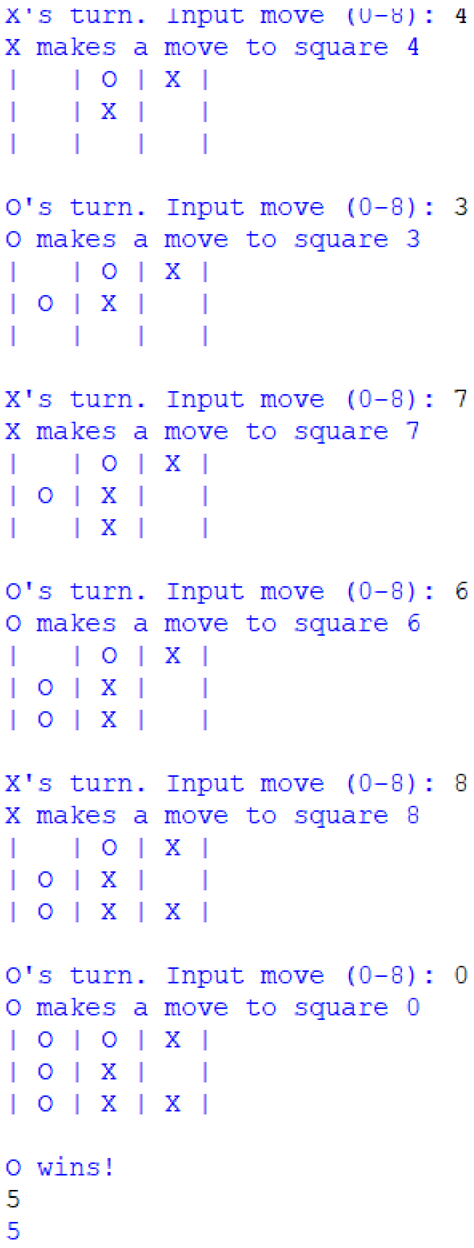
Program:

```
class TicTacToe:
    def __init__(self):
        self.board = [' ' for _ in range(9)]  # Initialize an empty board
        self.current_winner = None  # Keep track of the winner

    def print_board(self):
        for row in [self.board[i * 3:(i + 1) * 3] for i in range(3)]:
            print('| ' + ' | '.join(row) + ' |')

    @staticmethod
    def print_board_nums():
        number_board = [[str(i) for i in range(j * 3, (j + 1) * 3)] for j in range(3)]
        for row in number_board:
            print('| ' + ' | '.join(row) + ' |')

    def available_moves(self):
        return [i for i, spot in enumerate(self.board) if spot == ' ']

    def empty_squares(self):
        return ' ' in self.board

    def num_empty_squares(self):
        return self.board.count(' ')

    def make_move(self, square, letter):
        if self.board[square] == ' ':
            self.board[square] = letter
            if self.winner(square, letter):
                self.current_winner = letter
            return True
        return False

    def winner(self, square, letter):
        row_ind = square // 3
        row = self.board[row_ind*3:(row_ind+1)*3]
        if all([spot == letter for spot in row]):
            return True

        col_ind = square % 3
        col = [self.board[col_ind+i*3] for i in range(3)]
        if all([spot == letter for spot in col]):
            return True

        if square % 2 == 0:
            diagonal1 = [self.board[i] for i in [0, 4, 8]]
            if all([spot == letter for spot in diagonal1]):
                return True
            diagonal2 = [self.board[i] for i in [2, 4, 6]]
            if all([spot == letter for spot in diagonal2]):
                return True
        return False


def play(game, x_player, o_player, print_game=True):
    if print_game:
        game.print_board_nums()

    letter = 'X'
    while game.empty_squares():
        if letter == 'O':
            square = o_player.get_move(game)
        else:
            square = x_player.get_move(game)

        if game.make_move(square, letter):
            if print_game:
                print(letter + f' makes a move to square {square}')
                game.print_board()
                print('')  # Empty line

            if game.current_winner:
                if print_game:
                    print(letter + ' wins!')
                return letter  # Ends the loop and exits the game
            letter = 'O' if letter == 'X' else 'X'  # Switches player

    if print_game:
        print('It\'s a tie!')


class HumanPlayer:
    def __init__(self, letter):
        self.letter = letter
    def get_move(self, game):
        valid_square = False
        val = None
        while not valid_square:
            square = input(self.letter + '\'s turn. Input move (0-8): ')
            try:
                val = int(square)
                if val not in game.available_moves():
                    raise ValueError
                valid_square = True
            except ValueError:
                print('Invalid square. Try again.')
        return val


if __name__ == "__main__":
    x_player = HumanPlayer('X')
    o_player = HumanPlayer('O')
    t = TicTacToe()
    play(t, x_player, o_player, print_game=True)
```

Output:



```
====== RESTART: C:/Users/9550449358/OneDrive/Desktop/ai/12.tic tac toe.py ======
| 0 | 1 | 2 |
| 3 | 4 | 5 |
| 6 | 7 | 8 |
X's turn. Input move (0-8): 2
X makes a move to square 2
|   |   | X |
|   |   |   |
|   |   |   |

O's turn. Input move (0-8): 1
O makes a move to square 1
|   | O | X |
|   |   |   |
|   |   |   |

X's turn. Input move (0-8): 4
X makes a move to square 4
|   | O | X |
|   | X |   |
|   |   |   |

O's turn. Input move (0-8): 3
O makes a move to square 3
|   | O | X |
| O | X |   |
|   |   |   |
```



```
X's turn. Input move (0-8): 4
X makes a move to square 4
|   | O | X |
|   | X |   |
|   |   |   |

O's turn. Input move (0-8): 3
O makes a move to square 3
|   | O | X |
| O | X |   |
|   |   |   |

X's turn. Input move (0-8): 7
X makes a move to square 7
|   | O | X |
| O | X |   |
|   | X |   |

O's turn. Input move (0-8): 6
O makes a move to square 6
|   | O | X |
| O | X |   |
| O | X |   |

X's turn. Input move (0-8): 8
X makes a move to square 8
|   | O | X |
| O | X |   |
| O | X | X |

O's turn. Input move (0-8): 0
O makes a move to square 0
| O | O | X |
| O | X |   |
| O | X | X |

O wins!
5
5
```

Result: The given program has been executed successfully