AI LAB 1:

**Program Title: Tic Tac Toe game Code :**

import random

def check\_win(board, player):

# Check rows, columns, and diagonals for a win for row in board:

if all(spot == player for spot in row): return True

for col in range(3):

if all(board[row][col] == player for row in range(3)): return True

if all(board[i][i] == player for i in range(3)) or all(board[i][2 - i] == player for i in range(3)): return True

return False

def display\_board(board): for row in board:

print(row) print()

def get\_available\_moves(board):

return [(r, c) for r in range(3) for c in range(3) if board[r][c] == '-']

def bot\_move(board):

# Check if the bot can win in the next move for move in get\_available\_moves(board):

r, c = move board[r][c] = 'O'

if check\_win(board, 'O'):

print(f"Bot placed O at position: ({r + 1}, {c + 1})") display\_board(board)

return board[r][c] = '-'

# Check if the player is about to win, and block them for move in get\_available\_moves(board):

r, c = move board[r][c] = 'X'

if check\_win(board, 'X'): board[r][c] = 'O'

print(f"Bot placed O at position: ({r + 1}, {c + 1}) to block the player") display\_board(board)

return board[r][c] = '-'

# Otherwise, pick a random available move

move = random.choice(get\_available\_moves(board)) board[move[0]][move[1]] = 'O'

print(f"Bot placed O at position: ({move[0] + 1}, {move[1] + 1})") display\_board(board)

# Initial board setup

board = [['-', '-', '-'], ['-', '-', '-'], ['-', '-', '-']]

display\_board(board)

xo = 1 # 1 for human, 0 for bot

flag = 0 # Flag to check for win or draw

while '-' in board[0] or '-' in board[1] or '-' in board[2]: if xo == 1: # Human's turn (X)

print("Enter position to place X (row and column between 1-3):") try:

x = int(input("Row: "))

y = int(input("Column: ")) except ValueError:

print("Invalid input. Please enter numbers between 1 and 3.") continue

if x > 3 or y > 3 or x < 1 or y < 1: print("Invalid position") continue

if board[x - 1][y - 1] == '-':

board[x - 1][y - 1] = 'X' display\_board(board)

if check\_win(board, 'X'): print("X wins!")

flag = 1 break

xo = 0 # Switch to bot's turn else:

print("Invalid position") continue

else: # Bot's turn (O) print("Bot's turn:") bot\_move(board)

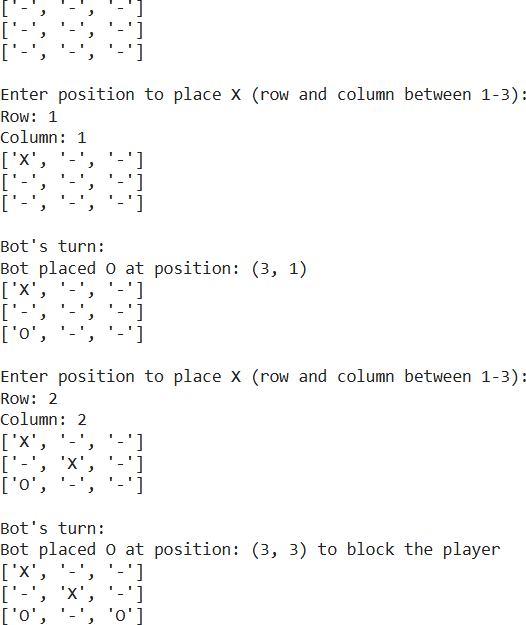
if check\_win(board, 'O'): print("O (Bot) wins!") flag = 1

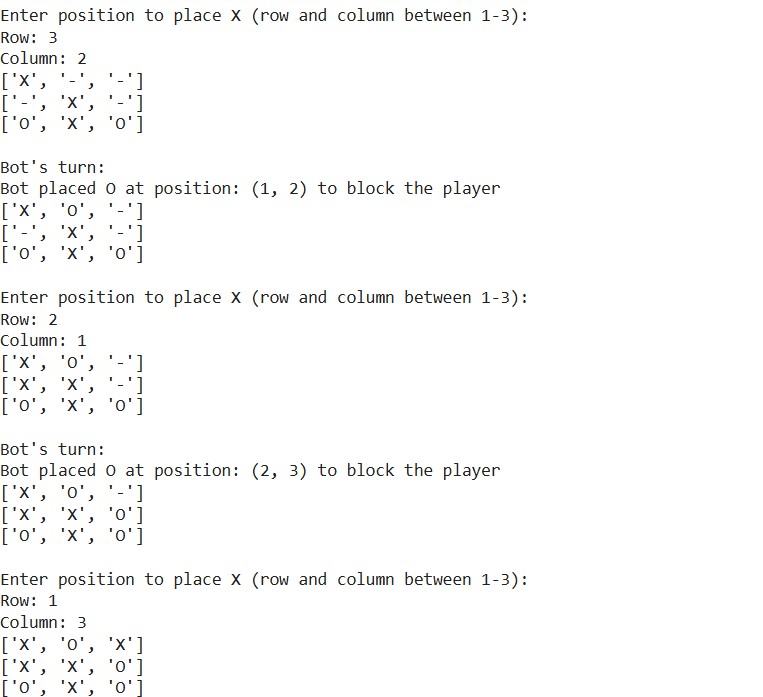
break

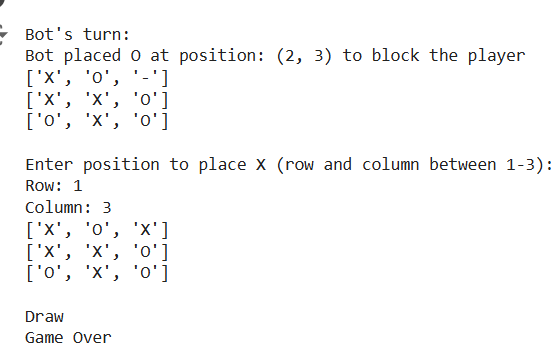
xo = 1 # Switch back to human's turn

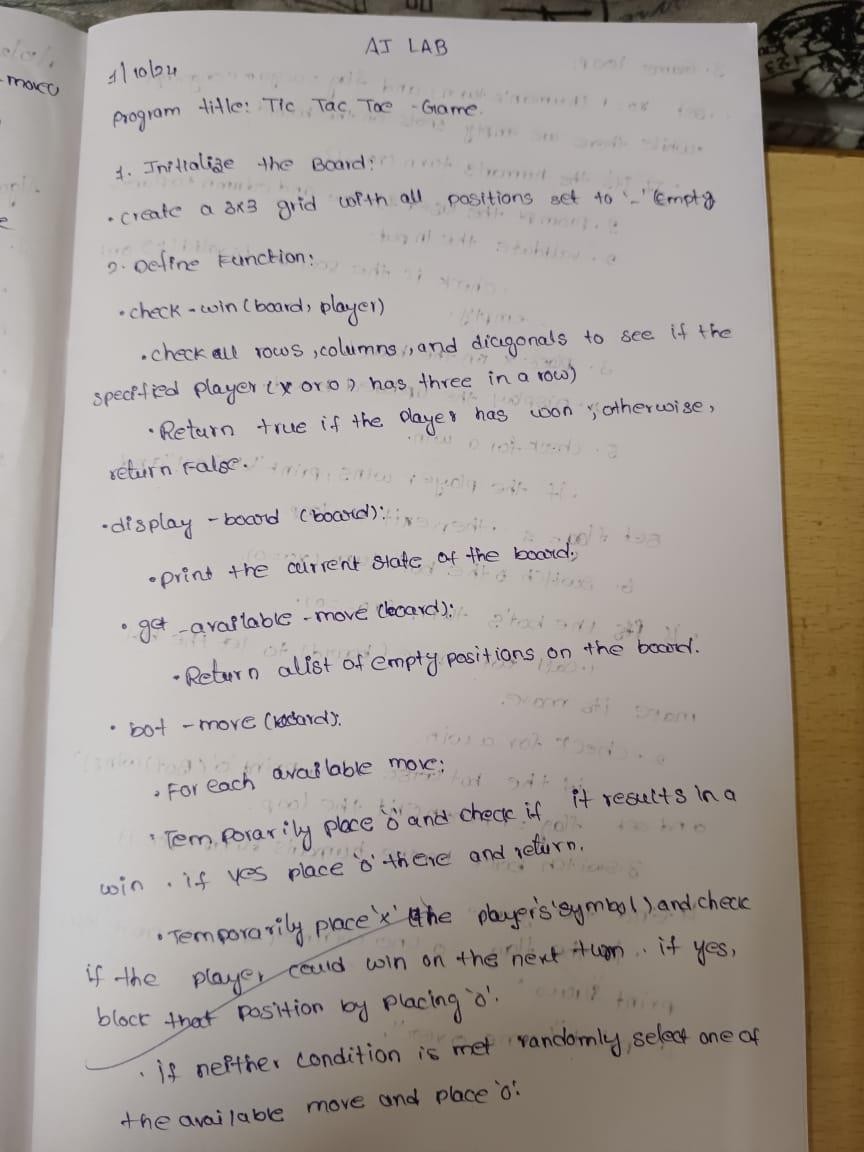
if flag == 0:

print("Draw") print("Game Over") **Output:**







**Algorithm:**

