**PROGRAM [1]:**

import random

class TicTacToe:

def \_\_init\_\_(self):

self.board = [['-' for \_ in range(3)] for \_ in range(3)]

def get\_random\_first\_player(self):

return random.choice(['X', 'O'])

def fix\_spot(self, row, col, player):

self.board[row][col] = player

def is\_player\_win(self, player):

n = len(self.board)

for i in range(n):

if all(self.board[i][j] == player for j in range(n)) or all(self.board[j][i] == player for j in range(n)):

return True

if all(self.board[i][i] == player for i in range(n)) or all(self.board[i][n - 1 - i] == player for i in range(n)):

return True

return False

def is\_board\_filled(self):

return all(item != '-' for row in self.board for item in row)

def swap\_player\_turn(self, player):

return 'X' if player == 'O' else 'O'

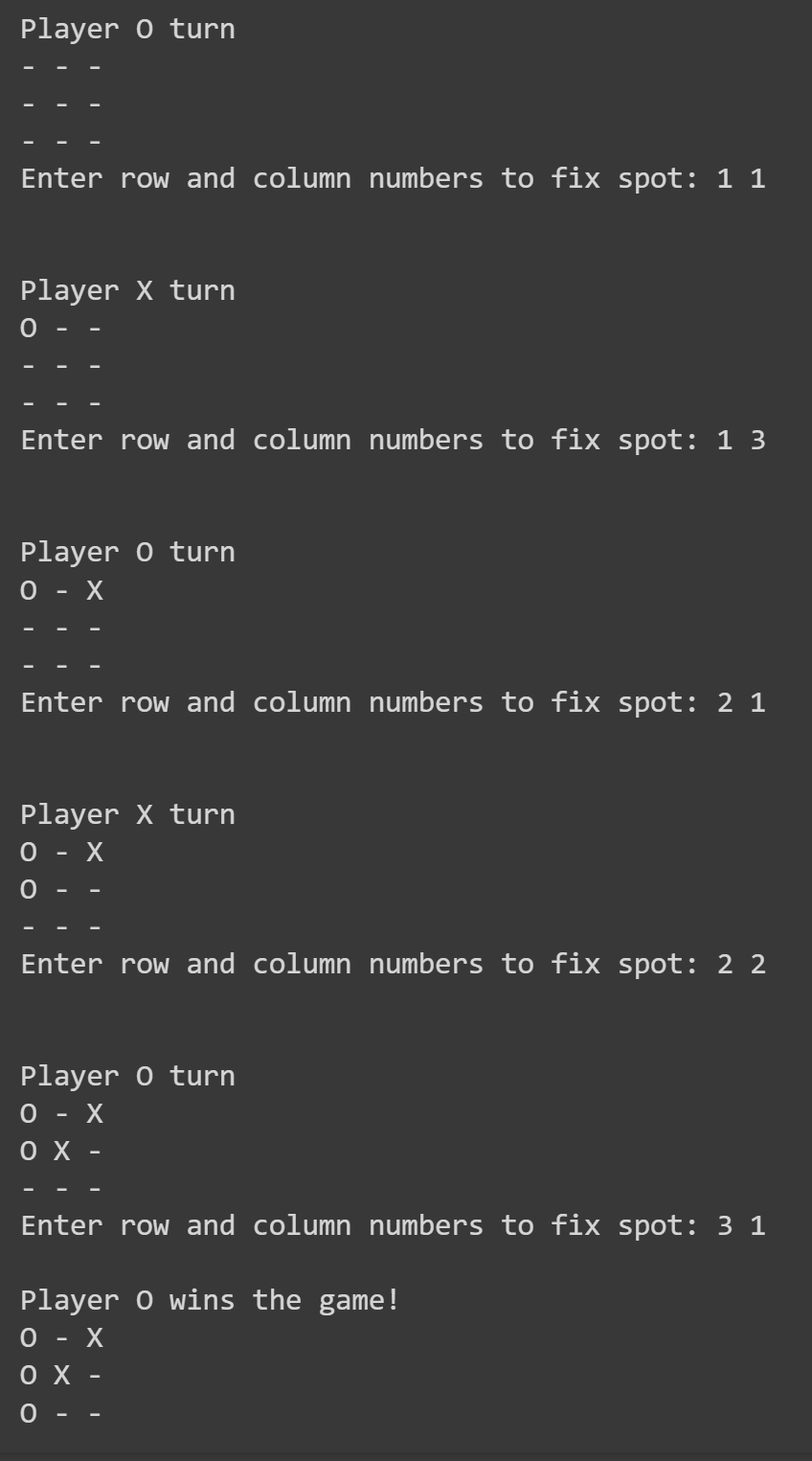
def show\_board(self):

for row in self.board:

print(\*row)

def start(self):

**OUTPUT [1]:**



self.get\_random\_first\_player()

player = 'X' if self.get\_random\_first\_player() == 'X' else 'O'

while True:

print(f"Player {player} turn")

self.show\_board()

row, col = map(int, input("Enter row and column numbers to fix spot: ").split())

print()

self.fix\_spot(row - 1, col - 1, player)

if self.is\_player\_win(player):

print(f"Player {player} wins the game!")

break

if self.is\_board\_filled():

print("Match Draw!")

break

player = self.swap\_player\_turn(player)

print()

self.show\_board()

tic\_tac\_toe = TicTacToe()

tic\_tac\_toe.start()