**9.A Bot to Play Tic Tac Toe**

def print\_board(board):

for row in board:

print(" | ".join(row))

print("-" \* 5)

def check\_winner(board, player):

# Check rows, columns, and diagonals

for i in range(3):

if all(board[i][j] == player for j in range(3)) or all(board[j][i] == player for j 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 is\_board\_full(board):

return all(board[i][j] != " " for i in range(3) for j in range(3))

def tic\_tac\_toe():

board = [[" " for \_ in range(3)] for \_ in range(3)]

players = ["X", "O"]

current\_player = players[0]

while True:

print\_board(board)

# Get player move

while True:

row = int(input("Enter row (0, 1, or 2): "))

col = int(input("Enter column (0, 1, or 2): "))

if 0 <= row < 3 and 0 <= col < 3 and board[row][col] == " ":

break

else:

print("Invalid move. Try again.")

# Make the move

board[row][col] = current\_player

# Check for a winner

if check\_winner(board, current\_player):

print\_board(board)

print(f"Player {current\_player} wins!")

break

# Check for a tie

if is\_board\_full(board):

print\_board(board)

print("It's a tie!")

break

# Switch to the other player

current\_player = players[1] if current\_player == players[0] else players[0]

if \_\_name\_\_ == "\_\_main\_\_":

tic\_tac\_toe()

