TIC-TAC-TOE code

def print\_board(board):

for row in board:

print(" | ".join(row))

print("-" \* 9)

def check\_winner(board, player):

# Check rows, columns, and diagonals and define the players win or tie

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):

#To insert the 'X' or 'O' values into the cells

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

def get\_player\_move():

#To get the input from the players both 'X' and 'O'

while True:

try:

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

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

if 0 <= row < 3 and 0 <= col < 3:

return row, col

else:

print("Invalid move. Row and column must be between 0 and 2.")

except ValueError:

print("Invalid input. Please enter a number.")

def play\_tic\_tac\_toe():

#To run the game untill the one player will or tie of the match

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

current\_player = 'X'

while True:

print\_board(board)

print(f"Player {current\_player}'s turn.")

row, col = get\_player\_move()

if board[row][col] == ' ':

board[row][col] = current\_player

else:

print("Invalid move. The cell is already occupied. Try again.")

continue

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

print\_board(board)

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

break

elif is\_board\_full(board):

print\_board(board)

print("It's a tie!")

break

current\_player = 'O' if current\_player == 'X' else 'X'

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

play\_tic\_tac\_toe()