import random

def print\_board(board):

print("\n".join([" | ".join(row) for row in board]))

def check\_win(board, player):

win\_conditions = [

[board[0][0], board[0][1], board[0][2]], # Row 1

[board[1][0], board[1][1], board[1][2]], # Row 2

[board[2][0], board[2][1], board[2][2]], # Row 3

[board[0][0], board[1][0], board[2][0]], # Column 1

[board[0][1], board[1][1], board[2][1]], # Column 2

[board[0][2], board[1][2], board[2][2]], # Column 3

[board[0][0], board[1][1], board[2][2]], # Diagonal \

[board[2][0], board[1][1], board[0][2]] # Diagonal /

]

return [player, player, player] in win\_conditions

def check\_full(board):

return all(cell != ' ' for row in board for cell in row)

def player\_move(board):

while True:

try:

move = int(input("Enter your move (1-9): ")) - 1

row, col = divmod(move, 3)

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

board[row][col] = 'X'

break

else:

print("Cell already taken, try again.")

except (ValueError, IndexError):

print("Invalid input, please enter a number between 1 and 9.")

def computer\_move(board):

empty\_cells = [(row, col) for row in range(3) for col in range(3) if board[row][col] == ' ']

row, col = random.choice(empty\_cells)

board[row][col] = 'O'

def tic\_tac\_toe():

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

print("Welcome to Tic-Tac-Toe!")

print\_board(board)

for turn in range(9):

if turn % 2 == 0:

player\_move(board)

else:

computer\_move(board)

print\_board(board)

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

print("Congratulations! You win!")

break

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

print("Computer wins! Better luck next time.")

break

elif check\_full(board):

print("It's a draw!")

break

tic\_tac\_toe()

Output:

