Task:1

Code:

import random

def print\_board(board):

"""Prints the Tic-Tac-Toe board."""

print("-------------")

for row in board:

print("|", row[0], "|", row[1], "|", row[2], "|")

print("-------------")

def check\_winner(board):

"""Checks if there is a winner or if the board is full."""

for row in board:

if row[0] == row[1] == row[2] != ' ':

return row[0]

for col in range(3):

if board[0][col] == board[1][col] == board[2][col] != ' ':

return board[0][col]

if board[0][0] == board[1][1] == board[2][2] != ' ':

return board[0][0]

if board[0][2] == board[1][1] == board[2][0] != ' ':

return board[0][2]

for row in board:

for cell in row:

if cell == ' ':

return None

return 'tie'

def evaluate(board):

"""Evaluates the current position for the minimax algorithm."""

winner = check\_winner(board)

if winner == 'X':

return 1

elif winner == 'O':

return -1

else:

return 0 # Tie

def minimax(board, depth, is\_maximizing):

"""Minimax algorithm implementation."""

if check\_winner(board) is not None:

return evaluate(board)

if is\_maximizing:

best\_value = -float('inf')

for row in range(3):

for col in range(3):

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

board[row][col] = 'X'

value = minimax(board, depth + 1, False)

board[row][col] = ' '

best\_value = max(best\_value, value)

return best\_value

else:

best\_value = float('inf')

for row in range(3):

for col in range(3):

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

board[row][col] = 'O'

value = minimax(board, depth + 1, True)

board[row][col] = ' '

best\_value = min(best\_value, value)

return best\_value

def get\_best\_move(board):

"""Gets the best move for the computer using minimax."""

best\_move = None

best\_value = -float('inf')

for row in range(3):

for col in range(3):

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

board[row][col] = 'X'

move\_value = minimax(board, 0, False)

board[row][col] = ' '

if move\_value > best\_value:

best\_value = move\_value

best\_move = (row, col)

return best\_move

def main():

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

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

print\_board(board)

while True:

while True:

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

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

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

board[row][col] = 'O'

break

else:

print("That spot is taken! Try again.")

print\_board(board)

winner = check\_winner(board)

if winner:

if winner == 'tie':

print("It's a tie!")

else:

print(f"Congratulations! {winner} wins!")

break

print("Computer's turn...")

row, col = get\_best\_move(board)

board[row][col] = 'X'

print\_board(board)

winner = check\_winner(board)

if winner:

if winner == 'tie':

print("It's a tie!")

else:

print("Computer wins!")

break

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

main()

output:







