LAB 1

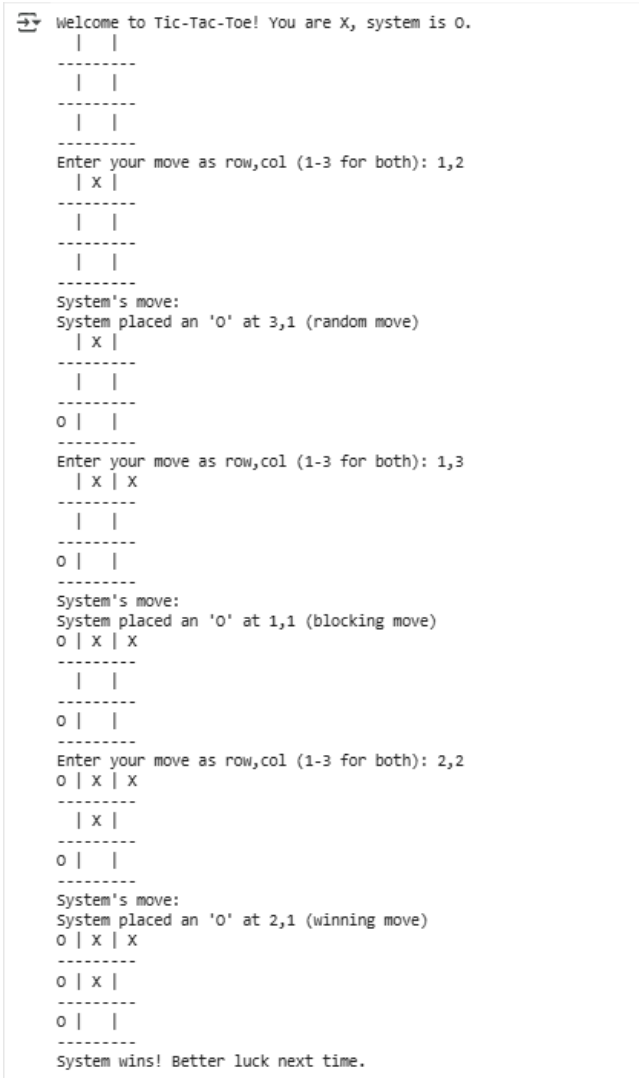
```
import random


def print_board(board):
    for row in board:
        print(" | ".join(row))
        print("-" * 9)


def check_winner(board, player):
    for i in range(3):
        if all(board[i][j] == player for j in range(3)):  # Row
            return True
        if all(board[j][i] == player for j in range(3)):  # Column
            return True
    if all(board[i][i] == player for i in range(3)):      # Diagonal \
        return True
    if all(board[i][2 - i] == player for i in range(3)):  # Diagonal /
        return True
    return False


def check_draw(board):
    return all(cell != " " for row in board for cell in row)


def player_move(board):
    while True:
        try:
            move = input("Enter your move as row,col (1-3 for both): ")
            row, col = map(int, move.split(","))
```

```
            if board[row-1][col-1] != " ":
                print("That spot is already taken.")
                continue
            board[row-1][col-1] = "X"
            break
        except (ValueError, IndexError):
            print("Invalid input. Use format row,col with values from 1 to 3.")

def get_available_moves(board):
    return [(r, c) for r in range(3) for c in range(3) if board[r][c] == " "]

def try_move(board, row, col, player):
    board[row][col] = player
    win = check_winner(board, player)
    board[row][col] = " "  # Undo move
    return win

def system_move(board):
    print("System's move:")

    # 1. Can system win in next move?
    for row, col in get_available_moves(board):
        if try_move(board, row, col, "O"):
            board[row][col] = "O"
            print(f"System placed an 'O' at {row + 1},{col + 1} (winning move)")
            return

    # 2. Can player win in next move? Block it.
    for row, col in get_available_moves(board):
```

```
        if try_move(board, row, col, "X"):
            board[row][col] = "O"
            print(f"System placed an 'O' at {row + 1},{col + 1} (blocking move)")
            return

    # 3. Otherwise, pick a random move
    row, col = random.choice(get_available_moves(board))
    board[row][col] = "O"
    print(f"System placed an 'O' at {row + 1},{col + 1} (random move)")

def main():
    board = [[" "]*3 for _ in range(3)]
    print("Welcome to Tic-Tac-Toe! You are X, system is O.")
    print_board(board)

    while True:
        player_move(board)
        print_board(board)
        if check_winner(board, "X"):
            print("Congratulations! You win!")
            break
        if check_draw(board):
            print("It's a draw!")
            break

        system_move(board)
        print_board(board)
        if check_winner(board, "O"):
            print("System wins! Better luck next time.")
```

```
            break
        if check_draw(board):
            print("It's a draw!")
            break

if __name__ == "__main__":
    main()
```

Output:



```
Welcome to Tic-Tac-Toe! You are X, system is O.
  |   |
---------
  |   |
---------
  |   |
---------
Enter your move as row,col (1-3 for both): 1,2
  | X |
---------
  |   |
---------
  |   |
---------
System's move:
System placed an 'O' at 3,1 (random move)
  | X |
---------
  |   |
---------
O |   |
---------
Enter your move as row,col (1-3 for both): 1,3
  | X | X
---------
  |   |
---------
O |   |
---------
System's move:
System placed an 'O' at 1,1 (blocking move)
O | X | X
---------
  |   |
---------
O |   |
---------
Enter your move as row,col (1-3 for both): 2,2
O | X | X
---------
  | X |
---------
O |   |
---------
System's move:
System placed an 'O' at 2,1 (winning move)
O | X | X
---------
O | X |
---------
O |   |
---------
System wins! Better luck next time.
```