//tictactoe nonAI

import java.util.Scanner;

public class TicTacToe\_NonAI

{

    public static void main(String[] args)

    {

        char[][] board = new char[3][3];

        initializeBoard(board);

        char currentPlayer = 'X';

        boolean gameOver = false;

        while (!gameOver)

        {

            printBoard(board);

            playerMove(board, currentPlayer);

            if (checkWin(board, currentPlayer)) {

                printBoard(board);

                System.out.println(currentPlayer + " wins!");

                gameOver = true;

            } else if (isBoardFull(board)) {

                printBoard(board);

                System.out.println("It's a draw!");

                gameOver = true;

            }

            currentPlayer = (currentPlayer == 'X') ? 'O' : 'X';

        }

    }

    private static void initializeBoard(char[][] board) {

        for (int i = 0; i < 3; i++) {

            for (int j = 0; j < 3; j++) {

                board[i][j] = '-';

            }

        }

    }

    private static void printBoard(char[][] board) {

        for (int i = 0; i < 3; i++) {

            for (int j = 0; j < 3; j++) {

                System.out.print(board[i][j] + " ");

            }

            System.out.println();

        }

    }

    private static void playerMove(char[][] board, char currentPlayer)

    {

        Scanner scanner = new Scanner(System.in);

        int row, col;

        do {

            System.out.print("Player " + currentPlayer + ", enter row (1-3) and column (1-3) for your move (e.g., 1 2): ");

            row = scanner.nextInt() - 1;

            col = scanner.nextInt() - 1;

        } while (row < 0 || row > 2 || col < 0 || col > 2 || board[row][col] != '-');

        board[row][col] = currentPlayer;

    }

    private static boolean isBoardFull(char[][] board) {

        for (int i = 0; i < 3; i++) {

            for (int j = 0; j < 3; j++) {

                if (board[i][j] == '-') {

                    return false;

                }

            }

        }

        return true;

    }

    private static boolean checkWin(char[][] board, char player) {

        // Check rows, columns, and diagonals for a win

        for (int i = 0; i < 3; i++) {

            if (board[i][0] == player && board[i][1] == player && board[i][2] == player) {

                return true;

            }

            if (board[0][i] == player && board[1][i] == player && board[2][i] == player) {

                return true;

            }

        }

        if (board[0][0] == player && board[1][1] == player && board[2][2] == player) {

            return true;

        }

        if (board[0][2] == player && board[1][1] == player && board[2][0] == player) {

            return true;

        }

        return false;

    }

}