import java.util.Scanner;

public class TicTacToe {

static char[][] board = {{' ', ' ', ' '}, {' ', ' ', ' '}, {' ', ' ', ' '}};

static char currentPlayer = 'X';

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int row, col;

boolean gameEnded = false;

// Game loop

while (!gameEnded) {

printBoard();

System.out.println("Player " + currentPlayer + ", it's your turn.");

System.out.print("Enter row (0, 1, or 2): ");

row = scanner.nextInt();

System.out.print("Enter column (0, 1, or 2): ");

col = scanner.nextInt();

// Check if the position is valid

if (row < 0 || row > 2 || col < 0 || col > 2 || board[row][col] != ' ') {

System.out.println("Invalid move, try again.");

continue;

}

// Make the move

board[row][col] = currentPlayer;

// Check if current player wins

if (checkWin()) {

printBoard();

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

gameEnded = true;

} else if (isBoardFull()) {

// Check if the board is full (tie condition)

printBoard();

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

gameEnded = true;

} else {

// Switch players

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

}

}

scanner.close();

}

// Function to print the board

public static void printBoard() {

System.out.println("-------------");

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

System.out.print("| ");

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

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

}

System.out.println();

System.out.println("-------------");

}

}

// Function to check if any player has won

public static boolean checkWin() {

// Check rows and columns

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

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

return true;

}

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

return true;

}

}

// Check diagonals

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

return true;

}

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

return true;

}

return false;

}

// Function to check if the board is full (no more moves possible)

public static boolean isBoardFull() {

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

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

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

return false;

}

}

}

return true;

}

}