import java.util.\*;

import java.util.ArrayList;

import java.util.List;

import java.util.Random;

class Point {

int x, y;

public Point(int x, int y) {

this.x = x;

this.y = y;

}

@Override

public String toString() {

return "[" + x + ", " + y + "]";

}

}

class PointAndScore {

int score;

Point point;

PointAndScore(int score, Point point) {

this.score = score;

this.point = point;

}

}

class Board {

List<Point> availablePoints;

Scanner scan = new Scanner(System.in);

int[][] board = new int[3][3];

public Board() {

}

public boolean isGameOver() {

//Game is over if someone has won, or board is full (draw)

return (hasXWon() || hasOWon() || getAvailableStates().isEmpty());

}

public boolean hasXWon() {

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

//System.out.println("X Diagonal Win");

return true;

}

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

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

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

// System.out.println("X Row or Column win");

return true;

}

}

return false;

}

public boolean hasOWon() {

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

// System.out.println("O Diagonal Win");

return true;

}

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

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

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

// System.out.println("O Row or Column win");

return true;

}

}

return false;

}

public List<Point> getAvailableStates() {

availablePoints = new ArrayList<>();

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

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

if (board[i][j] == 0) {

availablePoints.add(new Point(i, j));

}

}

}

return availablePoints;

}

public void placeAMove(Point point, int player) {

board[point.x][point.y] = player; //player = 1 for X, 2 for O

}

void takeHumanInput() {

System.out.println("Your move: ");

int x = scan.nextInt();

int y = scan.nextInt();

Point point = new Point(x, y);

placeAMove(point, 2);

}

public void displayBoard() {

System.out.println();

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

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

if(board[i][j]==1){

System.out.print("X ");

}else if(board[i][j]==2){

System.out.print("O ");

}else{

System.out.print("- ");

}

}

System.out.println();

}

}

Point computersMove;

public int minimax(int depth, int turn) {

if (hasXWon()) return +10;

if (hasOWon()) return -10;

List<Point> pointsAvailable = getAvailableStates();

if (pointsAvailable.isEmpty()) return 0;

int min = Integer.MAX\_VALUE, max = Integer.MIN\_VALUE;

for (int i = 0; i < pointsAvailable.size(); ++i) {

Point point = pointsAvailable.get(i);

if (turn == 1) {

placeAMove(point, 1);

int currentScore = minimax(depth + 1, 2);

max = Math.max(currentScore, max);

if(currentScore >= 0){ if(depth == 0) computersMove = point;}

if(currentScore == 1){board[point.x][point.y] = 0; break;}

if(i == pointsAvailable.size()-1 && max < 0){if(depth == 0)computersMove = point;}

} else if (turn == 2) {

placeAMove(point, 2);

int currentScore = minimax(depth + 1, 1);

min = Math.min(currentScore, min);

if(min == -1){board[point.x][point.y] = 0; break;}

}

board[point.x][point.y] = 0; //Reset this point

}

return turn == 1?max:min;

}

}

public class TicTacToe {

public static void main(String[] args) {

Board b = new Board();

Random rand = new Random();

Scanner input= new Scanner(System.in);

b.displayBoard();

System.out.println("Who plays first? (1) Computer (2) User? \n(Enter 1 or 2): ");

int choice = b.scan.nextInt();

if(choice == 1){

Point p = new Point(rand.nextInt(3), rand.nextInt(3)); //plays at a random box

b.placeAMove(p, 1); // 1 is computer and computer plays as X

b.displayBoard();

}

while (!b.isGameOver()) {

System.out.println("Your move: ");

int x=input.nextInt();

int y=input.nextInt();

while (b.board[x][y] == 1 || b.board[x][y] == 2) {

System.out.println("Invalid Move. Enter other position");

x=input.nextInt();

y=input.nextInt();

}

Point userMove = new Point(x, y);

b.placeAMove(userMove, 2); //2 for O and O is the user

b.displayBoard();

if (b.isGameOver()) break;

b.minimax(0, 1);

b.placeAMove(b.computersMove, 1);

b.displayBoard();

}

if (b.hasXWon())

System.out.println("Unfortunately, you lost!");

else if (b.hasOWon())

System.out.println("You win!"); //Will never happen ;)

else

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

}

}