

```

1 class Solution
2 {
3     static class Move
4     {
5         int row, col;
6     };
7
8     static char player = 'x', opponent = 'o';
9
10    static Boolean isMovesLeft(char board[][])
11    {
12        for (int i = 0; i < 3; i++)
13            for (int j = 0; j < 3; j++)
14                if (board[i][j] == '_')
15                    return true;
16        return false;
17    }
18
19    static int evaluate(char b[][])
20    {
21        for (int row = 0; row < 3; row++)
22        {
23            if (b[row][0] == b[row][1] &&
24                b[row][1] == b[row][2])
25            {
26                if (b[row][0] == player)
27                    return +10;
28                else if (b[row][0] == opponent)
29                    return -10;
30            }
31        }
32
33        for (int col = 0; col < 3; col++)
34        {
35            if (b[0][col] == b[1][col] &&
36                b[1][col] == b[2][col])
37            {
38                if (b[0][col] == player)
39                    return +10;
40
41                else if (b[0][col] == opponent)
42                    return -10;
43            }
44        }
45
46        if (b[0][0] == b[1][1] && b[1][1] == b[2][2])
47        {
48            if (b[0][0] == player)
49                return +10;
50            else if (b[0][0] == opponent)
51                return -10;
52        }
53
54        if (b[0][2] == b[1][1] && b[1][1] == b[2][0])
55        {
56            if (b[0][2] == player)
57                return +10;

```

```

58     else if (b[0][2] == opponent)
59         return -10;
60     }
61
62     return 0;
63 }
64
65 static int minimax(char board[][],
66     int depth, Boolean isMax)
67 {
68     int score = evaluate(board);
69
70     if (score == 10)
71         return score;
72
73     if (score == -10)
74         return score;
75
76     if (isMovesLeft(board) == false)
77         return 0;
78
79     if (isMax)
80     {
81         int best = -1000;
82
83         for (int i = 0; i < 3; i++)
84         {
85             for (int j = 0; j < 3; j++)
86             {
87                 if (board[i][j] == '_')
88                 {
89                     board[i][j] = player;
90
91                     best = Math.max(best, minimax(board,
92                         depth + 1, !isMax));
93
94                     board[i][j] = '_';
95                 }
96             }
97         }
98         return best;
99     }
100
101     else
102     {
103         int best = 1000;
104
105         for (int i = 0; i < 3; i++)
106         {
107             for (int j = 0; j < 3; j++)
108             {
109                 if (board[i][j] == '_')
110                 {
111                     board[i][j] = opponent;
112
113                     best = Math.min(best, minimax(board,
114                         depth + 1, !isMax));
115

```

```

116     board[i][j] = '_';
117 }
118 }
119 }
120 return best;
121 }
122 }
123
124 static Move findBestMove(char board[][])
125 {
126     int bestVal = -1000;
127     Move bestMove = new Move();
128     bestMove.row = -1;
129     bestMove.col = -1;
130
131     for (int i = 0; i < 3; i++)
132     {
133         for (int j = 0; j < 3; j++)
134         {
135             if (board[i][j] == '_')
136             {
137                 board[i][j] = player;
138
139                 int moveVal = minimax(board, 0, false);
140
141                 board[i][j] = '_';
142
143                 if (moveVal > bestVal)
144                 {
145                     bestMove.row = i;
146                     bestMove.col = j;
147                     bestVal = moveVal;
148                 }
149             }
150         }
151     }
152
153     System.out.printf("The value of the best Move " +
154         "is : %d\n\n", bestVal);
155
156     return bestMove;
157 }
158
159 public static void main(String[] args)
160 {
161     char board[][] = { { 'x', 'o', 'x' },
162         { 'o', 'o', 'x' },
163         { '_', '_', '_' } };
164
165     Move bestMove = findBestMove(board);
166
167     System.out.printf("The Optimal Move is :\n");
168     System.out.printf("ROW: %d COL: %d\n\n",
169         bestMove.row, bestMove.col );
170 }
171 }

```