

```
1: import java.io.BufferedReader;
2: import java.io.InputStreamReader;
3:
4:
5: public class Dive {
6:
7:     public static void main(String[] args) {
8:         BufferedReader bfr = new BufferedReader(new InputStreamReader(System.in));
9:
10:        try {
11:
12:            boolean run = true;
13:            String line;
14:            while (((line = bfr.readLine()) != null) && run) {
15:                String[] lineEl = line.split(" ");
16:                int rowCount = Integer.parseInt(lineEl[0]);
17:                int colCount = Integer.parseInt(lineEl[1]);
18:                if (rowCount == 0 && colCount == 0) {
19:                    break;
20:                }
21:
22:                char falconChar = lineEl[2].charAt(1);
23:                char[][] original = new char[rowCount][colCount];
24:                char[][] image = new char[rowCount][colCount];
25:                char[][] backGround = new char[rowCount][colCount];
26:                char[][] result = new char[rowCount][colCount];
27:
28:                boolean[][] diffs = new boolean[rowCount][colCount];
29:                int diffsCount = 0;
30:                int falconChars = 0;
31:
32:                int minRowOr, minColOr, maxColOr, maxRowOr;
33:                minColOr = colCount - 1;
34:                minRowOr = rowCount - 1;
35:                maxColOr = 0;
36:                maxRowOr = 0;
37:
38:                int minColIm, maxColIm, maxRowIm, minRowIm;
39:                minColIm = colCount - 1;
40:                minRowIm = rowCount - 1;
41:                maxColIm = 0;
42:                maxRowIm = 0;
43:
44:                int rowMove, colMove;
45:
46:
47:                for (int k = 0; k < 2; k++) {
48:                    for (int i = 0; i < rowCount; i++) {
```

```
49:         String row = bfr.readLine();
50:
51:         for (int j = 0; j < colCount; j++) {
52:             char currChar = row.charAt(j);
53:
54:             //background chars
55:             if (currChar != falconChar) {
56:                 backGround[i][j] = currChar;
57:             }
58:
59:             //original cycle
60:             if (k == 0) {
61:                 original[i][j] = currChar;
62:
63:                 if (currChar == falconChar){
64:                     falconChars++;
65:                 }
66:
67:                 if (i > maxRowOr) {
68:                     maxRowOr = i;
69:                 }
70:
71:                 if (i < minRowOr) {
72:                     minRowOr = i;
73:                 }
74:
75:                 if (j > maxColOr) {
76:                     maxColOr = j;
77:                 }
78:
79:                 if (j < minColOr) {
80:                     minColOr = j;
81:                 }
82:
83:             } else {
84:                 image[i][j] = currChar;
85:
86:                 if (original[i][j] != image[i][j]) {
87:                     diffs[i][j] = true;
88:                     diffsCount++;
89:
90:                     if (i > maxRowIm) {
91:                         maxRowIm = i;
92:                     }
93:
94:                     if (i < minRowIm) {
95:                         minRowIm = i;
96:                     }
```

```
97:
98:                                     if (j > maxColIm) {
99:                                         maxColIm = j;
100:                                     }
101:
102:                                     if (j < minColIm) {
103:                                         minColIm = j;
104:                                     }
105:
106:                                     } else {
107:                                         diffs[i][j] = false;
108:                                     }
109:                                 }
110:
111:                            }
112:                        }
113:                    bfr.readLine();
114:                }
115:
116:                //there is no falcon in silluethe
117:                if (diffsCount == falconChars) {
118:                    printCharField(image);
119:                } else {
120:
121:                    //we need to catch correct corner
122:                    int currRowOr, currColOr;
123:
124:                    //top-left corner
125:                    rowMove = minRowOr - minRowIm;
126:                    colMove = minColOr - minColIm;
127:
128:                    boolean allOk = true;
129:                    copyFiels(backGround, result);
130:                    for (int i = minRowIm; i <= maxRowIm && allOk; i++) {
131:                        currRowOr = i + rowMove;
132:
133:                        for (int j = minColIm; j <= maxColIm && allOk; j++) {
134:                            currColOr = j + colMove;
135:
136:                            //we didnt exceeded original field borders
137:                            if (currColOr >= 0 && currColOr < colCount
138:                                && currRowOr >= 0 && currRowOr < rowCount) {
139:
140:                                if (((image[i][j] == falconChar) &&
(original[currRowOr][currColOr] != falconChar))
141:                                    || ((image[i][j] != falconChar) &&
(original[currRowOr][currColOr] == falconChar))) {
142:                                    allOk = false;
```

```
143:                                     }
144:                                 }
145:
146:
147:                                 }
148:                             }
149:
150:                             //we found a solution
151:                             if (allOk) {
152:
153:                             }
154:                         }
155:                     }
156:
157:     } catch (Exception e) {
158:         e.printStackTrace();
159:     }
160:
161: }
162:
163: public static void copyFiels(char[][] fromField, char[][] toField) {
164:     for (int i = 0; i < fromField.length; i++) {
165:         for (int j = 0; j < fromField[0].length; j++) {
166:             toField[i][j] = fromField[i][j];
167:         }
168:     }
169:
170: }
171:
172: public static void printCharField(char[][] field) {
173:     for (int i = 0; i < field.length; i++) {
174:         for (int j = 0; j < field[0].length; j++) {
175:             System.out.print(field[i][j]);
176:         }
177:         System.out.println();
178:     }
179: }
180: }
```