```java
 1: import java.io.BufferedReader;
 2: import java.io.InputStreamReader;
 3: import java.util.*;
 4: import java.util.stream.Collectors;
 5:
 6: /**
 7:  * Created by tym12 on 10/22/16.
 8:  */
 9: public class Tribune {
10:
11:     public static void main(String[] args)  {
12:
13:         Scanner scanner = new Scanner(System.in);
14:
15:         // write your code here
16:
17:         BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));
18:         String nrS;
19:         try {
20:             do {
21:                 ArrayList<String> gr = new ArrayList<>();
22:                 ArrayList<char[]> chars = new ArrayList<>();
23:                 nrS= bufferedReader.readLine();
24:                 int n = Integer.parseInt(nrS);
25:                 char wrong;
26:
27:                 for (int i= 0; i < n; i++)
28:                 {
29:                     String l = bufferedReader.readLine();
30:                     gr.add(l);
31:                     char[] line = l.toCharArray();
32:                     Arrays.sort(line);
33:                     chars.add(line);
34:
35:
36:                     //System.out.println(line);
37:                     //System.out.println(l);
38:                 }
39:
40:                 //int bad = -1;
41:                 char bad = 0;
42:
43:                 char bad2 = 0;
44:                 char realyBad = 0;
45:                 for (int i = 1; i < n; i ++) {
46:                     if (!chars.get(i).equals(chars.get(i-1)))
47:                     {
48:                         Set<Character> s = uniques(gr.get(i),gr.get(i-1));
```

```java
49:                        Set<Character> s2 = uniques(gr.get(i-1),gr.get(i));
50:
51:                        int found = 0;
52:                        for (char c :s)
53:                        {
54:                            System.out.println("t2 "+c);
55:                            bad = c;
56:                            found ++;
57:                            break;
58:
59:                        }
60:                        for (char c :s2)
61:                        {
62:                            System.out.println("t3 "+c);
63:                            bad2 = c;
64:                            found ++;
65:                            break;
66:
67:                        }
68:                        if (found == 0)
69:                        {
70:                            char last = 0;
71:                            for (int j = 0; j < n; j++) {
72:                                if (chars.get(i)[j] != chars.get(i-1)[j]) {
73:                                    if (last != 0) {
74:                                        realyBad = last;
75:                                        break;
76:                                    }
77:                                }
78:                                else
79:                                    last = chars.get(i)[j];
80:                            }
81:                            System.out.println("test "+realyBad+" "+last+" "+s.size() +" "+s2.size());
82:                            //break;
83:                        }
84:                        if (realyBad != 0)
85:                            break;
86:
87:                        //break;
88:                    }
89:
90:                }
91:
92:
93:            int b1 = 0;
94:            int b2 = 0;
95:            int badRow = 0;
96:            if (realyBad == 0) {
```

```java
 97:                    for (int i = 0; i < n; i++) {
 98:                        if (gr.get(i).contains("" + bad))
 99:                            b1++;
100:
101:                        if (gr.get(i).contains("" + bad2))
102:                            b2++;
103:                        if (b1 > 1) {
104:                            realyBad = bad2;
105:                            break;
106:                        }
107:                        if (b2 > 1) {
108:                            realyBad = bad;
109:                            break;
110:                        }
111:                    }
112:
113:                }
114:                for (int i = 0; i < n; i++) {
115:                    int index = gr.get(i).indexOf(realyBad);
116:
117:                    if (index != -1)
118:                    {
119:                        char correct = 0;
120:
121:                        Set<Character> g = uniques(new String(chars.get(0)),gr.get(i));
122:                        for (char c : g)
123:                        {
124:                            System.out.println((i+1)+" "+(index+1)+" "+c);
125:                            break;
126:                        }
127:                        break;
128:                    }/*
129:                        continue;
130:
131:                    for (int j = 0; j <n; j++)
132:                    {
133:                        int nr = 0;
134:                        int index = -1;
135:                        for (int k = 0; k < n; k++) {
136:                            if (gr.get(bad).charAt(k) == chars.get(i)[j]) {
137:                                nr++;
138:                                if (index == -1)
139:                                    index = k;
140:                            }
141:                        }
142:                        if (nr > 1 || nr == 0)
143:                        {
144:                            if (index==-1) index = 0;
```

```java
145:                    correct = chars.get(i)[j];
146:                    System.out.println((bad+1)+" "+(index+1)+" "+correct);
147:                    break;
148:                }*/
149:
150:               }
151:               /*
152:               break;
153:
154:              }*/
155: //          System.out.println();
156:
157:           }
158:           while(nrS != null);
159:       } catch (Exception e) {
160:           //
161:       }
162:
163:    }
164:
165:    private static Set<Character> uniques(String first, String second) {
166:        Set<Character> a = first.chars().mapToObj(c -> (char) c).collect(Collectors.toSet());
167:        Set<Character> b = second.chars().mapToObj(c -> (char) c).collect(Collectors.toSet());
168:        a.removeAll(b);
169:        return a;
170:    }
171:
172: }
```