

```
1 //Vijay Damineni
2 //Junior
3 //Contest3 veitch
4 package vijay;
5 import java.util.Arrays;
6 import java.util.Scanner;
7
8 public class VijayDC3JrVeitich2020 {
9
10     public static String HextoBinary(String InputA, int num) {
11         char firstChar= InputA.charAt(num);
12         String binarynumber= "";
13         switch(firstChar) {
14             case 'A':
15                 binarynumber = "1010";
16                 break;
17             case 'B':
18                 binarynumber = "1011";
19                 break;
20             case 'C':
21                 binarynumber = "1100";
22                 break;
23             case 'D':
24                 binarynumber = "1101";
25                 break;
26             case 'E':
27                 binarynumber = "1110";
28                 break;
29             case 'F':
30                 binarynumber = "1111";
31                 break;
32             case '0':
33                 binarynumber="0000";
34                 break;
35             case '1':
36                 binarynumber = "0001";
37                 break;
38             case '2':
39                 binarynumber = "0010";
40                 break;
41             case '3':
42                 binarynumber = "0011";
43                 break;
44             case '4':
45                 binarynumber = "0100";
46                 break;
47             case '5':
48                 binarynumber = "0101";
49                 break;
50             case '6':
51                 binarynumber = "0110";
52                 break;
53             case '7':
54                 binarynumber = "0111";
55                 break;
56             case '8':
57                 binarynumber = "1000";
58                 break;
59             case '9':
60                 binarynumber = "1001";
61                 break;
62
63         }
64
65         return binarynumber;
66
67
68
69     }
```

```

70 public static String[][] Make2DArray(String InputA) {
71     String[][] chart = new String[2][4];
72     String FirstRow = HextoBinary(InputA,0);
73     String SecondRow = HextoBinary(InputA,1);
74     //System.out.println("firstrow: "+FirstRow);
75     //System.out.println("secondrow: "+SecondRow);
76     // chart[0][0].equals(FirstRow.charAt(0));
77     // chart[0][1].equals(FirstRow.charAt(1));
78     // chart[0][2].equals(FirstRow.charAt(2));
79     // chart[0][3].equals(FirstRow.charAt(3));
80     // chart[1][0].equals(SecondRow.charAt(0));
81     // chart[1][1].equals(SecondRow.charAt(1));
82     // chart[1][2].equals(SecondRow.charAt(2));
83     // chart[1][3].equals(SecondRow.charAt(3));
84
85     chart[0][0] = String.valueOf(FirstRow.charAt(0));
86     chart[0][1] = String.valueOf(FirstRow.charAt(1));
87     chart[0][2] = String.valueOf(FirstRow.charAt(2));
88     chart[0][3] = String.valueOf(FirstRow.charAt(3));
89     chart[1][0] = String.valueOf(SecondRow.charAt(0));
90     chart[1][1] = String.valueOf(SecondRow.charAt(1));
91     chart[1][2] = String.valueOf(SecondRow.charAt(2));
92     chart[1][3] = String.valueOf(SecondRow.charAt(3));
93
94
95     return chart;
96
97
98 }
99 public static String formula(String[][] arr, int i, int j) {
100     String frmla="";
101     if(i==0&&j==0&&arr[i][j].equals("1")) {
102         frmla = "ABc";
103     }
104     if(i==0&&j==1&&arr[i][j].equals("1")) {
105         frmla = "ABC";
106     }
107     if(i==0&&j==2&&arr[i][j].equals("1")) {
108         frmla = "aBC";
109     }
110     if(i==0&&j==3&&arr[i][j].equals("1")) {
111         frmla = "aBc";
112     }
113     if(i==1&&j==0&&arr[i][j].equals("1")) {
114         frmla = "Abc";
115     }
116     if(i==1&&j==1&&arr[i][j].equals("1")) {
117         frmla = "AbC";
118     }
119     if(i==1&&j==2&&arr[i][j].equals("1")) {
120         frmla = "abC";
121     }
122     if(i==1&&j==3&&arr[i][j].equals("1")) {
123         frmla = "abc";
124     }
125
126
127
128     //for (int row = 0; row < 4; row++){
129     //for (int col = 0; col < 2; col++) {
130
131     return frmla;
132
133 }
134
135
136 /*public static String Comparison (String InputA) {
137     //formula(Make2DArray(InputA),i,j)    used to copy and paste
138     int count=0;

```

```

139     while(count!=8) {
140         if(count==0 && formula(Make2DArray(InputA),0,0)!="") {
141             String A01= formula(Make2DArray(InputA),0,0);
142         }
143         if(count==1 && formula(Make2DArray(InputA),0,0)!="") {
144             String A01= formula(Make2DArray(InputA),0,0);
145         }
146         if(count==0 && formula(Make2DArray(InputA),0,0)!="") {
147             String A01= formula(Make2DArray(InputA),0,0);
148         }
149         if(count==0 && formula(Make2DArray(InputA),0,0)!="") {
150             String A01= formula(Make2DArray(InputA),0,0);
151         }
152         if(count==0 && formula(Make2DArray(InputA),0,0)!="") {
153             String A01= formula(Make2DArray(InputA),0,0);
154         }
155         if(count==0 && formula(Make2DArray(InputA),0,0)!="") {
156             String A01= formula(Make2DArray(InputA),0,0);
157         }
158         if(count==0 && formula(Make2DArray(InputA),0,0)!="") {
159             String A01= formula(Make2DArray(InputA),0,0);
160         }
161         if(count==0 && formula(Make2DArray(InputA),0,0)!="") {
162             String A01= formula(Make2DArray(InputA),0,0);
163         }
164     }
165     count++;
166 }
167 return null;
168
169
170 }*/
171 public static String[] combine( String [] Expressions) {
172
173     if (Expressions[0] !="" && Expressions[1]!=""&&Expressions[4] !="" &&
Expressions[5]!="") {
174         Expressions[0]="A";
175         Expressions[1]="";
176         Expressions[4]="";
177         Expressions[5]="";
178     }
179     if (Expressions[1] !="" && Expressions[2]!=""&&Expressions[5] !="" &&
Expressions[6]!="") {
180         Expressions[1]="C";
181         Expressions[2]="";
182         Expressions[6]="";
183         Expressions[5]="";
184     }
185     if (Expressions[2] !="" && Expressions[3]!=""&&Expressions[6] !="" &&
Expressions[7]!="") {
186         Expressions[2]="a";
187         Expressions[3]="";
188         Expressions[6]="";
189         Expressions[7]="";
190     }
191     if (Expressions[0] !="" && Expressions[4]!=""&&Expressions[3] !="" &&
Expressions[7]!="") {
192         Expressions[0]="c";
193         Expressions[4]="";
194         Expressions[3]="";
195         Expressions[7]="";
196     }
197     if (Expressions[0] !="" && Expressions[1]!=""&&Expressions[2] !="" &&
Expressions[3]!="") {
198         Expressions[0]="B";
199         Expressions[1]="";
200         Expressions[2]="";
201         Expressions[3]="";
202     }

```

```

203     if (Expressions[4] != "" && Expressions[5] != "" && Expressions[6] != "" &&
Expressions[7] != "") {
204         Expressions[4] = "b";
205         Expressions[5] = "";
206         Expressions[6] = "";
207         Expressions[7] = "";
208     }
209
210
211
212
213
214
215
216
217 //System.out.println(Arrays.toString(Expressions));
218     if (Expressions[0] != "" && Expressions[1] != "" && Expressions[1].length() == 3 &&
Expressions[0].length() == 3) {
219         Expressions[0] = "AB";
220         Expressions[1] = "";
221     }
222 //System.out.println(Arrays.toString(Expressions));
223     if (Expressions[1] != "" && Expressions[2] != "" && Expressions[1].length() == 3 &&
Expressions[2].length() == 3) {
224         Expressions[1] = "BC";
225         Expressions[2] = "";
226     }
227 //System.out.println(Arrays.toString(Expressions));
228     if (Expressions[2] != "" && Expressions[3] != "" && Expressions[2].length() == 3 &&
Expressions[3].length() == 3) {
229         Expressions[2] = "aB";
230         Expressions[3] = "";
231     }
232 //System.out.println(Arrays.toString(Expressions));
233     if (Expressions[4] != "" && Expressions[5] != "" && Expressions[4].length() == 3 &&
Expressions[5].length() == 3) {
234         Expressions[4] = "Ab";
235         Expressions[5] = "";
236     }
237 //System.out.println(Arrays.toString(Expressions));
238     if (Expressions[5] != "" && Expressions[6] != "" && Expressions[5].length() == 3 &&
Expressions[6].length() == 3) {
239         Expressions[5] = "bC";
240         Expressions[6] = "";
241     }
242 //System.out.println(Arrays.toString(Expressions));
243     if (Expressions[6] != "" && Expressions[7] != "" && Expressions[6].length() == 3 &&
Expressions[7].length() == 3) {
244         Expressions[6] = "ab";
245         Expressions[7] = "";
246     }
247 //System.out.println(Arrays.toString(Expressions));
248
249     if (Expressions[0] != "" && Expressions[4] != "" && Expressions[0].length() == 3 &&
Expressions[4].length() == 3) {
250         Expressions[0] = "Ac";
251         Expressions[4] = "";
252     }
253 //System.out.println(Arrays.toString(Expressions));
254     if (Expressions[1] != "" && Expressions[5] != "" && Expressions[1].length() == 3 &&
Expressions[5].length() == 3) {
255         Expressions[1] = "AC";
256         Expressions[5] = "";
257     }
258 //System.out.println(Arrays.toString(Expressions));
259     if (Expressions[2] != "" && Expressions[6] != "" && Expressions[2].length() == 3 &&
Expressions[6].length() == 3) {
260         Expressions[2] = "aC";
261         Expressions[6] = "";

```

```

262     }
263     if (Expressions[3] != "" && Expressions[7] != "" && Expressions[3].length() == 3 &&
Expressions[7].length() == 3) {
264         Expressions[3] = "ac";
265         Expressions[7] = "";
266         //System.out.println(Arrays.toString(Expressions));
267     }
268     if (Expressions[3] != "" && Expressions[0] != "" && Expressions[0].length() == 3 &&
Expressions[3].length() == 3) {
269         Expressions[0] = "Bc";
270         Expressions[3] = "";
271     }
272     //System.out.println(Arrays.toString(Expressions));
273     if (Expressions[7] != "" && Expressions[4] != "" && Expressions[4].length() == 3 &&
Expressions[7].length() == 3) {
274         Expressions[7] = "bc";
275         Expressions[4] = "";
276     }
277     //System.out.println(Arrays.toString(Expressions));
278     return Expressions;
279 }
280 /*public static String[] Expressions2( String Expressions[]) {
281     String Simplify[] = new String[8];
282     int n = 1;
283     for(int x = 0; x < 8; x++) {
284         for(int j = n; j < 8; j++) {
285             if( Expressions[x] != "" && Expressions[j] != "" &&
Expressions[x].length() == 3 && Expressions[j].length() == 3) {
286
287                 }
288             }
289             n++;
290         }
291         return Expressions;
292     }
293 */
294 public static String OutputStr(String a[]) {
295     String outputString = "";
296     String outputStringA = "";
297     String outputStringB = "";
298     int n = 0;
299     for(int x = 0; x < 8; x++) {
300         if(a[x] != ("") && n == 0) {
301             outputString = a[x];
302             n++;
303         }
304         else {
305
306             if(a[x] != "" && n == 1) {
307                 outputStringA = outputString + "+" + a[x] ;
308                 n++;
309             }
310
311             else {
312                 if(a[x] != "" && n == 2) {
313                     outputStringB = outputStringA + "+" + a[x] ;
314                     n++;
315                 }
316             }
317         }
318     }
319
320
321     if(n == 2) {
322         outputString = outputStringA;
323     }
324
325     if(n == 3) {
326         outputString = outputStringB;

```

```

327
328     }
329
330     return outputString;
331
332 }
333
334 public static void main(String[] args) {
335     // TODO Auto-generated method stub
336
337     Scanner s= new Scanner(System.in);
338
339     String inputdata[] = new String[5];
340
341     for(int i = 0; i < 5; i++)
342     {
343         inputdata[i] = s.nextLine();
344     }
345     s.close();
346
347
348     for (int x=0; x<5;x++)      {
349
350     String InputA= inputdata[x];
351     String[] formulas = new String[8];
352     int i =0;
353     for (int row = 0; row < 2; row++){
354         for (int col = 0; col < 4; col++) {
355
356             formulas[i] = formula(Make2DArray(InputA), row, col);
357             i++;
358
359         }}
360     String a[]= combine(formulas);
361
362     String op = OutputStr(a);
363     op = op.replaceAll("a", "~A");
364     op = op.replaceAll("b", "~B");
365     op = op.replaceAll("c", "~C");
366     System.out.println(op);
367
368     //opString = Arrays.deepToString(a);
369     //System.out.println(Arrays.toString(a));
370
371 }
372
373 //System.out.println(Arrays.toString(formulas));
374
375 //for (int x=0;x< 4;x++) {
376
377 // if (formulas[0] !="" && formulas[1]!="") {
378
379 // }
380 //}
381 }}

```