```java
1  package regextalk.password.toshow;
2
3  import java.util.Arrays;
4  import java.util.regex.Matcher;
5  import java.util.regex.Pattern;
6
7  public class Password01RegexThreeRules {
8      public static final String SPECIAL = "><?.,!@#$%^&*+=_)(\\}\\{\\]\\[";
9      public static final String LKA_LOWER = "(?=.*[a-z])"; //LKA=lookahead
10     public static final String LKA_UPPER = "(?=.*[A-Z])";
11     public static final String LKA_DIGIT = "(?=.*[0-9])";
12     public static final String LKA_SPECIAL = "(?=.*[" + SPECIAL + "])";
13     private static final String REGEX = "" +
14         "^" +                                       //start of input
15           "(?:" +                                   //non capturing group
16               LKA_LOWER + LKA_UPPER + LKA_DIGIT +     //Option 1
17               "|" +                                   //or
18               LKA_LOWER + LKA_UPPER + LKA_SPECIAL +    //Option 2
19               "|" +                                   //or
20               LKA_LOWER + LKA_DIGIT + LKA_SPECIAL +    //Option 3
21               "|" +                                   //or
22               LKA_UPPER + LKA_DIGIT + LKA_SPECIAL +    //Option 4
23           ")" +                                     //end group
24         "[A-Za-z0-9" + SPECIAL + "]" +     //all legal characters
25         "{8,24}" +                           //8 to 24 chars
26     "$";                                   //end of input
27
28     public static void main(String[] ignored) {
29
30         Matcher matcher = Pattern.compile(REGEX).matcher("ignored input");
31
32         Arrays.stream(newDemoInputArray()).forEach(input -> {
33             boolean valid = matcher.reset(input).find();
34             System.out.printf("\"%s\" is %s password.%n",
35                     input, (valid ? "a VALID" : "an invalid"));
36         });
37     }
38
39     public static String[] newDemoInputArray() {
40         return new String[]{
41             "",                         //bad (bad rules, bad length)
42             "abcdefghij",               //bad (bad rules, good length)
43             "abc123",                   //bad (good rules, bad length)
44             "abc123abc",                //3 rules: bad, 2 rules: good
45             "a1$A",                     //bad (good rules, bad length)
46             "abc123$%^ABC",             //good
47             "abcABC123$&*",             //good
48             "abc ABC123$#$"             //bad (whitespace)
49         };
50     }
51 }
52
```