

```

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 Password02RegexTwoRules {
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 + //Option 1
17            "|" + //or
18            LKA_LOWER + LKA_DIGIT + //Option 2
19            "|" + //or
20            LKA_LOWER + LKA_SPECIAL + //Option 3
21            "|" + //or
22            LKA_UPPER + LKA_DIGIT + //Option 4
23            "|" + //or
24            LKA_UPPER + LKA_SPECIAL + //Option 5
25            "|" + //or
26            LKA_DIGIT + LKA_SPECIAL + //Option 6
27        ")" + //end group
28        "[A-Za-z0-9" + SPECIAL + "]" + //All possible characters
29        "{8,24}" + //8 to 24 chars
30        "$"; //end of input
31
32    public static void main(String[] ignored) {
33        Matcher matcher = Pattern.compile(REGEX).matcher("ignored input");
34
35        Arrays.stream(new DemoInputArray()).forEach(input -> {
36            boolean valid = matcher.reset(input).find();
37            System.out.printf("\n%s\n is %s password.%n",
38                input, (valid ? "a VALID" : "an invalid"));
39        });
40    }
41
42    public static String[] newDemoInputArray() {
43        return new String[]{
44            "", //bad (bad rules, bad length)
45            "abcdefghij", //bad (bad rules, good length)
46            "abc123", //bad (good rules, bad length)
47            "abc123abc", //3 rules: bad, 2 rules: good
48            "a1$A", //bad (good rules, bad length)
49            "abc123$%^ABC", //good
50            "abcABC123$&*+", //good
51            "abc ABC123$#$" //bad (whitespace)
52        };
53    }
54 }
55

```