StringCompression.java

```
1    package com.example;
2
3    public class StringCompression {
4        /**
5         * Returns the compressed or encoded string
6         *
7         * @param ch character array that contains the group of characters to be encoded
8         * @return the compressed character array as string
9         */
10       public static String compress(String input) {
11           // Keeping the count as 1 since every element present will have atleast a count
12           // of 1
13           int count = 1;
14           String compressedString = "";
15           // Base condition to check whether the array is of size 1, if it is then we
16           // return the array
17 1         if (input.length() == 1) {
18 1             return "" + input.charAt(0);
19           }
20           // If the array has a length greater than 1 we move into this loop
21 4         for (int i = 0; i < input.length() - 1; i++) {
22               // here we check for similarity of the adjacent elements and change the count
23               // accordingly
24 2             if (input.charAt(i) == input.charAt(i + 1)) {
25 1                 count = count + 1;
26               }
27 5             if ((i + 1) == input.length() - 1 && input.charAt(i + 1) == input.charAt(i)) {
28                   compressedString = appendCount(compressedString, count, input.charAt(i));
29                   break;
30 2             } else if (input.charAt(i) != input.charAt(i + 1)) {
31 3                 if ((i + 1) == input.length() - 1) {
32 1                     compressedString = appendCount(compressedString, count, input.charAt(i)) + input.charAt(i + 1);
33                       break;
34                   } else {
35                       compressedString = appendCount(compressedString, count, input.charAt(i));
36                       count = 1;
37                   }
38               }
39           }
40 1         return compressedString;
41       }
42       /**
43        * @param res    the resulting string
44        * @param count current count
45        * @param ch     the character at a particular index
46        * @return the res string appended with the count
47        */
48       public static String appendCount(String res, int count, char ch) {
49 2         if (count > 1) {
50               res += ch + "" + count;
51               count = 1;
52           } else {
53               res += ch + "";
54           }
55 1         return res;
56       }
57   }
```

Mutations

17	1. negated conditional → KILLED
18	1. replaced return value with "" for com/example/StringCompression::compress → KILLED
21	1. changed conditional boundary → SURVIVED 2. Changed increment from 1 to -1 → KILLED 3. Replaced integer subtraction with addition → KILLED 4. negated conditional → KILLED
24	1. Replaced integer addition with subtraction → KILLED 2. negated conditional → KILLED
25	1. Replaced integer addition with subtraction → KILLED
27	1. Replaced integer addition with subtraction → KILLED 2. Replaced integer subtraction with addition → KILLED 3. Replaced integer addition with subtraction → KILLED 4. negated conditional → KILLED 5. negated conditional → KILLED
30	1. Replaced integer addition with subtraction → KILLED 2. negated conditional → KILLED
31	1. Replaced integer addition with subtraction → KILLED 2. Replaced integer subtraction with addition → KILLED 3. negated conditional → KILLED
32	1. Replaced integer addition with subtraction → KILLED
40	1. replaced return value with "" for com/example/StringCompression::compress → KILLED
49	1. changed conditional boundary → KILLED 2. negated conditional → KILLED
55	1. replaced return value with "" for com/example/StringCompression::appendCount → KILLED

Active mutators

- BOOLEAN_FALSE_RETURN
- BOOLEAN_TRUE_RETURN
- CONDITIONALS_BOUNDARY_MUTATOR
- EMPTY_RETURN_VALUES
- INCREMENTS_MUTATOR
- INVERT_NEGS_MUTATOR
- MATH_MUTATOR
- NEGATE_CONDITIONALS_MUTATOR
- NULL_RETURN_VALUES
- PRIMITIVE_RETURN_VALS_MUTATOR
- VOID_METHOD_CALL_MUTATOR

Tests examined

- com.example.StringCompressionTest.testCompressWithSingleCharacterRepeated(com.example.StringCompressionTest) (0 ms)
- com.example.StringCompressionTest.testCompressWithSingleCharacter(com.example.StringCompressionTest) (1 ms)
- com.example.StringCompressionTest.testCompressWithNoRepeatingCharacters(com.example.StringCompressionTest) (1 ms)
- com.example.StringCompressionTest.testCompressWithRepeatingCharacters(com.example.StringCompressionTest) (1 ms)
- com.example.StringCompressionTest.testCompressWithSingleCharacterRepeatedAtEnd(com.example.StringCompressionTest) (0 ms)
- com.example.StringCompressionTest.testCompressWithEmptyString(com.example.StringCompressionTest) (1 ms)

Report generated by [PIT](#) 1.5.0