```
class Solution {
    public int compress(char[] chars) {
        int indexAns = 0, index = 0;
        while(index < chars.length){
            char currentChar = chars[index];
            int count = 0;
            while(index < chars.length && chars[index] == currentChar){
                index++;
                count++;
            }
            chars[indexAns++] = currentChar;
            if (count != 1)
                for(char c : Integer.toString(count).toCharArray())
                    chars[indexAns++] = c;
        }
        return indexAns;
    }
}
```

Given an array of characters, compress it **in-place**.

The length after compression must always be smaller than or equal to the original array.

Every element of the array should be a **character** (not int) of length 1.

After you are done **modifying the input array in-place**, return the new length of the array.

```
Input:
["a","a","b","b","c","c","c"]
Output:
Return 6, and the first 6 characters of the input array
should be: [“a","2","b","2","c","3"]
```

```
Input:
["a"]
Output:
Return 1, and the first 1 characters of the input array
should be: ["a"]
```