

## String Compression

Given an array of characters `chars`, compress it using the following algorithm:

Begin with an empty string `s`. For each group of **consecutive repeating characters** in `chars`:

- If the group's length is 1, append the character to `s`.
- Otherwise, append the character followed by the group's length.

The compressed string `s` **should not be returned separately**, but instead, be stored **in the input character array `chars`**. Note that group lengths that are 10 or longer will be split into multiple characters in `chars`.

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

You must write an algorithm that uses only constant extra space.

### **Example 1:**

**Input:** `chars = ["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"]`

**Explanation:** The groups are "aa", "bb", and "ccc". This compresses to "a2b2c3".

### **Example 2:**

**Input:** `chars = ["a"]`

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

**Explanation:** The only group is "a", which remains uncompressed since it's a single character.

### **Example 3:**

**Input:** `chars = ["a","b","b","b","b","b","b","b","b","b","b","b","b","b"]`

**Output:** Return 4, and the first 4 characters of the input array should be: `["a","b","1","2"]`.

**Explanation:** The groups are "a" and "bbbbbbbbbbbb". This compresses to "ab12".

### **Constraints:**

- $1 \leq \text{chars.length} \leq 2000$
- `chars[i]` is a lowercase English letter, uppercase English letter, digit, or symbol.