75 Days of Code

Day9 Problem no: 443

Companies
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.

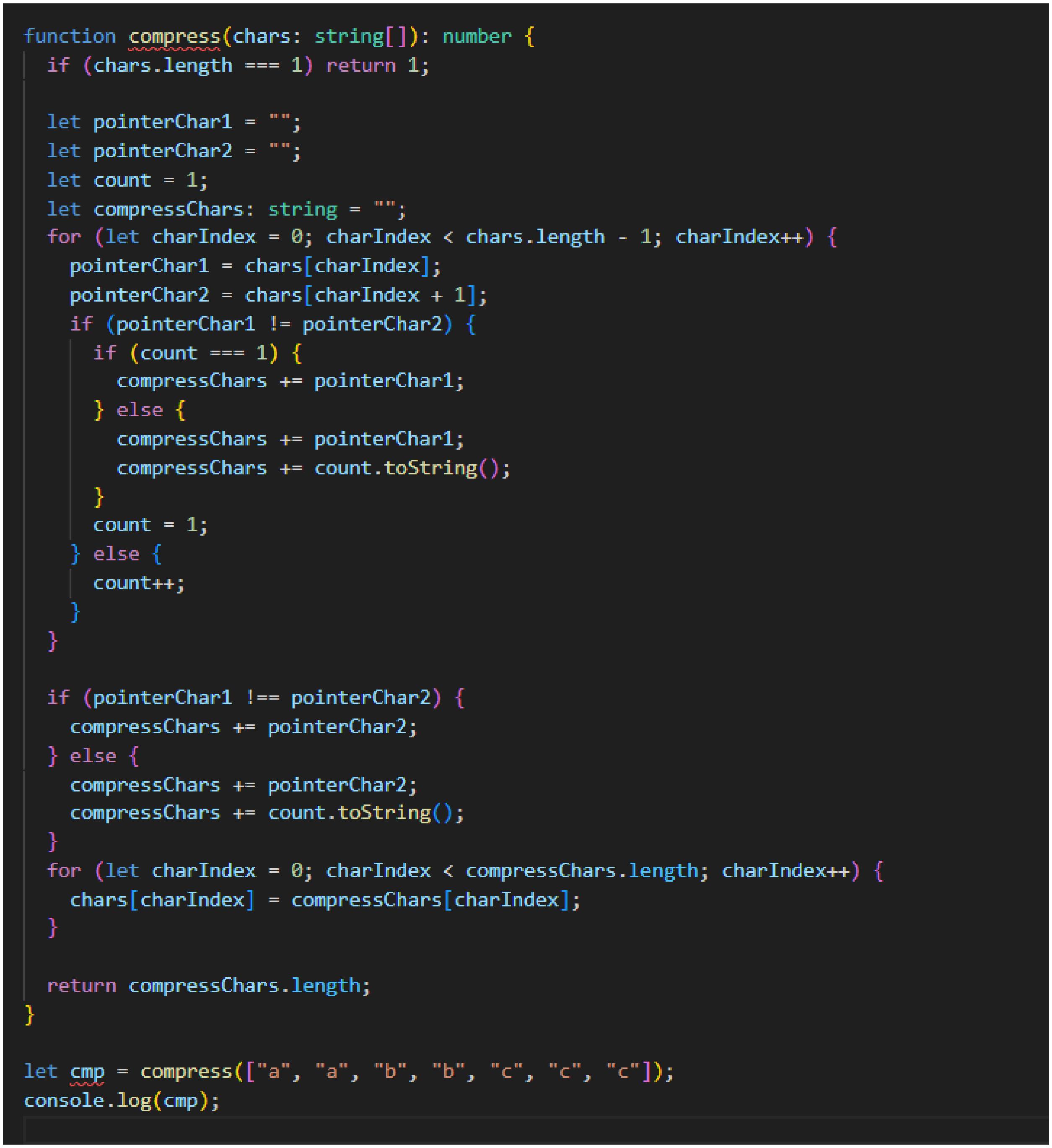
Example 3:

Input: chars = ["a","b","b","b","b","b","b","b","b","b","b","b","b"]
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".

For this problem ,

Shubham Agrahari

1. I am using here two pointers for this problem
2. Iterate through the chars array length up to second last element
3. We initialize pointerOne as the current element and pointer two as the next element and initially the count is 1.
4. On each iteration we will check if current element and next element are not equal , if not then we will check the count if one then only the string is concatenated else both string and count is concatenated . if both pointer are equal than count will increase
5. In loop we can't check the last element , so last element is checked outside the loop and after that we will modify the give input array and return our concatenated string's length;



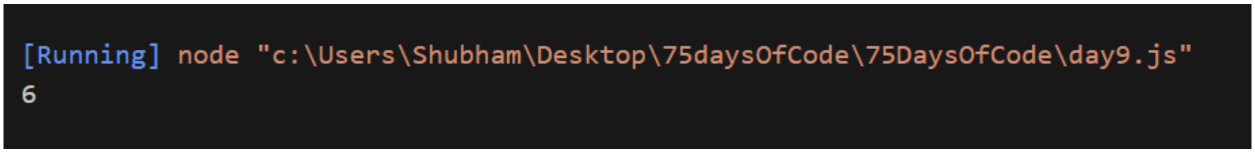
```
function compress(chars: string[]): number {
  if (chars.length === 1) return 1;

  let pointerChar1 = "";
  let pointerChar2 = "";
  let count = 1;
  let compressChars: string = "";
  for (let charIndex = 0; charIndex < chars.length - 1; charIndex++) {
    pointerChar1 = chars[charIndex];
    pointerChar2 = chars[charIndex + 1];
    if (pointerChar1 != pointerChar2) {
      if (count === 1) {
        compressChars += pointerChar1;
      } else {
        compressChars += pointerChar1;
        compressChars += count.toString();
      }
      count = 1;
    } else {
      count++;
    }
  }

  if (pointerChar1 !== pointerChar2) {
    compressChars += pointerChar2;
  } else {
    compressChars += pointerChar2;
    compressChars += count.toString();
  }
  for (let charIndex = 0; charIndex < compressChars.length; charIndex++) {
    chars[charIndex] = compressChars[charIndex];
  }

  return compressChars.length;
}

let cmp = compress(["a", "a", "b", "b", "c", "c", "c"]);
console.log(cmp);
```



```
[Running] node "c:\Users\Shubham\Desktop\75daysOfCode\75DaysOfCode\day9.js"
6
```

Shubham Agrahari