# 5483. Make The String Great

My Submissions (/contest/weekly-contest-201/problems/make-the-string-great/submissions/)

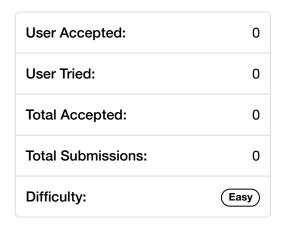
Back to Contest (/contest/weekly-contest-201/)

Given a string s of lower and upper case English letters.

A good string is a string which doesn't have **two adjacent characters** s[i] and s[i + 1] where:

- 0 <= i <= s.length 2
- s[i] is a lower-case letter and s[i + 1] is the same letter but in upper-case or **vice-versa**.

To make the string good, you can choose **two adjacent** characters that make the string bad and remove them. You can keep doing this until the string becomes good.



Return the string after making it good. The answer is guaranteed to be unique under the given constraints.

Notice that an empty string is also good.

## Example 1:

```
Input: s = "leEeetcode"
Output: "leetcode"
Explanation: In the first step, either you choose i = 1 or i = 2, both will result "leEeet
```

#### Example 2:

```
Input: s = "abBAcC"
Output: ""
Explanation: We have many possible scenarios, and all lead to the same answer. For example
"abBAcC" --> "aAcC" --> "cC" --> ""
"abBAcC" --> "abBA" --> "aA" --> ""
```

### Example 3:

```
Input: s = "s"
Output: "s"
```

#### Constraints:

- 1 <= s.length <= 100
- s contains only lower and upper case English letters.



```
1 • /**
     * @param {string} s
 2
     * @return {string}
 3
 4
     */
 5 v const makeGood = function (s) {
        if (s.length == 1) return s;
        let stack = [s[0]];
 7
        for (let i = 1; i < s.length; i++) {
 8 ▼
9 ▼
            if (stack.length == 0){
10
                 stack.push(s[i]);
            } else {
11 ▼
12
                 let end = stack[stack.length - 1];
13 ▼
                 if ((isUpperCase(end) && isLowerCase(s[i]) && isEqual(end, s[i]))
    || (isLowerCase(end) && isUpperCase(s[i]) && isEqual(end, s[i]))) {
                     stack.pop();
14
15
                     continue;
                 }
16
17
                 stack.push(s[i]);
18
            }
19
        }
        return stack.join("");
20
21
    };
22
23
24 \checkmark const isEqual = (s1, s2) => {
25 ▼
        if (s1.toLowerCase() == s2.toLowerCase()) {
26
             return true;
27
28
        return false;
29
    };
30
31 v const isUpperCase = (character) ⇒ {
32 ▼
        if (character == character.toUpperCase()) {
33
             return true;
34
35
        return false;
36
    };
37
38 v const isLowerCase = (character) ⇒ {
39 ▼
        if (character == character.toLowerCase()) {
40
             return true;
41
42
        return false;
43
    };
```

Custom Testcase

Use Example Testcases





# Submission Result: Accepted (/submissions/detail/378183681/) 2

More Details > (/submissions/detail/378183681/)

Share your acceptance!

**∢1** 

Copyright © 2020 LeetCode

Help Center (/support/) | Terms (/terms/) | Privacy Policy (/privacy/)

States (/region/)