

Description | Accepted | Editorial | Solutions | Submissions

1544. Make The String Great

Easy Topics Companies Hint

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 = "leEetcode"`
Output: `"leetcode"`
Explanation: In the first step, either you choose `i = 1` or `i = 2`, both will result `"leEetcode"` to be reduced to `"leetcode"`.

Example 2:
Input: `s = "abBAcC"`
Output: `""`

👍 3.1K 🔄 156 ☆ 📄 ? 10 Online

Code

Python3 🔒 Auto

```
1 class Solution:
2     def makeGood(self, s: str) -> str:
3         stack = []
4         for char in s:
5             if stack and abs(ord(char) - ord(stack[-1])) == 32:
6                 stack.pop()
7             else:
8                 stack.append(char)
9
10        res = ''.join(stack)
11        return res
```

Ln 1, Col 1 | Saved 🔒 Run Submit

☒ Testcase | > Test Result

Accepted Runtime: 0 ms

• Case 1

• Case 2

• Case 3

Input

s =
"leEetcode"

Output