3088. Make String Anti-palindrome

Description

We call a string `s` of **even** length `n` an **anti-palindrome** if for each index `0 <= i < n`, `s[i] != s[n - i - 1]`.

Given a string `s`, your task is to make `s` an **anti-palindrome** by doing **any** number of operations (including zero).

In one operation, you can select two characters from `s` and swap them.

Return *the resulting string. If multiple strings meet the conditions, return the lexicographically smallest one. If it can't be made into an anti-palindrome, return* `"-1"`.

Example 1:

Input: s = "abca"

Output: "aabc"

Explanation:

`"aabc"` is an anti-palindrome string since `s[0] != s[3]` and `s[1] != s[2]`. Also, it is a rearrangement of `"abca"`.

Example 2:

Input: s = "abba"

Output: "aabb"

Explanation:

`"aabb"` is an anti-palindrome string since `s[0] != s[3]` and `s[1] != s[2]`. Also, it is a rearrangement of `"abba"`.

Example 3:

Input: s = "cccd"

Output: "-1"

Explanation:

You can see that no matter how you rearrange the characters of `"cccd"`, either `s[0] == s[3]` or `s[1] == s[2]`. So it can not form an anti-palindrome string.

Constraints:

- `2 <= s.length <= 10^5`
- `s.length % 2 == 0`
- `s` consists only of lowercase English letters.