

Reverse Shuffle Merge



Given a string, A , we define some operations on the string as follows:

- $reverse(A)$ denotes the string obtained by reversing string A . Example: $reverse("abc") = "cba"$
- $shuffle(A)$ denotes any string that's a permutation of string A . Example:
 $shuffle("god") \in ['god', 'gdo', 'ogd', 'odg', 'dgo', 'dog']$
- $merge(A1, A2)$ denotes any string that's obtained by interspersing the two strings $A1$ & $A2$, maintaining the order of characters in both. For example, $A1 = "abc"$ & $A2 = "def"$, one possible result of $merge(A1, A2)$ could be $"abcdef"$, another could be $"abdecf"$, another could be $"adbecf"$ and so on.

Given a string s such that $s \in merge(reverse(A), shuffle(A))$ for some string A , find the lexicographically smallest A .

For example, $s = abab$. We can split it into two strings of ab . The reverse is ba and we need to find a string to shuffle in to get $abab$. The middle two characters match our reverse string, leaving the a and b at the ends. Our shuffle string needs to be ab . Lexicographically $ab < ba$, so our answer is ab .

Function Description

Complete the `reverseShuffleMerge` function in the editor below. It must return the lexicographically smallest string fitting the criteria.

`reverseShuffleMerge` has the following parameter(s):

- s : a string

Input Format

A single line containing the string s .

Constraints

- s contains only lower-case English letters, `ascii[a-z]`
- $1 \leq |s| \leq 10000$

Output Format

Find and return the string which is the lexicographically smallest valid A .

Sample Input 0

```
eggegg
```

Sample Output 0

```
egg
```

Explanation 0

Split "eggegg" into strings of like character counts: "egg", "egg"

reverse("egg") = "gge"
shuffle("egg") can be "egg"
"eggegg" belongs to the merge of ("gge", "egg")

The merge is: **eggegg**.

'egg' < 'gge'

Sample Input 1

abcdefgabcdefg

Sample Output 1

agfedcb

Explanation 1

Split the string into two strings with like characters: *abcde fg* and *abcde fg*.

Reverse *abcde fg* = *gfedcba*

Shuffle *agfedcb* can be *bcde fga*

Merge to *abcdefgabcdefg*

Sample Input 2

aeiouuoiea

Sample Output 2

eaid

Explanation 2

Split the string into groups of like characters: *aeiou*

Reverse *aeiou* = *uoiea*

These merge to *aeiouuoiea*