Lexicographical order is often known as alphabetical order when dealing with strings. A string is greater than another string if it comes later in a lexicographically sorted list.

Given a word, create a new word by swapping some or all of its characters. This new word must meet two criteria:

- It must be greater than the original word

- It must be the smallest word that meets the first condition

Example

w=abcd

The next largest word is: abdc

Function Description

Write a function biggerIsGreater function to create and return the new string meeting the criteria. If it is not possible, return no answer.

biggerIsGreater has the following parameter(s):

string w: a word

Returns

- string: the smallest lexicographically higher string possible or no answer

Constraints

w will contain only letters in the range ascii[a..z].

Samples 1:

Input		**Output**
ab	->	ba
bb	->	null
hefg	->	hegf
dhck	->	dhkc
dkhc	->	hcdk

Test case 1:

ba is the only string which can be made by rearranging ab. It is greater.

Test case 2:

It is not possible to rearrange bb and get a greater string.

Test case 3:

hegf is the next string greater than hefg.

Test case 4:

dhkc is the next string greater than dhck.

Test case 5:

hcdk is the next string greater than dkhc.

Samples 2:

Input		**Output**
lmno	->	lmon
dcba	->	null
dcbb	->	null
abdc	->	acbd
abcd	->	abdc
fedcbabcd	->	fedcbabdc