Carlo's Library (indexing)

Carlo loves organizing his library and has arranged all books in lexicographical order. To enhance their appearance, he wants to insert a new book between two existing titles. The two titles, A and B, are given as strings containing only lowercase letters of the English alphabet.

Your task is to help Carlo find a title C, also consisting of only lowercase letters of the English alphabet for the new book such that:

- C is lexicographically strictly between A and B.
- The length of C is minimized.

Figure 1: Carlo in search of the perfect book title.

Help Carlo find a title C satisfying these requirements, or determine that it doesn't exist.

☞ A string A is lexicographically smaller than a string B if and only if one of the following holds:

- A is a prefix of B, but $A \neq B$.
- in the first position where A and B differ, the string A has a letter that appears earlier in the alphabet than the corresponding letter in B.

☞ Among the attachments of this task you may find a template file `indexing.*` with a sample incomplete implementation.

Input

The input file consists of:

- a line containing string A.
- a line containing string B.

Output

The output file must contain a single line consisting of string C or `-1` if it doesn't exist. If there is more than one correct answer, you can print any.

Constraints

- $1 \leq len(A), len(B) \leq 1\,000\,000$.
- $A < B$ lexicographically.
- A and B contain only lowercase letters of the English alphabet.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

– **Subtask 1** (0 points) Examples.

– **Subtask 2** (15 points) $len(A) = len(B)$.

– **Subtask 3** (15 points) A and B contain only vowels.

– **Subtask 4** (30 points) $len(A), len(B) \leq 1000$.

– **Subtask 5** (40 points) No additional limitations.

Examples

input	output
abc abca	-1
abc def	c
pcn pk	pf
mppxtzmo mppxu	mppxtzz
abc abcaa	abca

Explanation

In the **first sample case** there is no title C that satisfies the constraints.

In the **second sample case** we have that `abc` $<$ `c` $<$ `def`, and there is no shorter title C that satisfies the constraints. Note that $C =$ `b` or $C =$ `d` are strings of length 1 which satisfy the constraints and are, therefore, accepted by the grader.