Czech ACM Student Chapter
Dept. of Computer Science and Engineering, Czech Technical University in Prague
Faculty of Mathematics and Physics, Charles University in Prague
Faculty of Electrical Engineering and Computer Science, Technical University of Ostrava
Faculty of Informatics and Information Technologies, Slovak University of Technology
Faculty of Informatics, Masaryk University
Faculty of Management Science and Informatics, University of Zilina

CTU Open Contest 2008

Careful Declaration

`declare.c`, `declare.C`, `declare.java`, `declare.p`

"Do not panic!" That is not only the inscription on a famous book cover, but it is also a message that all banks need to deliver to their clients during every and each crisis, whatever negligible. If the clients started to panic, it would mean the end for any bank.

To calm down their clients, the banks and governments may issue various declarations stating that there is *really* nothing to worry about. It is very important for such declarations to be consistent, otherwise the people may not believe them.

Imagine that we have two declarations, one suggested by a bank and the other coming from the government. They finally decided to merge those two declarations into a single one and issue it together. (People may think there is some problem if there are too many declarations.) Both the bank and the government insist that the common declaration must contain the whole text of their original declarations.

Your task is to create the joint declaration that will be issued at a press conference tonight. The main condition is that the resulting text must contain all words from the two original proposals, in the same order. Also, we have to keep the declaration as short as possible.

Input Specification

The input contains several test cases, each consisting of two lines of text giving two original declarations. Both lines contain at least one and at most two thousand words separated by a space. Each of the words is formed by lowercase letters of the English alphabet (a–z). Each word will contain at least one such letter and no more than 10 letters. There is a single dot ("`.`") after the last word of each declaration, the dot is also preceded by one space.

There is one additional line with a single dot after the last test case in the input.

Output Specification

Your program has to output a single line of text for each test case. The line should contain a sequence of words separated by a space. The sequence must have the following properties:

1. Both input texts are subsequences of the output text, i.e., all words of each original text appear in the result in the same order, though they may be interleaved by other words.
2. Among all such possible texts, your output must contain the smallest possible number of words.
3. If there are several possibilities of the same length, choose the one that is *lexicographically smallest*.

Output a space and a single dot "`.`" after the last word of the declaration.

Text A is called to be lexicographically smaller than text B, if it “would be first in a regular dictionary”. More exactly, if the first word that is different in both texts is lexicographically smaller in text A than in text B.

Similarly, word C is called to be lexicographically smaller then word D, if (a) for the first letter that is different in both words, the letter in word C comes earlier in the alphabet than the letter in word D, or (b) word C is shorter and it is a prefix of word D.

Sample Input

```
all our banks are sound and safe .
all deposits in our bank are safe do not worry .
everything fine firmly under control .
deposits in bank abc are fine like nothing is .
.
```

Output for Sample Input

```
all deposits in our bank banks are sound and safe do not worry .
deposits everything in bank abc are fine firmly like nothing is under control .
```