671 Spell checker

You, as a member of a development team for a new spell checking program, are to write a module that will check the correctness of given words using a known dictionary of all correct words in all their forms.

If the word is absent in the dictionary then it can be replaced by correct words (from the dictionary) that can be obtained by one of the following operations:

- deleting of one letter from the word;
- replacing of one letter in the word with an arbitrary letter;
- inserting of one arbitrary letter into the word.

Your task is to write the program that will find all possible replacements from the dictionary for every given word.

Input

The first part of the input file contains all words from the dictionary. Each word occupies its own line. This part is finished by the single character '`#`' on a separate line. All words are different. There will be at most 10000 words in the dictionary.

The next part of the file contains all words that are to be checked. Each word occupies its own line. This part is also finished by the single character '`#`' on a separate line. There will be at most 50 words that are to be checked.

All words in the input file (words from the dictionary and words to be checked) consist only of small alphabetic characters and each one contains 15 characters at most.

Output

Write to the output file exactly one line for every checked word in the order of their appearance in the second part of the input file. If the word is correct (i.e. it exists in the dictionary) write the message: "< *checkedword* > `is correct`". If the word is not correct then write this word first, then write the character '`:`' (colon), and after a single space write all its possible replacements, separated by spaces. The replacements should be written in the order of their appearance in the dictionary (in the first part of the input file). If there are no replacements for this word then the line feed should immediately follow the colon.

Sample Input

```
i
is
has
have
be
my
more
contest
me
too
if
award
#
me
aware
m
contest
hav
oo
or
i
fi
mre
#
```

Sample Output

```
me is correct
aware: award
m: i my me
contest is correct
hav: has have
oo: too
or:
i is correct
fi: i
mre: more me
```