Problem A:
Predictive Text Input

Source file: `predict.{c | cpp | java}`
Input file: `predict.in`

Smart mobile devices have dramatically changed the way we interact with machines. One of the smart activities of the smart software in those devices is the ability to predict the word that we are going to type. For example, as I just typed only two letters “mo”, the smart software suggested the words “mobile”, “mother”, and “motion”,. In fact, “mobile” is the word that I was going to type.

In this problem, you are asked to mimic the smart device by writing the smart predictive text software. Whenever the user types a word or a part of a word, the program will suggest the top *three* most frequent words from a dictionary of words such that the typed word is a *prefix* of the dictionary word. Prefix of a word is defined as “A substring of the word obtained by removing zero or more contiguous letters from the end of the word”. For example, *h, he, hel, hell,* and *hello* all are prefixes of *hello*. However, *hello* is not a prefix of *hell* because hello is not a substring of hell. Also, *ell* is not a prefix of *hello* because the letters removed (h and o) are not contiguous in hello and *llo* is also not a prefix of *hello* because although the letters removed (h and e) are contiguous, they are not at the end of the word.

Input

The input may contain several test cases. Each test case contains two parts: a *dictionary* and a set of *queries*. The dictionary starts with an integer N (0 < N < 1000) in one line denoting the number of *words* in the dictionary. The following lines contain the words. The words will be separated by spaces and spread over one or more lines. Each word shall contain only the lowercase letters (‘a’-‘z’) and shall have at least 1 letter and at most 10 letters. The frequency of the *i*-th word in the dictionary is strictly greater than the *i+1*st word. For example, the frequency of the first word > the frequency of the second word > the frequency of the third word, and so on. Therefore, the last word in the dictionary has the lowest frequency, and the first word has the highest frequency. A word in the dictionary will be listed exactly once.

The query part will contain one word per line and each word shall have at least 1 and at most 10 small letters (‘a’-‘z’). The query part will end with a line containing only the word “###”.

The input will be terminated with the value of N = 0.

Output

For each query word *Q*, you are to output one line. First you should output the query word *Q*, followed by a colon and a space and then the top three words, W1, W2, W3, separated by exactly one space. W1 is the highest frequency word in the dictionary such that *Q* is a prefix of W1. Then W2 is the next highest frequency word having Q as a prefix and W3 is the next highest frequency word having Q has a prefix. In case there are only two words in the dictionary that have Q as prefix, you should output those two words. If there is only one word in the dictionary having Q has prefix, you should output only that word. If there is no word in the dictionary that has Q as prefix, then you should output the query word itself.

Sample Input

```
34
gulf programming competition is my favorite it the best
held every year
i come here good luck this
world finals first prize gold medal and then gala medal important
company sponsors commends all winners
g
i
com
pr
gol
sing
###
3
american university dubai
am
d
x
###
0
```

Output for Sample Input

```
g: gulf good gold
i: is it i
com: competition come company
pr: programming prize
gol: gold
sing: sing
am: american
d: dubai
x: x
```