Problem K **Dictionary Attack** **30 points**

One of the simplest forms of cypher is the deranged alphabet substitution cypher. The idea is that we rearrange the letters of the alphabet in some arbitrary (deranged) order, and write the alphabet in its normal order underneath.

```
p o n m t s r q w j u z y x d c b a h g f e l k v i
a b c d e f g h i j k l m n o p q r s t u v w x y z
```

To cypher a message we replace each letter in the message with the one written above. So "`all is well`" will become "`pzz wh ltzz`". Note that there is no rule to prevent some letters being substituted as themselves. In the example above j is cyphered as j.

This is not a strong cypher, and there are many approaches to breaking it. A particular weakness in this case is the fact that the spaces between words have been preserved. This allows us to identify words, and know immediately the length of each word. The method of attack you will implement in this problem is based on this fact. It is a kind of dictionary attack.

Your task is to take a dictionary (a list of all possible words that might be used in a message), and use it to find all possible interpretations of an enciphered message. Be aware that the dictionary might be quite large (one supplied with the test set has 58,111 words).

For example, given a dictionary with just the four words: `aab` `bbc` `abb` `cde`
The phrase: `zzq qqt` can be translated as `aab bbc`
The phrase: `prt` can be translated as `cde`
The phrase: `zzr` can be translated either as `aab` or as `bbc`

Input

Input consists of a series of problem sets. Each problem set consists of a dictionary and a number of cypher texts to translate. The first line of the dictionary holds an integer $0 < N < 60000$, the number of words in the dictionary. It is followed by N lines, each holding one word. A word is entirely in lower case letters with no special characters. No word is longer than 22 letters. Next is a line with a number $C > 0$ of texts to be translated, followed by C lines each holding a text. Texts are single spaced sequences of 'words'. No text is longer than 200 characters. End of input is signalled by a line with the number zero.

Output

For each text to be translated, output a line with 'Problem n' where n = 1, 2, 3, 4… Follow this with the possible translations you have found, one per line, lines being in alphabetic order. It is possible that some texts will have no translations. In that case you output the 'Problem n' header, but have no lines following it. The problem numbering should continue across problem sets.

Sample Input

```
4
aab
bbc
abb
cde
3
zzq qqt
prt
zzr
1
soup
1
abcd
0
```

Sample Output

```
Problem 1
aab bbc
Problem 2
cde
Problem 3
aab
bbc
Problem 4
soup
```