

Problem C

Weaker than Planned

Input: Standard Input
Time Limit: 30 seconds

The committee members of the Kitoshima programming contest had decided to use cryptographic software for their secret communication. They had asked a company, Kodai Software, to develop cryptographic software that employed a cipher based on highly sophisticated mathematics.

According to reports on IT projects, many projects are not delivered on time, on budget, with required features and functions. This applied to this case. Kodai Software failed to implement the cipher by the appointed date of delivery, and asked to use a simpler version that employed a type of substitution cipher for the moment. The committee members got angry and strongly requested to deliver the full specification product, but they unwillingly decided to use this inferior product for the moment.

In what follows, we call the text before encryption, *plaintext*, and the text after encryption, *ciphertext*.

This simple cipher substitutes letters in the plaintext, and its substitution rule is specified with a set of pairs. A pair consists of two letters and is unordered, that is, the order of the letters in the pair does not matter. A pair (A, B) and a pair (B, A) have the same meaning. In one substitution rule, one letter can appear in at most one single pair. When a letter in a pair appears in the plaintext, the letter is replaced with the other letter in the pair. Letters not specified in any pairs are left as they are.

For example, by substituting the plaintext

ABCDEFGHIJKLMN**OP**QRSTUVWXYZ

with the substitution rule $\{(A, Z), (B, Y)\}$ results in the following ciphertext.

ZY**CD**EF~~GH~~IJKLMN**OP**QRSTU**VW**XBA

This may be a big chance for us, because the substitution rule seems weak against cracking. We may be able to know communications between committee members. The mission here is to develop a deciphering program that finds the plaintext messages from given ciphertext messages.

A ciphertext message is composed of one or more ciphertext words. A ciphertext word is generated from a plaintext word with a substitution rule. You have a list of candidate words

containing the words that can appear in the plaintext; no other words may appear. Some words in the list may not actually be used in the plaintext.

There always exists at least one sequence of candidate words from which the given ciphertext is obtained by some substitution rule. There may be cases where it is impossible to uniquely identify the plaintext from a given ciphertext and the list of candidate words.

Input

The input consists of multiple datasets, each of which contains a ciphertext message and a list of candidate words in the following format.

```
n
word1
⋮
wordn
sequence
```

n in the first line is a positive integer, representing the number of candidate words. Each of the next *n* lines represents one of the candidate words. The last line, *sequence*, is a sequence of one or more ciphertext words separated by a single space and terminated with a period.

You may assume the number of characters in each *sequence* is more than 1 and less than or equal to 80 including spaces and the period. The number of candidate words in the list, *n*, does not exceed 20. Only 26 uppercase letters, A to Z, are used in the words and the length of each word is from 1 to 20, inclusive.

A line of a single zero indicates the end of the input.

Output

For each dataset, your program should print the deciphered message in a line. Two adjacent words in an output line should be separated by a single space and the last word should be followed by a single period. When it is impossible to uniquely identify the plaintext, the output line should be a single hyphen followed by a single period.

Sample Input

```
4
A
AND
CAT
DOG
Z XUW ZVX Z YZT.
```

2
AZ
AY
ZA.
2
AA
BB
CC.
16
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
ABCDEFGHIJKLMNO
A B C D E F G H I J K L M N O ABCDEFGHIJKLMNO.
O

Output for the Sample Input

A DOG AND A CAT.
AZ.
-.
A B C D E F G H I J K L M N O ABCDEFGHIJKLMNO.