508 Morse Mismatches

Samuel F. B. Morse is best known for the coding scheme that carries his name. Morse code is still used in international radio communication. The coding of text using Morse code is straightforward. Each character (case is insignificant) is translated to a predefined sequence of *dits* and *dahs* (the elements of Morse code). Dits are represented as periods (".") and dahs are represented as hyphens or minus signs ("–"). Each element is transmitted by sending a signal for some period of time. A dit is rather short, and a dah is, in perfectly formed code, three times as long as a dit. A short silent space appears between elements, with a longer space between characters. A still longer space separates words. This dependence on the spacing and timing of elements means that Morse code operators sometimes do not send perfect code. This results in difficulties for the receiving operator, but frequently the message can be decoded depending on context.

In this problem we consider reception of words in Morse code without spacing between letters. Without the spacing, it is possible for multiple words to be coded the same. For example, if the message "dit dit dit" were received, it could be interpreted as "EEE", "EI", "IE" or "S" based on the coding scheme shown in the sample input. To decide between these multiple interpretations, we assume a particular context by expecting each received word to appear in a dictionary.

For this problem your program will read a table giving the encoding of letters and digits into Morse code, a list of expected words (*context*), and a sequence of words encoded in Morse code (*morse*). These morse words may be flawed. For each *morse* word, your program is to determine the matching word from *context*, if any. If multiple words from *context* match *morse*, or if no word matches perfectly, your program will display the best matching word and a mismatch indicator.

If a single word from *context* matches *morse* perfectly, it will be displayed on a single line, by itself. If multiple *context* words match *morse* perfectly, then select the matching word with the fewest characters. If this still results in an ambiguous match, any of these matches may be displayed. If multiple *context* words exist for a given *morse*, the first matching word will be displayed followed by an exclamation point ("!").

We assume only a simple case of errors in transmission in which elements may be either truncated from the end of a *morse* word or added to the end of a *morse* word. When no perfect matches for *morse* are found, display the word from *context* that matches the longest prefix of *morse*, or has the fewest extra elements beyond those in *morse*. If multiple words in *context* match using these rules, any of these matches may be displayed. Words that do not match perfectly are displayed with a question mark ("?") suffixed.

The input data will only contain cases that fall within the preceding rules.

Input

The Morse code table will appear first and consists of lines each containing an uppercase letter or a digit C, zero or more blanks, and a sequence of no more than six periods and hyphens giving the Morse code for C. Blanks may precede or follow the items on the line. A line containing a single asterisk ("*"), possibly preceded or followed by blanks, terminates the Morse code table. You may assume that there will be Morse code given for every character that appears in the *context* section.

The *context* section appears next, with one word per line, possibly preceded and followed by blanks. Each word in *context* will contain no more than ten characters. No characters other than upper case letters and digits will appear. Thered will be at most 100 *context* words. A line containing only a single asterisk ("*"), possibly preceded or followed by blanks, terminates the *context* section.

The remainder of the input contains morse words separated by blanks or end-of-line characters. A line containing only a single asterisk ("*"), possibly preceded or followed by blanks, terminates the input. No *morse* word will have more than eighty (80) elements.

Output

For each input *morse* word, display the appropriate matching word from *context* followed by an exclamation mark ("!") or question mark ("?") if appropriate. Each word is to appear on a separate line starting in column one.

Sample Input

```
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       --..
0       ------
1       .-----
2       ..---
3       ...--
4       ....-
5       .....
6       -....
```

```
7         --...
8         ---..
9         ----.
*
AN
EARTHQUAKE
EAT
GOD
HATH
IM
READY
TO
WHAT
WROTH
*
.--.....--   .....--....
--.----..   .--.-.----..
.--.....--   .--.
..-.-.-....--.-..-.--.-.
..--   .-...--..-.--
----        ..--
*
```

Sample Output

```
WHAT
HATH
GOD
WROTH?
WHAT
AN
EARTHQUAKE
EAT!
READY
TO
EAT!
```