10584   Text Formalization

One duty Jimmy has at the ACM is to formalize the language and grammar used in texts. Part of this job is expanding contractions and certain acronyms.

A contraction in English is a word or phrase formed by omitting or combining some of the sounds of a longer phrase. For example, "don't" is a contraction for "do not" and "o'clock" comes from "of the clock".

An acronym is a series of letters (or word) formed from the initial letters of a name or from combining parts of a series of words. For example, "ACM" for "Association for Computing Machinery" or "radar" for "radio detecting and ranging".

Your job is to take a list of contractions and acronyms, and expand all contractions and some acronyms in a text.

Input

Input begins with two numbers, $C < 50$ and $A < 50$, indicating respectively the number of contractions and acronyms Jimmy must expand. The next C lines list a contraction and its formal expansion. Following will be a list of A acronyms and their expansions, each on individual lines. Both contractions and acronyms will be presented in the following format:

```
"contraction or acronym" -> "expansion"
```

Since contractions, acronyms and expansions may contain spaces, each will be enclosed in quotation marks and be no more than 80 characters in length. Following the lists of contractions and acronyms will be a series of texts to expand. Each text will consist of lines no longer than 80 characters. No contraction or acronym will be split over multiple lines. A text will be terminated with a line consisting only of the character '#'.

Output

Output each text exactly as input, except for necessary expansions.

All contractions must be fully expanded. Each contraction may appear as listed, entirely uppercase, or capitalized (first letter uppercase, remaining letters as listed). The expansion should follow the same rule; if a contraction is uppercased, the expansion should be uppercased as well. If more than one case applies, choose the earliest matching case in the list: "as listed", "uppercased", and "capitalized".

Since acronyms are useful for understanding and identifying names, only modify the first instance of an acronym in each text. An instance of an acronym must match the case exactly ("acm" is not an

instance of "ACM"). The modification consists of replacing the acronym with the expansion, followed by a space, followed by the acronym in brackets. This allows the reader to connect the acronym with the fully expanded term.

The terminating line of '#' should be printed after each text. If more than one expansion or acronym match can be valid, use the one which starts earlier in the text. If several begin at the same letter, use the one appearing earliest in the input lists. Use the sample below to illustrate the process.

Sample Input

```
3 2
"doesn't" -> "does not"
"isn't" -> "is not"
"can't" -> "cannot"
"ACM" -> "Association for Computing Machinery"
"CS" -> "Computing Science"
The ACM can't solve
all the problems in CS. Though large and having
many resources at its disposal, the ACM doesn't use magic. Magic isn't
part of science, and hence not part of CS. Thank you for your
suggestions.
Signed,

ACM
#
The ACM doesn't like magic.
It's not that the ACM won't use it, it's
just that the ACM doesn't understand magic.
#
```

Sample Output

```
The Association for Computing Machinery (ACM) cannot solve
all the problems in Computing Science (CS). Though large and having
many resources at its disposal, the ACM does not use magic. Magic is not
part of science, and hence not part of CS. Thank you for your
suggestions.
Signed,

ACM
#
The Association for Computing Machinery (ACM) does not like magic.
It's not that the ACM won't use it, it's
just that the ACM does not understand magic.
#
```