The *ACME Publishing Company* has bought thousands of cheap keyboards from ACM (A Computer Manufacturer) and has given one to each of their authors. ACME's writers are not great typists and compose their exquisite novels typing with just one finger of each hand and carefully looking for each one of the keys as they need to press it. As a result, they cannot look at the screen while they are typing. Also, due to the tight deadlines imposed by ACME, the writers cannot revise what they write.

ACM keyboards were amazingly cheap, but that was only because they were slightly defective: each keyboard had a couple of keys incorrectly placed.

For example, one of the keyboards had the "b" and "t" keys each placed where the other one should be. So, if Shakespeare had used it to write *Hamlet* then the phrase "Bo te or nob bo te" would be quite famous today.

Unfortunately, ACME didn't realize the problem until the authors sent several hundreds of novels for publishing, each of them written with a keyboard that had a pair of keys misplaced. Since publishing such gibberish is not acceptable even for ACME standards, they have decided to fix the novels before printing them.

You have to write a program that will read the novels and decide how to fix each of them. For that, it will decide which letters to switch so that the result contains as few **different** wrong words as possible according to a dictionary.

Input

The input format is as follows:

An integer in a single line which says the number of problems to solve. Then, for each problem:

- An integer *nw* in a line of itself denoting the number of words in the dictionary.
- *nw* words, each in a single line. These words will be in lowercase and contain only the characters quoted next: '`abcdefghijklmnopqrstuvwxyz'-`'.
- An integer *nb* in a line of itself telling the number of books to revise using the same dictionary.
- *nb* books. Each book starts with a line containing exactly the string '`.ONCE UPON A TIME.`' and ends with a line containing exactly the string '`.THE END.`'. Books can contain any character between those lines, but the words to be considered can contain only uppercase or lowercase letters from '`a`' to '`z`', hyphens ('`-`') and apostrophes ('`'`'), and cannot start nor end with either a hyphen or an apostrophe.

Output

The output consists of one line for each book showing the two letters (from '`a`' to '`z`') that should be swapped in the text to obtain a text containing the minimum possible number of different words that cannot be found in the dictionary, separated by a space and in alphabetic order. If there are several minimum solutions, only the first one in lexicographic order should be shown.

The case of the letters of a word is not relevant when looking it up in the dictionary. The two letters of the solution must be different.

Sample Input

```
1
4
be
or
to
not
3
.ONCE UPON A TIME.
Bo te or nob bo te.
.THE END.
.ONCE UPON A TIME.
Now pay parbicular abbenbion bo bhis firsb clause, tecause ib's mosb imporbanb.
Bhere's bhe parby of bhe firsb parb shall te known in bhis conbracb as bhe
parby of bhe firsb parb. How do you like bhab, bhab's prebby neab eh?
.THE END.
.ONCE UPON A TIME.
Simple case :-)
.THE END.
```

Sample Output

```
b t
b t
a b
```