The following is an excerpt from an unknown but trusted source:

> "Aoccdrnig to a rscheearch at Cmabrigde Uinervtisy, it deosn't mttaer in waht oredr the ltteers in a wrod are, the olny iprmoetnt tihng is taht the frist and lsat ltteer be at the rghit pclae. The rset can be a toatl mses and you can sitll raed it wouthit porbelm. Tihs is bcuseae the huamn mnid deos not raed ervey lteter by istlef, but the wrod as a wlohe."

In this problem, you will be given a dictionary of words and some text. The letters inside each word of the text have been scrambled, but it is known that the first and last letter are in their original place. Your task is to clean up the text by reordering each word in the original text with its correct version from the dictionary.

Input

The first line of the input contains an integer N indicating the number of test cases ($1 \leq N$). Then, N test cases follow. Each test case is described in exactly two lines: the first line contains the list of words in the dictionary, sorted in ascending lexicographical order, and the second line contains the list of words to be cleaned up. Words in both lists are separated by single blanks and are made of just English lowercase letters. You can assume that each list is non-empty and has at most 200 characters, and that the dictionary does not contain duplicates.

Output

For each test case, output a single line with the clean version of the text: each word w in the input text should be replaced by a word d in the dictionary such that d can be converted to w by reordering letters except for the first and last ones. If there is more than one word d in the dictionary that could be used for w, replace it with the one that comes first in lexicographical order; if there are no words in the dictionary for w, then take $d = w$.

Sample Input

```
2
make me programming rich will
pagmrnmorig will mkae me rcih
dreaming drinaemg yeah
yaeh right keep drinaemg
```

Sample Output

```
programming will make me rich
yeah right keep dreaming
```