Problem 4
Ballot Sort

Since candidates whose names appear earlier on a ballot tend to get more votes than candidates whose names appear later, the State of Confusion requires that the names of the candidates for an office appear in different orders in different electoral districts. The intent is to even out, over all of the districts, the advantage or disadvantage of name order for all of the candidates. To do this, the names of the candidates for each office in each district are sorted according to an arbitrary "alphabetic order" that is chosen for that office and district by rules designed to ensure fairness. We won't go into the details of these rules, since you are not being asked to choose the orderings for the alphabet, just to sort the candidate's names according to an ordering that's already been chosen.

Your program must be able to handle the alphabetic ordering and list of candidates for a single office in a single district. To handle different offices or different districts, it will simply be rerun with the appropriate input. The first line of input will be an "alphabet" line of 26 characters, representing the sort order of the alphabet for the office and district being handled. The names of the candidates will follow this line, one line per candidate. The list of candidate's names will be terminated by end-of-file.

There will be no more than 1024 candidates for any office, and no candidate's name will be longer than 255 characters. All characters in the input are either upper case alphabetic (A-Z), or the space character.

The "alphabet" line contains just the letters from "A" to "Z", but in the order they should be used in the sort. A letter earlier in the line should sort before a letter later in the line. The space character doesn't appear in the "alphabet" line, since it always sorts before any of the alphabetic characters.

Each line in the list of candidates contains the name of one candidate, with no leading or trailing space characters, and a single space character between each of the parts of the name. The parts of the name have been ordered as they should be sorted, so there is no need to reorder the parts of a name on a line to ensure that, for example, the surname has precedence in the sort. If it does, the surname will be first on the input line.

The output of the program is the list of candidates from the input, in the sort order specified by the "alphabet" line of the input. The program must reorder the lines, if necessary, but it must not alter their contents. Since the output will contain all of the lines in the input, except the "alphabet" line, it will be exactly one line shorter than the input.

Sample Input

```
XWYPZRNQMKTJOIGHDBCLAEVUFS
ADAMS JOHN QUINCY
ADAMS JOHN
FRANKLIN BENJAMIN
HANCOCK JOHN
JEFFERSON THOMAS
LEE RICHARD HENRY
LEE ROBERT E
LINCOLN ABRAHAM
```

Output for the Sample Input

```
JEFFERSON THOMAS
HANCOCK JOHN
LINCOLN ABRAHAM
LEE ROBERT E
LEE RICHARD HENRY
ADAMS JOHN
ADAMS JOHN QUINCY
FRANKLIN BENJAMIN
```