Problem 3 | **5 Points**

Variable Key String Sort

Program Name: sort.cpp   Input File: sort.dat

Although most string sorts are done based on the order assigned to characters in the English language, it is possible that other languages or codes assign a different order to the characters. For this problem, you are to write a program that will sort up to 30 strings based on a different ordering of the alphabetic characters.

Your program is given the new ordering of the characters in the alphabet as a single string of the characters in the order (ascending -- from first to last) being prescribed. Your program is then given up to 30 strings which are up to 10 characters in length each. Your program is required to sort the strings in ascending order.

For example, if your program is given the following alphabetic ordering
QWERTYUIOPASDFGHJKLZXCVBNM

Then it would sort the words “HONEYBEE”, “HONEY”, “SPIDER”, “SPITE”, “WEB”, and “RAIDERS” as follows.
WEB
RAIDERS
SPITE
SPIDER
HONEY
HONEYBEE

Note that the words “HONEY” and “HONEYBEE” are sorted such that two words which are equal up to the length of the shorter word will be ordered such that the shorter word will be considered to have the lesser value of the two.

Input

The first line of input to your program consists of exactly 26 alphabetic characters. The characters on this line define the proper ordering of the character set you should use as a basis for sorting. All characters of the English alphabet appear exactly once and are in upper case. The remaining lines in the input file each contain exactly one word which should be sorted according to the ordering of the first line of the file. The words to be sorted are all in upper case and are between 1 and 10 alphabetic characters in length. There are between 1 and 30 words to be sorted and there are no embedded lines or blanks anywhere in the input file.

Output

Output from your file will consist of the words on the lines after the first line of input in the input file. These words should be sorted in ascending order per the rules described above. Your output should not contain any extraneous output including blank lines or embedded spaces.

Example: Input file

```
LAKSJDHFGPQOWIEURYTMZNXBCV
LAKESIDE
RED
EDUCATION
I
FORTUNE
SORTING
VLASIC
VLADIMIR
SCHOLZBOT
MORK
GROOVY
FONZIE
ORSON
ORK
COMPULSORY
```

Output to screen

```
LAKESIDE
SORTING
SCHOLZBOT
FORTUNE
FONZIE
GROOVY
ORK
ORSON
I
EDUCATION
RED
MORK
COMPULSORY
VLASIC
VLADIMIR
```