In the not-so-distance future, space engineers can invent a new technology for traveling through space and name it as warp-drive. This "warp-drive" can make a spaceship travel faster than light speed. It works by bending an amount of distance in space and make a ship travel through that bended space in a single "hop". And since the time spent for each hop is equal, the total traveling time for any spaceship (that is equipped with this warp-drive) depends on the number of hops it make. Moreover, these engineers notice that distance of a hop depends on some kind of force fields in the space where that hop is operated. To travel from a beginning to an ending point, a spaceship may have to pass through many force fields, which may make that spaceship hop so many times.

And from their experiments, they found some facts as follow.

- A number of force field types is quite small.
  - So one English character can be used to name each of these force filed types.
- A spaceship can pass through every single force field in a single hop.
- A spaceship can pass through a certain sequences of two or more force fields in a single hop. And these certain sequences are kept as rules of single-hop-able sequences.
  - An example of list of these rules is shown as in the following table.

Rules of Single-hop-able Sequences
`ab`
`abcd`
`abd`
`bde`

  - In this example, there are 4 rules, which means that a spaceship can pass through each of them using a single hop. The first one is "`ab`", which means that a spaceship can pass through '`a`' and '`b`' force field in a single hop. The second one is "`abcd`" which means that it can pass through '`a`', '`b`', '`c`' and '`d`' in one hop.
  - Please be notified that there is no "`abc`" sequence/rule in this example, so even though the ship can pass through "`abcd`" sequence, it is unable to pass through "`abc`" (in a single hop). It has to make 2 hops, the first hop is to pass through "`ab`" sequence and the second hop is to pass through "`c`" force field.

Goal:

Suppose that you are an engineer on a battle spaceship. Your duty is to drive your spaceship through space as fast as possible. Your spaceship has a probe device that can identify a sequence of force fields along the path to your destination. In order to accomplish your duty, you have to build a computer program to find the minimum hop based on rules and any given sequences of force fields.

Input

Input is standard inputs which contain 2 parts of data which are separated by a blank line.

The first part is a set of rules of single-hop-able sequences. The number of rules is between 1 and 10,000.

- Each line in this first part contains one rule.
- Each rule is a (sub)sequence of force fields specified by a string of alphabets (`a-z`, `A-Z`).
- The order in every (sub)sequence is from left to right.
- The size of any (sub)sequence is between 2 and 20.

The second part is a set of force fields along path that a spaceship has to pass through. The number of sequence is between 1 and 200.

- Each line in this second part contains one sequence of force field in one path.
- Each sequence is specified by a string of alphabets (`a-z`, `A-Z`).
- The order in every sequence is from left to right.
- The size of any sequence is between 2 and 500.

The blank line after the second part is the termination of the input.

Output

For each sequence in the second part of input, write 2 parts of output as follows

- In the first line, write the total number of solutions followed by a space and the number of minimum hops.
- In the following lines, write each of the solutions in each line. Each solution contains a sequence of force fields as given in the input, where a space is inserted between each hop sub sequences. If there are two or more solutions, they must be sorted by ascending and lexicographical (ASCII) order.

More Explanations: There are 5 rules and 5 input sequences in this sample input.

In the first sample output, there is 1 solution with 2 hops. The first hop is from the first rule.

In the second output, there is 1 solution with 1 hop using the second rule for the whole sequence.

In the third output, there are 2 solutions with 2 hops. Its first solution is from the 4-th rule and the second solution is from the 3-rd rule.

In the fourth output, there are 4 solutions with 4 hops. The first solution is applied using the 4-th rule twice. The second and third is applied using the 3-rd and 4-th rules in different position. The last solution is applied using the 3-rd rules twice

The last output is quite similar to the fourth one, but it is also applied with the fifth rule in the middle of the sequence.

Sample Input

```
ab
abcd
abd
bde
CgeF

abc
abcd
abde
abdeabde
abdeCgeFabde
```

Sample Output

```
1 2
ab c
1 1
abcd
2 2
a bde
abd e
4 4
a bde a bde
a bde abd e
abd e a bde
abd e abd e
4 5
a bde CgeF a bde
a bde CgeF abd e
abd e CgeF a bde
abd e CgeF abd e
```