

# Problem TEAMWORK: Teamwork

You are participating a programming contest with Alice and Bob in your team. You just solved a problem and you know, that the last remaining problem is solvable, but has an input, which is awful to parse and you absolutely don't want to do this. So you decide to go to the buffet, while Alice or Bob should start programming. When you come back you see Bob typing and you think your plan works. But then Alice tells you that Bob is programming the algorithm part of the solution. She is going to write some wrapper for scanf/printf. So you just have to write some input parsing. They have already reformulated the problem statement for you:

## Input

The input is a road network. But for some stupid reason the information is hidden in some text. First all involved cities are given in a text, in which every word with a starting uppercase letter, which doesn't follow one of [.!?] is one involved city. A city name contains only alphanumeric characters. The first word in the text is never a city. Cities may not occur more than once. This text ends with a single empty line.

Then a text with the connections follows. The connections are also embedded in an arbitrary text, where cities and distances occur. All numbers in the text are distances and represent a connection from the previous mentioned City to the next mentioned city. One occurrence of a city may belong to more than one connection. There is at least one city between two numbers. There are no more than 100 cities in a test case and at most 256 characters in a line.

## Output

You should output the road network as so called adjacency list. Every city gets a ascending number  $i_c$  in the order of first occurrence in the first text, starting with 1. You have to output the number of cities in the first line. And for every city one line with it's outgoing connections. First output the number of outgoing connections. For every connection output the length, formatted like the input and the number of the destination city separated by one space. The connections are separated by spaces, too. Cities and Connections are sorted by the occurrence in the text.

### Sample Input 1

```
The most important cities for me are Erlangen, Nuernberg and Tennenlohe.  
Not so important for me is Munich.
```

```
Starting from Erlangen Suedgelaende I need 15 minutes to reach Tennenlohe with the bike.  
But I need 20 minutes back to Erlangen. To travel to Nuernberg I whould prefer the train.  
With the Train you need 30.5 minutes from Erlangen.
```

### Sample Output 1

```
4  
1 15 3  
1 30.5 1  
1 20 1  
0
```