Example

As you may have probably noticed a problem statement in a programming contest consists of several sections. The most important section is, of course, the "Example" section. Some seasoned contestants even start reading the problem statement from the examples. And, unfortunately, the least read section is the problem description section. It is quite disappointing for the problem authors because they feel that their writing skills are largely wasted.

So the authors decided to describe examples in the same language, as the rest of the problem statement using the following rules.

- Natural integer numbers shall be written in plain English. The numbers shall be less than one hundred. The designator number shall be written before numbers, except when the corresponding number is used as a **repetition factor**. For example, *number zero*, *number sixteen*, or *number sixty one*.
- Sequences of characters (strings) shall be written either in quotes, or in apostrophes, for example *"John's pen"*, or *'5" tall'*. Note, that ' may be used in strings enclosed in " and vice versa. The designator string shall be written before strings, for example *string 'Hello'*.
- The designator *space* denotes one space character.
- A number, string or space may be prefixed with a **repetition factor**. The repetition factor is a number greater than one. The designator after the repetition factor is written in plural form. For example, *four numbers five*, or *six strings 'A'*. If the repetition factor is used for numbers, the numbers are separated with one space character. So, the former example means *5 5 5 5*, but the latter example *AAAAAA*.
- Let the numbers, strings and spaces with possible repetition factors be called **fragments**. Fragments may be organized into **sequences** using the *followed by* copulative. For example, *number five followed by number six*. One implicit space character is assumed between numbers, a number and a string, and a string and a number so the example above means *5 6*.
- An example is described line by line. The first line is always described as *The first line...*. The following lines are described either as *The next line...* or as *The next # lines...*, where *#* is a number greater than one. Empty lines are described as *is empty* or *are empty*. Other lines are described as *contains* or *contain* followed by sequences. The first letter of a sentence is capitalized. The sentence is terminated by a full stop (**.**). The full stop is not separated by space from the preceding word, but is separated by at least one space from the next word.

Input

The input contains a free-flow text describing an example. Words are separated by an arbitrary number of spaces and newlines. There are no whitespace characters after the last full stop. The total size of the input will be no greater than 1 MB.

Output

The output should contain the decoded example. The total size of the output shall be no greater than 1 MB.

Example

Input:
The first line contains four numbers twenty eight. The next line is empty. The next two lines contain six strings '-'.
The next line contains number four followed by number seventy seven followed by string 'meat' followed by three strings "!".

Output:
```
28 28 28 28

------
------
4 77 meat!!!
```