



## Perfect rhyme

*A perfect rhyme is not a crime,  
it is something that exceeds time,  
a bit of science, a piece of art,  
soft as a pillow, sharp as a dart.*

Everyone tried it, but only few chosen ones succeeded. It is a hard task with an unclear path, but a famous end – should you reach it. Many compare it to finding the Holy Grail, or even to finding Waldo. The task is to find a perfect rhyme.

### Problem specification

Given is a wordlist  $L$ , and a word  $w$ . Your task is to find a word in  $L$  that forms a perfect rhyme with  $w$ . This word  $u$  is uniquely determined by these properties:

- It is in  $L$ .
- It is different from  $w$ .
- Their common suffix is as long as possible.
- Out of all words that satisfy the previous points,  $u$  is the lexicographically smallest one.

### Notes

A prefix of a word is any string that can be obtained by repeatedly deleting the last letter of the word. Similarly, a suffix of a word is any string that can be obtained by repeatedly deleting the first letter of the word.

For example, consider the word **different**.

This word is both its own prefix and suffix. Its longest other prefix is **differen**, and its longest other suffix is **ifferent**. The string **rent** is its yet another, even shorter suffix. The strings **eent** and **iffe** are neither prefixes nor suffixes of the word **different**.

Let  $u$  and  $v$  be two different words. We say that  $u$  is lexicographically smaller than  $v$  if either  $u$  is a prefix of  $v$ , or if  $i$  is the first position where they differ, and the  $i$ -th letter of  $u$  is earlier in the alphabet than the  $i$ -th letter of  $v$ .

For example, **dog** is smaller than **dogs**, which is smaller than **dragon** (because **o** is less than **r**).

### Input specification

The input file consists of two parts. The first part contains the wordlist  $L$ , one word per line. Each word consists of lowercase English letters only, and no two words are equal.

The first part is terminated by an empty line.

The second part follows, with one query word  $w$  per line.

### Output specification

For each query in the input file output a single line with its perfect rhyme. The output must be in lowercase.

**Example**

input

```
perfect  
rhyme  
crime  
time
```

```
crime  
rhyme
```

output

```
time  
crime
```

*In the second test case, there were two candidates that had an equally long common suffix (crime and time), the lexicographically smaller one was chosen.*



## Quiz

In ancient Rome people used to say: “Historia magistra vitae.” (History is the teacher of lives.) Knowing the history helps us to correctly decide in present, to avoid dangerous paths and sometimes it can even give us advantage against our opponents.

By taking part in this practice session, you’ll gain such an advantage against teams that skip it – by getting to know the history of IPSC. Towards this end we prepared a small quiz for you.

### Input specification

In the input file, each paragraph represents one quiz question. The paragraphs are separated by blank lines

### Output specification

For each question, output a single line with the correct answer.

There are two types of answers:

- Numeric answers: The line must contain exactly one decimal number.
- Verbal answers: The line must contain one string of UPPERCASE English letters.

You have to answer all questions correctly in order to solve the given input. Note that exact spelling matters. Read carefully, good luck, and have fun while learning about the famous history of IPSC!

### Example

input	output
How many letters does "IPSC" have? The first letter in Greek alphabet?	4 ALPHA



## Railroad map

The Slovak national railroad company has recently built new tracks. They want to update their railroad map according to these changes. But they want the map to be as simple as possible. So they decided to remove from the map all the stations that have exactly two other direct connections to other stations (i.e., a single railroad passing through the station).

### Problem specification

You will be given the complete map of Slovak railroads. It consists of railway stations numbered from 1 to  $N$ , and railroad segments between some pairs of these stations. For each railroad segment we are given its length.

Your task is to remove all such stations that are directly connected with exactly two other stations, and output the new map. The new map must contain correct distances between the remaining stations.

### Input specification

The first line of the input file contains an integer  $T$  specifying the number of test cases. Each test case is preceded by a blank line.

Each test case begins with a line with two integers  $N$  and  $M$ . The number  $N$  denotes the number of stations and  $M$  is the number of railroad segments.  $M$  lines follow, each with 3 integers  $a$ ,  $b$ , and  $c$  ( $1 \leq a, b \leq N$ ) specifying that there is a railroad segment of length  $c$  connecting stations  $a$  and  $b$ .

You can assume that in each test case there is a path between every two stations. Also, there will always be at least 2 stations that are not directly connected to exactly two other stations.

### Output specification

For each test case, the output shall consist of multiple lines. The first line shall contain a positive integer  $K$  – the number of railroads on the simplified map. Each of the next  $K$  lines shall contain three integers  $a$ ,  $b$ , and  $c$  stating that there is a railroad of length  $c$  between stations  $a$  and  $b$  on the simplified map.

Print a blank line between outputs for different test cases.

### Example

input	output
<pre>2  3 2 1 2 1 2 3 1  4 3 1 2 1 2 3 2 3 4 3 4 2 1</pre>	<pre>1 1 3 2  2 1 2 1 2 2 6</pre>

*In the first case we removed station 2 because it had exactly 2 direct connections.*

*In the second case we removed stations 3 and 4. We see that there is now a railroad from station 2 back to itself.*