The word "search engine" may not be strange to you. Generally speaking, a search engine searches the web pages available in the Internet, extracts and organizes the information and responds to users' queries with the most relevant pages. World famous search engines, like GOOGLE, have become very important tools for us to use when we visit the web. Such conversations are now common in our daily life:

> "What does the word like $* * * * **$ mean?"
> "Um… I am not sure, just google it."

In this problem, you are required to construct a small search engine. Sounds impossible, does it? Don't worry, here is a tutorial teaching you how to organize large collection of texts efficiently and respond to queries quickly step by step. You don't need to worry about the fetching process of web pages, all the web pages are provided to you in text format as the input data. Besides, a lot of queries are also provided to validate your system.

Modern search engines use a technique called *inversion* for dealing with very large sets of documents. The method relies on the construction of a data structure, called an *inverted index*, which associates *terms* (words) to their occurrences in the collection of documents. The set of terms of interest is called the *vocabulary*, denoted as V. In its simplest form, an inverted index is a dictionary where each search key is a term $\omega \in V$. The associated value $b(\omega)$ is a pointer to an additional intermediate data structure, called a bucket. The bucket associated with a certain term ω is essentially a list of pointers marking all the occurrences of ω in the text collection. Each entry in each bucket simply consists of the *document identifier (DID)*, the ordinal number of the document within the collection and the ordinal line number of the term's occurrence within the document.

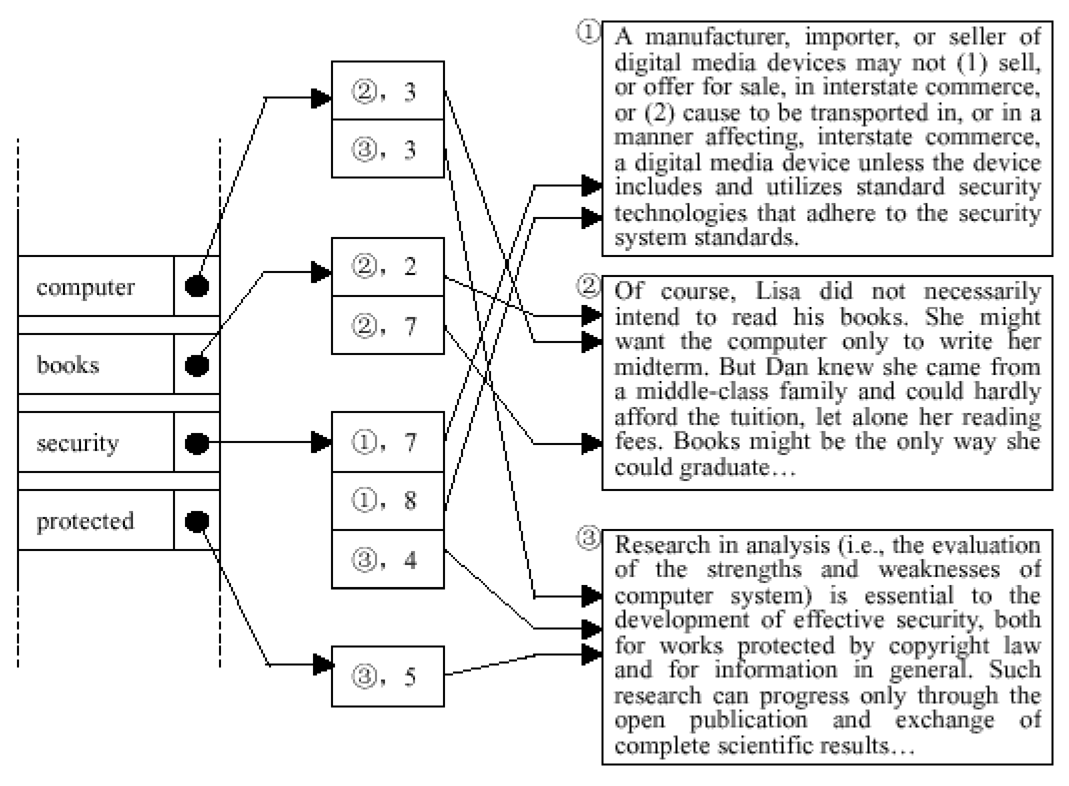
Let's take Figure-1 for an example, which describes the general structure. Assuming that we only have three documents to handle, shown at the right part in Figure-1; first we need to tokenize the text for words (blank, punctuations and other non-alphabetic characters are used to separate words) and construct our vocabulary from terms occurring in the documents. For simplicity, we don't need to consider any phrases, only a single word as a term. Furthermore, the terms are case-insensitive (e.g. we consider "book" and "Book" to be the same term) and we don't consider any morphological variants (e.g. we consider "books" and "book", "protected" and "protect" to be different terms) and hyphenated words (e.g. "middle-class" is not a single term, but separated into 2 terms "middle" and "class" by the hyphen). The vocabulary is shown at the left part in Figure-1. Each term of the vocabulary has a pointer to its bucket. The collection of the buckets is shown at the middle part in Figure-1. Each item in a bucket records the DID of the term's occurrence.

After constructing the whole inverted index structure, we may apply it to the queries. The query is in any of the following formats:

term
term `AND` *term*
term `OR` *term*
`NOT` *term*

A single term can be combined by Boolean operators: '`AND`', '`OR`' and '`NOT`' ('*term*1 `AND` *term*2' means to query the documents including *term*1 and *term*2; '*term*1 `OR` *term*2' means to query the documents including *term*1 or *term*2; '`NOT` *term*1' means to query the documents not including *term*1). Terms are single words as defined above. You are guaranteed that no non-alphabetic characters appear in a term, and all the terms are in lowercase. Furthermore, some meaningless stop words (common words such as articles, prepositions, and adverbs, specified to be "the, a, to, and, or, not" in our problem) will not appear in the query, either.

For each query, the engine based on the constructed inverted index searches the term in the vocabulary, compares the terms' bucket information, and then gives the result to user. Now can you construct the engine?



A manufacturer, importer, or seller of digital media devices may not (1) sell, or offer for sale, in interstate commerce, or (2) cause to be transported in, or in a manner affecting, interstate commerce, a digital media device unless the device includes and utilizes standard security technologies that adhere to the security system standards.

Of course, Lisa did not necessarily intend to read his books. She might want the computer only to write her midterm. But Dan knew she came from a middle-class family and could hardly afford the tuition, let alone her reading fees. Books might be the only way she could graduate…

Research in analysis (i.e., the evaluation of the strengths and weaknesses of computer system) is essential to the development of effective security, both for works protected by copyright law and for information in general. Such research can progress only through the open publication and exchange of complete scientific results…

Figure-1. Inverted Index

Input

The input starts with integer N ($0 < N < 100$) representing N documents provided. Then the next N sections are N documents. Each section contains the document content and ends with a single line of ten asterisks.

```
**********
```

You may assume that each line contains no more than 80 characters and the total number of lines in the N documents will not exceed 1500.

Next, integer M ($0 < M \leq 50000$) is given representing the number of queries, followed by M lines, each query in one line. All the queries correspond to the format described above.

Output

For each query, you need to find the document satisfying the query, and output just the lines within the documents that include the search term (For a '`NOT`' query, you need to output the whole document). You should print the lines in the same order as they appear in the input. Separate different documents with a single line of 10 dashes.

```
----------
```

If no documents matching the query are found, just output a single line: '`Sorry, I found nothing.`'. The output of each query ends with a single line of 10 equal signs.

```
==========
```

Sample Input

```
4
A manufacturer, importer, or seller of
digital media devices may not (1) sell,
or offer for sale, in interstate commerce,
or (2) cause to be transported in, or in a
manner affecting, interstate commerce,
a digital media device unless the device
includes and utilizes standard security
technologies that adhere to the security
system standards.
**********
Of course, Lisa did not necessarily
intend to read his books. She might
want the computer only to write her
midterm. But Dan knew she came from
a middle-class family and could hardly
afford the tuition, let alone her reading
fees. Books might be the only way she
could graduate
**********
Research in analysis (i.e., the evaluation
of the strengths and weaknesses of
computer system) is essential to the
development of effective security, both
for works protected by copyright law
and for information in general. Such
research can progress only through the
open publication and exchange of
complete scientific results
**********
I am very very very happy!
What about you?
**********
6
computer
books AND computer
books OR protected
NOT security
very
slick
```

Sample Output

```
want the computer only to write her
---------
computer system) is essential to the
==========
intend to read his books. She might
want the computer only to write her
fees. Books might be the only way she
==========
intend to read his books. She might
fees. Books might be the only way she
---------
for works protected by copyright law
==========
Of course, Lisa did not necessarily
intend to read his books. She might
want the computer only to write her
midterm. But Dan knew she came from
a middle-class family and could hardly
afford the tuition, let alone her reading
fees. Books might be the only way she
could graduate
---------
I am very very very happy!
What about you?
==========
I am very very very happy!
==========
Sorry, I found nothing.
==========
```