Problem I - I Don't Like Sand

Time limit: 6 seconds

"I don't like sand. It's coarse, and rough, and irritating, and it gets everywhere" – Anakin Skywalker

There are many things that people don't like. There also things that people do like.

When Lucca is planning his movie night sessions, he wants to make sure that all of his M friends are happy. Since he loves democracy, he wanted the people to decide what donuts to order from his shop before the movies start:

"If we order cinnamon, don't order custard."

"Order cinnamon and glazed!"

"I'll be happy if we have vanilla or blueberry."

To simplify things, Lucca asked everybody at practice to structure requests into simple requests and complex requests.

Simple requests are of one of the following forms:

- `name` is a simple request. This request is fulfilled if, and only if, Lucca orders donut `name`.
- `-name` is a simple request. This request is fulfilled if, and only if, Lucca doesn't order donut `name`.

Complex requests are of one of the following forms:

- `(x & y)` where `x` and `y` are simple requests. To fulfill this request, Lucca needs to fulfill both request `x` and request `y`.
- `(x | y)` where `x` and `y` are simple requests. To fulfill this request, Lucca needs to fulfill either request `x` or request `y`.
- `(x => y)` where `x` and `y` are simple requests. To fulfill this request, Lucca needs to fulfill request `y` if he fulfills `x`. In other words, Lucca cannot fulfill request `x` without also fulfilling request `y`.

Lucca gets all of his friends to submit exactly one complex request, and requires all of them to submit their requests at the same time. Each of his friends will be happy if their complex request is fulfilled. Lucca had an emergency at his donut shop, and asked you to figure out a way to order donut so that **no one is unhappy** (or let him know if it is not possible).

Input

The first line contains a single integer, T specifying the number of test cases.

Each test case begins with a single integer, N ($1 \leq N \leq 25,000$), on its own line denoting the number of different types of donut that can be ordered. This is followed by N lines with the name of a different in each. It is guaranteed that the total length of all donut names consists of less than 10^6 characters and each donut name is less than 10 characters in length. Characters consists only of lower case letters `a-z`.

Then there will be M ($1 \leq M \leq 150,000$), on its own line denoting the number of different requests that Lucca received. In each of the following M lines is one complex request.

Output

For each test case output k the number of donuts Lucca needs to order. Then on the following k lines, output the names of the k donuts that satisfies all the requests in any order. You can print any valid solution. If there is no valid solution, output `-1`.

Sample Input

```
3
1
vanilla
1
(vanilla & -vanilla)
2
redvelvet
vanilla
2
(-vanilla | vanilla)
(-redvelvet | -redvelvet)
2
redvelvet
banana
2
(banana => redvelvet)
(redvelvet => banana)
```

Sample Output

```
-1
0
2
banana
redvelvet
```