# Problem F - Fulfilling Requests

Daniel has decided to take requests from the people at ACM practice for pizza. For the sake of variety, he orders at most one of a given type of pizza. People start clamouring to make requests:

"*If we order pepperoni, don't order pineapple.*"

"*Order pepperoni and cheese please!*"

"*I'll be happy if we have cheese or pineapple.*"

To simplify things, Daniel 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 Daniel orders pizza `name`.
- `-name` is also a simple request. This request is fulfilled if and only if Daniel doesn't order pizza `name`.

Complex requests of the following forms:

- ($x$ `&` $y$) where $x$ and $y$ are simple requests. To fulfill this request Daniel needs to fulfill both request $x$ and request $y$.
- ($x$ `|` $y$) where $x$ and $y$ are simple requests. To fulfill this request Daniel needs to fulfill at least one of request $x$ and request $y$.
- ($x$ `=>` $y$) where $x$ and $y$ are simple requests. To fulfill this request Daniel neeeds to fulfill request $y$ if Daniel is planning on fulfilling request $x$.

Daniel doesn't want to deal with all the different types of requests at once, so he gets everybody to submit requests of exactly one of the complex types and requires that the requests are consistent. Help Daniel figure out what pizzas to order, or determine that it is impossible to order pizzas to satisfy everyone's request (so Daniel will just order the pizzas he likes to eat).

## Input

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

Each test case begins with a single integer, $1 \leq n \leq 25000$, on its own line denoting the number of different types of pizza that can be ordered. This is followed by $n$ lines with the name of a different pizza on each line. It is guarenteed that the total length of all pizza names consists of less than $10^6$ characters and each pizza name is less than 10 characters in length. Characters consists only of lower case letters `a-z`.

Then there will be $1 \leq m \leq 150000$, on its own line denoting the number of different requests that Daniel recieved. On the following $m$ lines is one request per line. It is guarenteed that all the requests are of exactly one complex request type.

## Output

For each test case, if there is no solution, output `-1`, otherwise first output $k$ the number of pizzas Daniel needs to order, then on the following $k$ lines, output the names of the $k$ pizzas that satisfies all the requests in any order. If there is a solution, it may not be unique.

## Sample Input

```
3
1
taco
1
(taco & -taco)
2
pepperoni
taco
2
(-taco | taco)
(-pepperoni | -pepperoni)
2
pepperoni
banana
2
(banana => pepperoni)
(pepperoni => banana)
```

## Sample Output

```
-1
0
2
banana
pepperoni
```