A – Accountant notes

You had an accountant, who recorded all your expenses. Each day you sent her notes on expenses and she appended them to a single big summary file. Each of your notes consisted of several rows, like the following:

```
bed = 100
table = 150
furniture = bed + table
furniture = furniture + 10
```

which (probably) meant that you spent 250 on furniture and 10 on transport (probably). More generally, each row in your note was one of the following:

```
name = number
name = item + item
```

where `item` was either `number` or `name`, `number` was a natural number not exceeding 10^9 (with no leading zeroes) and `name` was an arbitrary sequence of small and capital letters.

While transferring your notes to the summary file, the accountant neither changed the order of rows of a note nor shuffled rows of different notes. But sometimes she changed names... She did it consistently though, changing all occurrences of the name in a single note to the same new one. Also, different names from the same note were rewritten as different names in the summary file. But there is no guarantee that it was consistent with your other notes — the name *beer* from one note could have been rewritten as *drink*, the name *tee* from the other note could have been rewritten as *drink* as well, while the *beer* from the next note could have been rewritten as *food*. Lets call her version of your note *a transcription* of it.

The problem is that your accountant has just quit (and, as it turned out, took most of your savings) and you are left with the summary file and a huge unsorted pile of notes. You want to check if it is possible that some of your notes were not recorded in the summary, so for each note you are searching for a piece of summary that can be its transcription.

Multiple Test Cases

The input contains several test cases. The first line of the input contains a positive integer $Z \leq 15$, denoting the number of test cases. Then Z test cases follow, each conforming to the format described in section *Single Instance Input.* For each test case, your program has to write an output conforming to the format described in section *Single Instance Output.*

Single Instance Input

The first line of an input instance contains an integer k, the number of notes ($1 \leq k \leq 50000$). The following lines first contain the descriptions of k notes, then the description of the summary file follows. The description of each note and the summary starts with a line consisting of a single integer d ($1 \leq d$). Then d lines follow, each containing a single row of a note or the summary. The format of each row is as described in the problem statement above. You can assume that names, numbers, signs "=" and "+" are separated by single spaces and each line is at most 100 characters long. Both the length of the text and the sum of all patterns lengths will be at most $3 * 10^6$.

Single Instance Output

The output for a single instance should consist of k lines. In the i-th line there should be the answer for the i-th note. It should be the number of *the first row* in the summary file, where a possible transcription of the i-th note starts (the first row of the summary file has number 1) or word `NONE` if no fragment of the summary file could be such possible transcription.

Example

Input	Output

2
3
4
bed = 100
table = 150
furniture = bed + table
furniture = furniture + 10
1
beer = 100
2
a = a + a
a = a + 10
10
furniture = 100
table = 150
furniture = furniture + table
furniture = furniture + 10
sofa = 100
stol = 150
meble = sofa + stol
meble = meble + 10
picie = 100
drink = 0100
2
1
a = 100 + 200
1
a = 100
2
x = 100 + 200
y = 100
	5
	1
	NONE
	1
	2