problem **16**

Family Tree

11 points

Introduction

In programming many problems can be solved by using parent-child relationships. This principle is used, for example, in the structure of HTML documents and the implementation of interface hierarchies. This approach is powerful because programs can apply rules to their data based on the notion of inheritance. Of course, the same data structure could also be used to track biological families, as in the recording and researching of people's family trees, or for selective breeding of food crops.

For this program, the relationship between two parents and one or more children is denoted by an expression like this: “`A + B : C , D , E .`” In this example A and B are the parents, while C, D, and E are the children.

There are two types of queries the program must be able to answer. The first is written like this: “`A > D ?`” which means, "is A an ancestor of D?" The second type of query is written like this: “`O ^ V ?`” which means, “do O and V share a common ancestor?”

Input

The first line of input is the number of parent-child expressions, followed by the expressions, one per line. Every expression will have two parents and at least one child. The next line is the number of queries, followed by the queries, one per line. Letters and symbols will be separated by a single space.

```
9
A + B : C , D , E .
F + G : H .
I + J : K , L .
C + H : M , N .
D + K : O .
L + E : P , Q , R , S .
N + Q : T , U , V .
O + S : W , X .
Y + H : Z .
7
K > O ?
F > W ?
B ^ U ?
O ^ V ?
A > Z ?
F > Z ?
X ^ Z ?
```

Output

The program must print each query, followed by the word TRUE or FALSE, indicating if the queried relationship exists. Note that a letter cannot be a parent/ancestor of itself.

```
K > O ? TRUE
F > W ? FALSE
B ^ U ? FALSE
O ^ V ? TRUE
A > Z ? FALSE
F > Z ? TRUE
X ^ Z ? FALSE
```