8071 Is-A? Has-A? Who Knowz-A?

Two familiar concepts in object oriented programming are the is-a and has-a relationships. Given two classes A and B, we say that A is-a B if A is a subclass of B; we say A has-a B if one of the fields of A is of type B. For example, we could imagine an object-oriented language (call it ICPC++) with code like that in Figure E.1, where the class `Day` is-a `Time`, the class `Appointment` is both a `DateBook` and a `Reminder`, and class `Appointment` has-a `Day`.

```
class Day extends Time
{
...
}
```

```
class Appointment extends Datebook, Reminder
{
    private Day date;
...
}
```

Figure E.1: Two ICPC++ classes.

These two relationships are transitive. For example if A is-a B and B is-a C then it follows that A is-a C. This holds as well if we change all the is-a's in the last sentence to has-a's. It also works with combinations of is-a's and has-a's: in the example above, `Appointment` has-a `Time`, since it has-a `Day` and `Day` is-a `Time`. Similarly, if class `DateBook` has-a `Year` then `Appointment` has-a `Year`, since `Appointment` is-a `DateBook`.

In this problem you will be given a set of is-a and has-a relationships and a set of queries of the form A is/has-a B. You must determine if each query is true or false.

Input

The input file contains several test cases, each of them as described below.

Input starts with two integers n and m, $(1 \leq n, m \leq 10000)$, where n specifies the number of given is-a and has-a relationships and m specifies the number of queries. The next n lines each contain one given relationship in the form c_1 r c_2 where c_1 and c_2 are single-word class names, and r is either the string "is-a" or "has-a". Following this are m queries, one per line, using the same format. There will be at most 500 distinct class names in the $n + m$ lines, and all class names in the last m lines will appear at least once in the initial n lines. All is-a and has-a relationships between the given classes can be deduced from the n given relationships. Is-a relationships can not be circular (apart from the trivial identity "x is-a x").

Output

For each test case, and for each query, display the query number (starting at one) and whether the query is true or false. Of course, for each test case the numbers start again at one.

Sample Input

```
5 5
Day is-a Time
Appointment is-a Datebook
```

```
Appointment is-a Reminder
Appointment has-a Day
Datebook has-a Year
Day is-a Time
Time is-a Day
Appointment has-a Time
Appointment has-a Year
Day is-a Day
```

Sample Output

```
Query 1: true
Query 2: false
Query 3: true
Query 4: true
Query 5: true
```