

The Hungarian Paul Erdős (1913–1996, speak as “Ar-dish”) not only was one of the strangest mathematicians of the 20th century, he was also one of the most famous. He kept on publishing widely circulated papers up to a very high age and every mathematician having the honor of being a co-author to Erdős is well respected.

Not everybody got the chance to co-author a paper with Erdős, so many people were content if they managed to publish a paper with somebody who had published a scientific paper with Erdős. This gave rise to the so-called *Erdős numbers*. An author who has jointly published with Erdős had Erdős number 1. An author who had not published with Erdős but with somebody with Erdős number 1 obtained Erdős number 2, and so on.

Today, nearly everybody wants to know which Erdős number he or she has. Your task is to write a program which computes Erdős numbers for a given set of scientists.

## Input

The first line of the input contains the number of scenarios.

The input for each scenario consists of a paper database and a list of names. It begins with the line  $PN$

where  $P$  and  $N$  are natural numbers. Following this line are  $P$  lines containing descriptions of papers (this is the paper database). A paper appears on a line by itself and is specified in the following way:

```
Smith, M.N., Martin, G., Erdos, P.: Newtonian forms of prime factors matrices
```

Note that umlauts like ‘ö’ are simply written as ‘o’. After the  $P$  papers follow  $N$  lines with names. Such a name line has the following format:

```
Martin, G.
```

## Output

For every scenario you are to print a line containing a string “Scenario  $i$ ” (where  $i$  is the number of the scenario) and the author names together with their Erdős number of all authors in the list of names. The authors should appear in the same order as they appear in the list of names. The Erdős number is based on the papers in the paper database of this scenario. Authors which do not have any relation to Erdős via the papers in the database have Erdős number “infinity”.

## Sample Input

```
1
4 3
Smith, M.N., Martin, G., Erdos, P.: Newtonian forms of prime factor matrices
Erdos, P., Reisig, W.: Stuttering in petri nets
Smith, M.N., Chen, X.: First oder derivates in structured programming
Jablonski, T., Hsueh, Z.: Selfstabilizing data structures
Smith, M.N.
Hsueh, Z.
Chen, X.
```

## Sample Output

```
Scenario 1
Smith, M.N. 1
Hsueh, Z. infinity
Chen, X. 2
```