2486 Company parties

The Acme company has an hierarchical organization, i.e., a tree-like structure with the CEO at the root and each other employee a child node of his/her manager. In addition to his/her position in the organization, each employee has a unique employee id (a string with no particular meaning) and a sociability measure (an integer number).

The CEO of the Acme company wants to organize a party for their employees. To make the party agreeable the CEO wants to make invitations such that:

- the CEO attends the party,
- an employee can be invited only if his/her direct manager is absent,
- the sum of the sociability measures of all who attend is a maximum.

Write a program that will determine the maximum sociability sum under these conditions for a given company structure.

Input

The input text consists of a number of company structures. The first line of a set is a title giving the company name. The company name may contain any printable non-space characters; and embedded spaces are also permitted. A single '#' on a line indicates the end of input.

The "name" line is followed by one line consisting of a single integer n, $1 \leq n \leq 100000$, that indicates the number of employees in this company. This line is followed by n further lines, one line for each employee.

Each "employee" line consists of three items separated by single spaces:

- an *employee id*, which is a sequence of 1 to 10 letters and/or digits,
- a *sociability measure*, which is an integer number between 0 and 100,
- a *manager id*, which is the employee id of the current employee's direct manager, or the character '-' for the CEO.

The *employee id* is a unique identifier within the company. The order in which the employees appear is arbitrary, i.e., not related to their employee or manager ids.

Output

There is a single output line for each company. Each output line consists of the company title, followed by a colon and a space, and finally the maximum attainable sociability measure under the above conditions.

Sample Input

```
ACME 1
1
IDO 10 -
ACME II
```

```
2
ID0 10 -
ID1 21 ID0
ACME, INC.
8
ID4 20 ID3
ID5 1 ID4
ID6 1 ID3
ID7 10 ID6
ID0 10 -
ID1 21 ID0
ID2 10 ID1
ID3 11 ID0
#
```

Sample Output

```
ACME 1: 10
ACME II: 10
ACME, INC.: 50
```