You are given a description of class inheritance in JSON format.
The description is an array of JSON objects that correspond to the classes.
Each JSON object has a field **name** that contains the class name, and a **parent** field that contains a list of object's direct ancestors names.

Example:	`[{"name": "A", "parents": []}, {"name": "B", "parents": ["A", "C"]}, {"name": "C" , "parents": ["A"]}]`

Python equivalent:	`class A:`
	`pass`
	
	`Class B (A, C):`
	`pass`
	
	`class C (A):`
	`pass`

It is guaranteed that no class is inherited from itself explicitly or indirectly, and that no class is inherited explicitly from one class more than once.

For each class, calculate the number of its descendants and output this information in the following format.
`<class name>: <number of descendants>`

Classes should be displayed in lexicographical order.

To understand why B has one child execute:
```
class B:
    pass
print(issubclass(B, B))
```

Sample Input:
`[{"name": "A", "parents": []}, {"name": "B", "parents": ["A", "C"]}, {"name": "C", "parents": ["A"]}]`
Sample Output:
A : 3
B : 1
C : 2

Sample Input:

```
[{"name": "B", "parents": ["A", "C"]}, {"name": "C", "parents":
["A"]}, {"name": "A", "parents": []}, {"name": "D", "parents":["C",
"F"]}, {"name": "E", "parents":["D"]}, {"name": "F", "parents":[]}]
```

Sample Output:

A : 5
B : 1
C : 4
D : 2
E : 1
F : 3