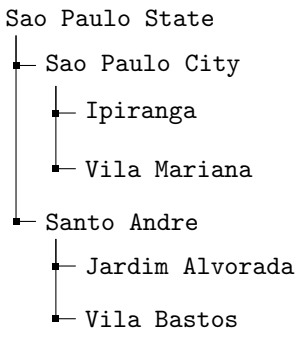
Locations

HVAR Consulting

2019

1 The Problem

In various regions of the world, physical locations can be represented in a father-child relationship, in which a father location contains a number of child locations, and those child locations can be fathers of their own children, and so on. This is a very useful way of representing locations because of its versatility. Take as an example the state of Sao Paulo, it has a number of children locations that are cities, which have a number of children locations that are districts. It can be represented with the following father-child relationship:



Once organized in that structure, one can easily count the number of child related nodes (children, grandchildren, grand-grandchildren, and so on) to that determined location. For example, the number of related nodes of the Santo Andre location is 2, while the number of related nodes of the Sao Paulo State location is 6.

Write a program that, given a list of locations and their relationships, determines the number child of related nodes to each location (children, grandchildren, grand-grandchildren, and so on). Note that, in this example, there are only three levels in the hierarchy: state $\rightarrow$ city $\rightarrow$ district. Your program must account for hierarchies with n number of levels, being $1 <= n <= 100$.

2 Input

The input will consist of two parts. The first line will contain a string that will determine the name of the root location (Sao Paulo Sate in the example). The remaining lines will contain two strings separated by a space character. Those will determine the name of a location and the name of its parent location. You can assume all names are one word long.

```
BCD
ABC BCD
DFG ABC
HIJ ABC
KLM BCD
NOP DFG
DXZ ABC
```

3 Output

Output will consist of a series of lines. Each line will consist of a string and an integer, accounting for the name of a location and the number of related notes. The locations must be outputted in alphabetical order.

```
ABC 4
BCD 6
DFG 1
DXZ 0
HIJ 0
KLM 0
NOP 0
```

Good luck and happy coding!

Problem created by HVAR Consulting, 2019.