## 2.Armies

Write a function that stores information about an army leader and his armies. The input will be an array of strings. The strings can be in some of the following formats:  
**"{leader} arrives"** – add the leader (no army)

**"{leader}: {army name}, {army count}"** – add the army with its count to the leader (if he exists)  
**"{army name} + {army count}"** – if the army exists somewhere add the count  
**"{leader} defeated"** – delete the leader and his army (if he exists)

When finished reading the input sort the **leaders** by **total army count** in **descending**. Then each **army** should be sorted by **count in descending**.

### Output

Print in the following format:  
**"{leader one name}: {total army count}  
>>> {armyOne name} - {army count}  
>>> {armyTwo name} - {army count}  
 …  
{leader two name}: {total army count}  
…"**

### Constrains

* The **new leaders** will always be **unique**
* When **adding a new army** to the leader, the army will be **unique**

### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| ['Rick Burr arrives', 'Fergus: Wexamp, 30245', 'Rick Burr: Juard, 50000', 'Findlay arrives', 'Findlay: Britox, 34540', 'Wexamp + 6000', 'Juard + 1350', 'Britox + 4500', 'Porter arrives', 'Porter: Legion, 55000', 'Legion + 302', 'Rick Burr defeated', 'Porter: Retix, 3205'] | Porter: 58507  >>> Legion - 55302  >>> Retix - 3205  Findlay: 39040  >>> Britox - 39040 |
| ['Rick Burr arrives', 'Findlay arrives', 'Rick Burr: Juard, 1500', 'Wexamp arrives', 'Findlay: Wexamp, 34540', 'Wexamp + 340', 'Wexamp: Britox, 1155', 'Wexamp: Juard, 43423'] | Wexamp: 44578  >>> Juard - 43423  >>> Britox - 1155  Findlay: 34880  >>> Wexamp - 34880  Rick Burr: 1500  >>> Juard - 1500 |