# rose3rose2roseProblem 4 – Ashes of Roses

The time has come for the great Rosen Flame Festival. Million Roses must be prepared for the reincarnation of the Icarus, The Great Phoenix God. Different regions that participate in the Festival will grow roses for Icarus to burn, upon resurrecting. You are asked to create a software program which oversees that process.

Regions can **grow** a **specific amount** of **specifically colored** roses.

The **valid input** will come in the following format:

* **Grow <{regionName}> <{colorName}> {roseAmount}** – grows the given amount of the given color roses in the given region.

The **region names** must always **start** with a **capital** **English alphabet letter**, and **consist** only of **lowercase** letters. The **color’s name** must consist only of **English alphabet letters** and **digits**. The **rose amount** must be an **integer**.   
Any input that **does NOT** **follow** the, specified above, rules is to be treated as **invalid**, and must be **ignored**.

If you receive an input with **existent** region, you should **add** the new color and roses to it. If even **the** **color exists, increase** their **current value** with the **given one**.

The input ends when you receive the command **“Icarus, Ignite!”**. That means that the Festival is ready to begin, and the roses are all ready and set. You must **print all the regions** and their roses, but in a **specific order**.

Regions must be **ordered by amount** of roses they have, in **descending order**, as prime criteria and in **alphabetical order**, as secondary criteria. Their colors must be **ordered by amount of roses**, in **ascending order**, as prime criteria and in **alphabetical order**, as secondary criteria.

### Input

* The input will consist only of the commands specified above.
* The input ends when you receive the command **“Icarus, Ignite!”**.

### Output

* As output you must print information about each region and its roses in the specified order.
* The format of output is:
  + **“{region1Name}**
  + **\*--{color1Name} | {roseCount}**
  + **\*--{color2Name} | {roseCount}**
  + **…”**

### Constraints

* The input lines can consist of any ASCII character. You must find only the valid ones.
* The input count of roses will be a valid integer in range [0, 231 – 1].
* Allowed time/memory: 100ms/16MB

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| **Grow <Dorne> <Indigo> 2000**  **Grow <Dorne> <Purple> 5000**  **Grow <Dorne> <Purple> 1000**  **Grow <Ironislands> <Blue> 20000**  **Grow <North> <Scarlet> 1000000**  **Icarus, Ignite!** | **North**  **\*--Scarlet | 1000000**  **Ironislands**  **\*--Blue | 20000**  **Dorne**  **\*--Indigo | 2000**  **\*--Purple | 6000** |

|  |  |
| --- | --- |
| **Input** | **Output** |
| **Grow <Valeofarryn> <White> 1**  **Grow <Stormlands> <White> 1**  **Grow <TheGift> <White> 1**  **Grow <Yiti> <White> 1**  **Grow <Valeofarryn> <Blue> 1**  **Grow <Stormlands> <Green> 1**  **Icarus, Ignite!** | **Stormlands**  **\*--Green | 1**  **\*--White | 1**  **Valeofarryn**  **\*--Blue | 1**  **\*--White | 1** |

Grow <Dorne> <Indigo> *2147483648*

Grow <Dorne> <Indigo> *2147483647*