## ForceBook

The force users are struggling to remember which side are the different forceUsers from, because they switch them too often. So you are tasked to create a web application to manage their profiles. You should store an information for every **unique forceUser**, registered in the application.

You will receive **several input lines** in one of the following formats:

{forceSide} | {forceUser}

{forceUser} -> {forceSide}

The forceUser and forceSide are strings, containing any character.

If you receive forceSide | forceUser:

* If there is no such **forceUser** and no such **forceSide -> create new forceSide** and **add** the **forceUser** to the corresponding side.
* Only **if there is no such forceUser** in any **forceSide** -> **add** the **forceUser** to the corresponding side.
* If there is such **forceUser** already -> **skip** the command and continue to the next operation.

If you receive a forceUser -> forceSide:

* If there is such **forceUser** already -> **change his/her side**.
* If there is no such **forceUser** in any **forceSide** -> add the **forceUser** to the corresponding **forceSide**.
* If there is no such **forceUser** and no such **forceSide -> create new forceSide** and **add** the **forceUser** to the corresponding side.
* Then you should print on the console: "{forceUser} joins the {forceSide} side!" .

You should end your program when you receive the command "Lumpawaroo". At that point you should print each force side, **ordered descending by forceUsers count, than ordered by name**. For each side print the **forceUsers**, **ordered by name**.

In case there are **no forceUsers in a side**, you **shouldn`t print** the side information.

### Input / Constraints

* The input comes in the form of commands in one of the formats specified above.
* The input ends, when you receive the command "Lumpawaroo".

### Output

* As output for each forceSide, **ordered descending by forceUsers count**, **then by name**, you must print all the forceUsers, **ordered by name alphabetically**.
* The output format is:

Side: {forceSide}, Members: {forceUsers.Count}

! {forceUser}

! {forceUser}

! {forceUser}

* In case there are **NO** forceUsers, don`t print this side.

### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| Light | Gosho  Dark | Pesho  Lumpawaroo | Side: Dark, Members: 1  ! Pesho  Side: Light, Members: 1  ! Gosho | We register Gosho in the Light side and Pesho in the Dark side. After receiving "Lumpawaroo" we print both sides, ordered by membersCount and then by name. |
| Lighter | Royal  Darker | DCay  Ivan Ivanov -> Lighter  DCay -> Lighter  Lumpawaroo | Ivan Ivanov joins the Lighter side!  DCay joins the Lighter side!  Side: Lighter, Members: 3  ! DCay  ! Ivan Ivanov  ! Royal | Although Ivan Ivanov doesn`t have profile, we **register** him and add him to the Lighter side.  We **remove DCay** from Darker side and add him to Lighter side.  We print only Lighter side because Darker side **has no members.** |