**Messages Manager**

Create a program that manages **messages** **sent** and **received** of **users**. You need to keep information about **username**, their **sent** and **received** messages. You will **receive** the **capacity** of **possible** **messages** **kept** **at once per user**. You will be receiving **lines** with commands until you receive the **"****Statistics"** command. There are three **possible** commands:

* **"Add={username}={sent}={received}":**
  + **Add** the **username**, his/her **sent** and **received** messages to your **records**. **If** person with the given **username** already **exists ignore** **the line**.
* **"Message={sender}={receiver}":**
  + **Check** if both usernames **exist** and if **they do**, **increase** the **sender’s** **sent messages** by 1 and the **receiver’s received messages** by 1. If anyone **reaches** the **capacity** (**first check the sender**), he/she should be **removed** fromthe **record** and you should **print** the following message:
    - **"****{username} reached the capacity!"**
* **"Empty={username}":**
  + Delete **all** records of the **given user**, **if** he **exists**. If "**All**" is **given as username** - delete **all records** you have.

In the end, you have to **print the count of users, each person** with his/her **messages** (the **count** of both **sent** and **received**) sorted in **descending order** by the **received messages** and **then by** their **username** in **ascending** order in the following format:

**Users count: {count}**

**{username} - {messages}**

**{username} - {messages}**

## Input

* On the **first** **line**, you will **receive** the **capacity** - an **integer** number in the range [1-10000].
* You will be receiving linesuntil you receive the **"Statistics"** command.
* The **initial messages** (**sent** and **received**)will **always** be **below** the **capacity**.
* The input will **always** be **valid**.

## Output

* Print the appropriate message after the **"Message"** command, **if** someone reaches the capacity.
* Print the users with their **messages** in the **format** described above.

## Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 10  Add=Mark=5=4  Add=Clark=3=5  Add=Berg=9=0  Add=Kevin=0=0  Message=Berg=Kevin  Statistics | Berg reached the capacity!  Users count: 3  Clark - 8  Mark - 9  Kevin - 1 |
| **Comments** | |
| First, we **receive** the **capacity** (10). Then we **start** **receiving** **commands**. The **first four commands** are for **adding new users**, so we do it. Then we have the command “**Message=Berg=Kevin**” and **Berg** **reached** the **capacity**, so we **remove** him, but **Kevin** has only his **received** messages **incremented**. When we receive the “**Statistics**” command, we **print** the **output** as described **above**. | |
|  | |
| 20  Add=Mark=3=9  Add=Berry=5=5  Add=Clark=4=0  Empty=Berry  Add=Blake=9=3  Add=Michael=3=9  Add=Amy=9=9  Message=Blake=Amy  Message=Michael=Amy  Statistics | Amy reached the capacity!  Users count: 4  Mark - 12  Michael - 13  Blake - 13  Clark - 4 |
|  | |
| 12  Add=Bonnie=3=5  Add=Johny=4=4  Empty=All  Add=Bonnie=3=3  Statistics | Users count: 1  Bonnie - 6 |