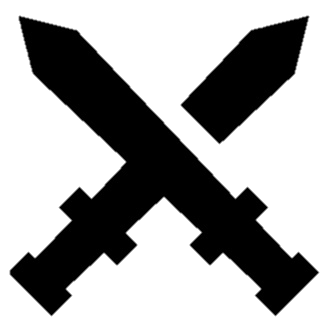
# Problem 3 - Battle Manager



Create a program that maintains battles. You need to keep information about **people**, the **health** and **energy** they have. You will be receiving **lines** with commands until you receive the **"Results"** command. There are three **possible** commands:

* **"Add:{personName}:{health}:{energy}"**:
  + **Add** the **person**, their **health**, and **energy** to your **records**. If a person with the given name already **exists**, just increase the **person's health** with the **current** one **provided**.
* **"Attack:{attackerName}:{defenderName}:{damage}"**:
  + **Check** if both people **exist** and if **they do**, **reduce both** the **defender's** **health** with the **damage** **given and the attacker's** **energy** **by** **1**. If the **defender's health** or the **attacker's energy** reaches **0** or **less** (first check the defender), the **person** is **disqualified,** and you need to **remove** them from your **records** and **print** the following **message**:
    - **"{name} was disqualified!"**
* **"Delete:{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 should **print the count of people left.** Then, **print each person** with their **health** and **energy.** The **output** should be in the format:

**"People count: {count}"**

**{firstPersonName} - {health} - {energy}**

**{secondPersonName} - {health} - {energy}**

**…**

**{NPersonName} - {health} - {energy}"**

### Input

* You will be receiving linesuntil you receive the **"Results"** command.
* The **health** is an **integer** number in the range **[1...100000]**.
* The **energy** is an **integer** number in the range **[1...100]**.
* The input will **always** be **valid**.

### Output

* Print the appropriate message after the **"Attack"** command **if** someone is **disqualified**.
* Print the people with their **health and energy** in the **format** described above.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Add:Mark:1000:5  Add:Clark:1000:2  Attack:Clark:Mark:500  Attack:Clark:Mark:800  Add:Charlie:4000:10  Results | Mark was disqualified!  Clark was disqualified!  People count: 1  Charlie - 4000 - 10 |
| Add:Bonnie:3000:5  Add:Kent:10000:10  Add:Johny:4000:10  Attack:Johny:Bonnie:400  Add:Johny:3000:5  Add:Peter:7000:1  Delete:Kent  Results | People count: 3  Bonnie - 2600 - 5  Johny - 7000 - 9  Peter - 7000 - 1 |
| Add:Bonnie:3000:5  Add:Johny:4000:10  Delete:All  Add:Bonnie:3333:3  Add:Aleks:1000:70  Add:Tom:4000:1  Results | People count: 3  Bonnie - 3333 - 3  Aleks - 1000 - 70  Tom - 4000 - 1 |

### JS Examples

The input will be provided as an array of strings.

|  |  |
| --- | --- |
| **Input** | **Output** |
| (["Add:Mark:1000:5",  "Add:Clark:1000:2",  "Attack:Clark:Mark:500",  "Attack:Clark:Mark:800",  "Add:Charlie:4000:10",  "Results"]) | Mark was disqualified!  Clark was disqualified!  People count: 1  Charlie - 4000 - 10 |
| (["Add:Bonnie:3000:5",  "Add:Kent:10000:10",  "Add:Johny:4000:10",  "Attack:Johny:Bonnie:400",  "Add:Johny:3000:5",  "Add:Peter:7000:1",  "Delete:Kent",  "Results"]) | People count: 3  Bonnie - 2600 - 5  Johny - 7000 - 9  Peter - 7000 - 1 |
| (["Add:Bonnie:3000:5", "Add:Johny:4000:10", "Delete:All",  "Add:Bonnie:3333:3",  "Add:Aleks:1000:70",  "Add:Tom:4000:1", "Results"]) | People count: 3  Bonnie - 3333 - 3  Aleks - 1000 - 70  Tom - 4000 - 1 |