Peter owns a zoo for exotic animals, but he is having difficulties keeping track of the animals' food and feeding schedule. He needs your help to facilitate the process.

Create a program that organizes the **daily feeding of the animals at the zoo**. You need to keep information about **animals**, their **daily food limit,** and the **areas they live in.** You will be receiving **lines** with commands until you receive the "**EndDay**" message. There are **two possible** commands:

- **"Add: {animal_name}-{needed_food_quantity}-{area}":**
  - Add the animal and the quantity of needed food to your records. It is guaranteed that the **names** of the animals are **unique**, and there will **never** be animals with the **same** name.
  - **If** the animal already **exists**, just **increase** the value of its **needed food** with the **given one**.
  - **You should keep track of the animal living in each area.**
- **"Feed: {animalName}-{food}":**
  - If the animal **exists**, **reduce** its **quantity of needed food** with the given **food for feeding**.
  - If the animal **does not exist**, **ignore** the command.
  - If its **limit** reaches **0** or **less**, the **animal** is considered **successfully fed**, and you need to **remove** it from your **records** and **print** the following **message**:
    - **"{animalName} was successfully fed"**

You need to know **the number of hungry animals** there are left in **each area**. If an animal has a daily food **limit above 0**, it is considered **hungry**.

In the end, you should **print each animal** with its **quantity of needed food** in the following format:

```
"Animals:"
" {animal_name} -> {needed_food_quantity}g"
…
" {animal_name} -> {needed_food_quantity}g"
```

Afterward, **print** only the **areas** with **hungry animals** in the following **format**:

```
"Areas with hungry animals:"
" {area_name}: {number_of_hungry_animals}"
…
" {area_name}: {number_of_hungry_animals}"
```

Input / Constraints

- You will be receiving lines until you receive the **"EndDay"** command.
- The **food** comes in **grams** and is an **integer** number in the range **[1...100000]**.
- The input will **always** be **valid**.
- There will never be a case in which an animal is in two or more areas at the same time.

Output

- Print the appropriate message after the **"Feed"** command **if** an **animal** is **fed**.
- Print the animals with their **quantity of needed food** in the **format** described above.
- Print the **areas** with the **number of hungry animals** in them in the **format** described above.

Examples

Input	Output
Add: Adam-4500-ByTheCreek Add: Maya-7600-WaterfallArea Add: Maya-1230-WaterfallArea Feed: Jamie-2000 EndDay	Animals: Adam -> 4500g Maya -> 8830g Areas with hungry animals: ByTheCreek: 1 WaterfallArea: 1
Add: Jamie-600-WaterfallArea Add: Maya-6570-WaterfallArea Add: Adam-4500-ByTheCreek Add: Bobbie-6570-WaterfallArea Feed: Jamie-2000 Feed: Adam-2000 Feed: Adam-2500 EndDay	Jamie was successfully fed Adam was successfully fed Animals: Maya -> 6570g Bobbie -> 6570g Areas with hungry animals: WaterfallArea: 2