# Animal Sanctuary

*You work in an animal sanctuary and every day you receive encrypted information about the animals that need help.*

Create a program that decrypts messages about animals, their kind and the country they are in. You will be given a number **n** – the number of **lines**, which you will receive. Afterwards on the next **n lines**, you will receive the **messages**. You are looking for:

* **animalName** 
  + contains **any ASCII character except for ";”**
* **animalKind** 
  + contains **any ASCII character except for “;”**
* **animalCountry** 
  + contains **only letters and spaces**

A **valid** message is in the following **format**: **"n:{animalName};t:{animalKind};c--{animalCountry}"**

The **output names, kinds** and **countries** of the animals should contain **only letters** and **white spaces. For example:**"**K@o$5a#la Be^4a5r**" is a **valid** match, but we need to print only – "**Koala Bear**". After each **valid message**, you should print a line in the format:

"**{animalName} is a {animalKind} from {country}"**

You need to knowthe **total weight** of **all** the **animals.** The weight of **each** **animal alone** is **calculated** by the **sum** of **every digit** in the **name** andthe **kind** of **the animal**. In the end print a line in the following **format** with the following **message**:

**"Total weight of all animals is {weight}KG"**.

## Input / Constraints

* First line will be a number **n** in range [1…100].
* The next **n** lines will be **strings**.

## Output

* Print each **valid** message in the format described above.
* Print the **total weight** of **all animals**.

## Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  n:M5%ar4#le@y;t:B3#e!!a2#2r;c--Australia  n:G3e%6org34e;t:C€$at2%;c--Africa  n:AlicE:Won;c-India | Marley is a Bear from Australia  George is a Cat from Africa  Total weight of animals: 34KG |
| **Comments** | |
| The first two lines are valid matches and the third is not, because it is not in the right format. We take the digits in the names and kinds of the two valid matches - M**5**%ar**4**#le@y, B**3**#e!!a**2**#**2**r, G**3**e%**6**or**34**ge, C€$at**2**% - 5 + 4 + 3 + 2 + 2 + 3 + 6 + 3 + 4 + 2 = 34. After each of the valid lines, we print the appropriate message. In the end we print the total weight of all animals, which is 34. | |
|  | |
| 4  n:Bo^%4b35454bie#$;t:Ele5ph#$34a%nt;c--Africa  n:Honey;t:Ti^^5ger;c--India  bla;t:1234a;c--America  n:A#$@545n;t:Cat241$@#23;cGermany | Bobbie is a Elephant from Africa  Honey is a Tiger from India  Total weight of animals: 42KG |

**Feed the Animals**

*The sanctuary needs to provide food for the animals and feed them, so your task is to help with the process*

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

* **"Add:{animalName}:{dailyFoodLimit}:{area}":**
  + **Add** the **animal** and **its** **daily food limit** 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** it already **exists**, just increase the value of the **daily** **food** **limit** with the **current** one that is **given**.
* **"Feed:{animalName}:{food}:{area}":**
  + **Check** if the animal **exists** and if **it does**, **reduce** its daily **food limit** with the given **food** **for** **feeding**. 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 count of** **hungry** **animals** there are left in **each area** in the end. If an animal has daily food **limit above 0**, it is considered **hungry**.

In the end, you have to **print each animal** with its **daily** food **limit** sorted in **descending order** by the **daily food limit** and **then by** its **name** in **ascending** order in the following format:

**Animals:**

**{animalName} -> {dailyFoodLimit}g**

**{animalName} -> {dailyFoodLimit}g**

Afterwards, **print** the **areas** with the **count of animals**, which are **not** **fed** in **descending** order by the **count** of **animals.** If an **area** has **0** **hungry animals** in it, **don't** print it. The **output** must be in the following **format**:

**Areas with hungry animals:**

**{areaName} : {countOfUnfedAnimals}**

**{areaName} : {countOfUnfedAnimals}**

## Input / Consrtaints

* You will be receiving linesuntil you receive the **"Last Info"** 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 **daily food limit** in the **format** described above.
* Print the **areas** with the **count of unfed** **animals** in them in the **format** described above.

## Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Add:Maya:7600:WaterfallArea  Add:Bobbie:6570:DeepWoodsArea  Add:Adam:4500:ByTheCreek  Add:Jamie:1290:RiverArea  Add:Gem:8730:WaterfallArea  Add:Maya:1230:WaterfallArea  Add:Jamie:560:RiverArea  Feed:Bobbie:6300:DeepWoodsArea  Feed:Adam:4650:ByTheCreek  Feed:Jamie:2000:RiverArea  Last Info | Adam was successfully fed  Jamie was successfully fed  Animals:  Maya -> 8830g  Gem -> 8730g  Bobbie -> 270g  Areas with hungry animals:  WaterfallArea : 2  DeepWoodsArea : 1 |
| **Comments** | |
| First, we receive the "**Add**" command, so we **add** "**Maya**" to our **records** and we keep her **daily food limit** - **7600**. We know that she is in **WaterfallArea**. We keep adding the new animals until we receive "**Maya**" **again** and we have to **increase** her food **limit** with **1230**, so it becomes **8830**. After that we receive "**Jamie**" and we need to **increase** his daily food **limit** with **560**, after which it **becomes** **1850**. Then we start receiving "**Feed**" commands. First, we must **decrease** **Bobbie's** food **limit** with **6300**, so it becomes **270**. Then, we need to decrease **Adam's** food **limit** with **4650**. It **becomes** **less than zero** and we **remove** **him** from the collection – he is **considered fed**, respectively that is **one less hungry** **animal** in the **area** that he is in – **ByTheCreek**. Then we "**Feed**" **Jamie** with **2000** and his **limit** becomes **less than zero**, so we print "**Jamie was successfully fed**" and we **remove** him from our records and note that there is **one** **less** **hungry animal** in his area – **RiverArea**. In the end, we **print the animals** we still have in our collection, with their daily food **limits** in **descending order** by the food **limits**. Afterwards we print only the **areas** in which there are **remaining** **hungry** **animals** and their **count** in **descending** order. | |
|  | |
| Add:Bonie:3490:RiverArea  Add:Sam:5430:DeepWoodsArea  Add:Bonie:200:RiverArea  Add:Maya:4560:ByTheCreek  Feed:Maya:2390:ByTheCreek  Feed:Bonie:3500:RiverArea  Feed:Johny:3400:WaterFall  Feed:Sam:5500:DeepWoodsArea  Last Info | Sam was successfully fed  Animals:  Maya -> 2170g  Bonie -> 190g  Areas with hungry animals:  RiverArea : 1  ByTheCreek : 1 |