Problem 4 – Dragon Army

Heroes III is the best game ever. Everyone loves it and everyone plays it all the time. Stamat is no exclusion to this rule. His favorite units in the game are all **types** of dragons – black, red, gold, azure etc… He likes them so much that he gives them **names** and keeps logs of their **stats**: **damage, health** and **armor**. The process of aggregating all the data is quite tedious, so he would like to have a program doing it. Since he is no programmer, it's your task to help him

You need to categorize dragons by their **type**. For each dragon, identified by **name,** keep information about his **stats.** Type is **preserved** as in the order of input, but dragons are **sorted** alphabetically by name. For each type, you should also print the average **damage**, **health** and **armor** of the dragons. For each dragon, print his own stats.

There **may** be **missing** stats in the input, though. If a stat is missing you should assign it default values. Default values are as follows: health **250**, damage **45**, and armor **10**. Missing stat will be marked by **null**.

The input is in the following format **{type} {name} {damage} {health} {armor}.** Any of the integers may be assigned null value. See the examples below for better understanding of your task.

If the same dragon is added a second time, the new stats should **overwrite** the previous ones. Two dragons are considered **equal** if they match by **both** name and type.

Input

- On the first line, you are given number N -> the number of dragons to follow
- On the next N lines you are given input in the above described format. There will be single space separating each element.

Output

- Print the aggregated data on the console
- For each type, print average stats of its dragons in format **{Type}::({damage}/{health}/{armor})**
- Damage, health and armor should be rounded to two digits after the decimal separator
- For each dragon, print its stats in format **-{Name} -> damage: {damage}, health: {health}, armor: {armor}**

Constraints

- N is in range [1...100]
- The dragon type and name are one word only, starting with capital letter.
- Damage health and armor are integers in range [0 … 100000] or **null**

Examples

Input	Output
5 Red Bazgargal 100 2500 25 Black Dargonax 200 3500 18 Red Obsidion 220 2200 35 Blue Kerizsa 60 2100 20 Blue Algordox 65 1800 50	Red::(160.00/2350.00/30.00) -Bazgargal -> damage: 100, health: 2500, armor: 25 -Obsidion -> damage: 220, health: 2200, armor: 35 Black::(200.00/3500.00/18.00) -Dargonax -> damage: 200, health: 3500, armor: 18 Blue::(62.50/1950.00/35.00) -Algordox -> damage: 65, health: 1800, armor: 50 -Kerizsa -> damage: 60, health: 2100, armor: 20

Input	Output
4 Gold Zzazx null 1000 10 Gold Traxx 500 null 0 Gold Xaarxx 250 1000 null Gold Ardrax 100 1055 50	Gold::(223.75/826.25/17.50) -Ardrax -> damage: 100, health: 1055, armor: 50 -Traxx -> damage: 500, health: 250, armor: 0 -Xaarxx -> damage: 250, health: 1000, armor: 10 -Zzazx -> damage: 45, health: 1000, armor: 10