# Problem 3. The progress of the heroes

You have been tasked to keep track of heroes and their attacks. A hero can attack in several phases and types. When it attacks, the hero will deal some damage in opponent.

You will receive input lines in the following format:  
{heroName} -> {attackType} -> {damage}

The heroName and attackType will be **strings**. The damage will be an **integer**. Your task is to store every **hero** and his **attacks**.   
If you receive an existent heroName, you should **add** the **new** **attack** to it.

A single **hero** may have **many attacks** with the **same type** and the **same damage**.

In some rare cases you may receive the following input:  
{heroName}

When you receive only a heroName, you must **check if there is** such a **hero**, and if there is, you must print all of its **attacks** by **order of input**.

The **input sequence ends** when you **receive** the command “Fight”.   
Then you must print all heroes and their attacks. The hero must be printed by **order of input**. Each **hero’s attacks** must be **ordered** by **damage** in **descending order**.

### Input

* The input will come in the form of lines in the format specified above.
* In some rare cases you may have only one element of the input – the heroName.
* The input sequence ends when you receive the command “Fight”.

### Output

* **Heroes** and their **attacks** must be printed in the following format:

“^ {heroName}  
 {attackType} <> {damage}  
 {attackType} <> {damage}

…”

* If you have received a heroName and you are **printing its attacks**, the order is – by **order of input**.
* If you have received the **ending command**, and you are printing the **heroes attacks**, the order is – by damage in **descending order**.

### Constrains

* The heroName and attackType are strings which may contain any ASCII character   
  (except ‘-’, ‘>’).
* The damage will be an **integer** in **range [0, 1.000.000.000]**.
* There will be **NO invalid** input data.
* Allowed time / memory: **100ms / 16 MB**.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Phantom Assassin -> Stifling Dagger -> 400  Meepo -> Physical -> 250  Phantom Assassin -> Physical -> 1200  Juggernaut -> Omnislash -> 2000  Juggernaut -> Physical -> 500  Fight | ^ Phantom Assassin  Physical <> 1200  Stifling Dagger <> 400  ^ Meepo  Physical <> 250  ^ Juggernaut  Omnislash <> 2000  Physical <> 500 |
| Kunkka -> Torrent -> 75  Meepo -> Poof -> 150  Kunkka -> Tidebringer -> 350  Meepo -> Poof -> 300  Kunkka -> Torrent -> 100  Meepo  Kunkka  Fight | ^ Meepo  Poof <> 150  Poof <> 300  ^ Kunkka  Torrent <> 75  Tidebringer <> 350  Torrent <> 100  ^ Kunkka  Tidebringer <> 350  Torrent <> 100  Torrent <> 75  ^ Meepo  Poof <> 300  Poof <> 150 |