# Problem 4 – Iron Girder

*As a thinker you are always given new tasks. This time you are working for Mr. Harry King. You have to take part in the new railway system and keep track on how things are going there.*

You will receive input lines in one of the following formats:

**{townName}:{time}->{passengersCount}**

If you receive the line above, Iron Girder has travelled to **certain town** for a **certain amount of time** with **certain count of passengers**. You need to keep track **for each town**. You have to save **the fastest time Iron Girder reaches** a town and the **total count of passengers for each town.**

**{townName}:ambush->{passengersCount}**

If you receive the line above, somewhere along the track to the current town **Iron Girder was ambushed and the passengers can't reach there.** If this happens you need to **set the time record for this town to "0" and remove the current count of passengers from the total count.** If it's the **first time Iron Girder travels to this town** then you **simply ignore this line.**

When you receive **"Slide rule"** you end the program and print data for each town in the following format:

**"{townName} -> Time: {fastestTime} -> Passengers: {totalCountPassengers}"**

The output should be ordered by best time and then by town's name. If a town is with no recorded time **(the time is equal to 0) or there are no passengers (count is equal or less than 0)** you **should not print it.**

## Input / Constraints

* Until you receive **"Slide rule"** you will be receiving participant submissions in one of the formats specified above
* The time will always be **positive** **integer in the range [1-1000]**
* The count of passengers will always be **positive integer in the range [1-100000]**

## Output

* Print recorded data in the following format:
* **"{townName} -> Time: {bestTime} -> Passengers: {totalCountPassengers}"**
* Allowed working **time** / **memory**: **100ms** / **16MB**

## Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comment** |
| Sto-Lat:8->120  Ankh-Morpork:3->143  Sto-Lat:9->80  Ankh-Morpork:4->143  Sto-Lat:3->20  Quirm:12->40  Quirm:13->29  Slide rule | Ankh-Morpork -> Time: 3 -> Passengers: 286  Sto-Lat -> Time: 3 -> Passengers: 220  Quirm -> Time: 12 -> Passengers: 69 | We have Sto-Lat multiple times, but we keep only the best time equal to 3 with the total count of passengers equal to 220. Ankh- Morpork is with fastest time 3, so we compare those two by names. Quirm comes third with time of 12. |
| Quirm:12->258  Ankh-Morpork:ambush->200  Ankh-Morpork:3->143  Sto-Lat:4->80  Ankh-Morpork:4->143  Ankh-Morpork:ambush->143  Sto-Lat:3->20  Ankh-Morpork:5->17  Slide rule | Sto-Lat -> Time: 3 -> Passengers: 100  Ankh-Morpork -> Time: 5 -> Passengers: 160  Quirm -> Time: 12 -> Passengers: 258 | The record time for Ankh-Morpork is equal to 5 since the previos one was set to 0 during the ambush. Note that we keep the count of passengers. |

“It was like ... like wizardry, but without the wizards and the mess.”