# Problem 4 – Population Aggregation

Write a program that receives as input information about **country, city and** its **population** and prints an **aggregated statistic**. There are **2 types of input** lines

* **{Country}**\**{city}**\**{population}**
* **{city}**\**{Country}**\**{population}**

The **country name always starts with a capital letter** and the **city name always starts with a lowercase letter**. The country name and the city name are both polluted with **prohibited symbols (@, #, $, & and digits from 0 to 9)**. Your task is to **clear all prohibited symbols** and **print aggregated data** about the **all the** **countries ordered alphabetically** in format:

**{Country} -> {number of cities}**

And **top 3 cities with biggest population** ordered in **descending** **order** **by population** in format:

**{city} -> {population}**

In case of **repeating towns**, count the last seen population for each town (ignore the others).

**Count all towns** in each country. In case of repeating towns for certain country, **count the duplicated towns**.

### Input

* The input data should be read from the console.
* It consists of a variable number of lines and ends when the command "**stop**" is received.
* The input data will always be valid and in the format described. There is no need to check it explicitly.

### Output

* The output should be printed on the console.
* Print the aggregated data for each country and city in the described format.

### Constraints

* The name of the city, country and the population will be separated from each other by **a back slash ('\')**.
* The **number of input lines** will be in the range [2 … 50].
* The **population count** of each city will be an integer in the range [0 … 263 − 1].
* Allowed working time for your program: 0.1 seconds. Allowed memory: 16 MB.

### Examples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| Bulgaria\sofia\123000  burgas\Bulgaria\4456576  stop | Bulgaria -> 2  burgas -> 4456576  sofia -> 123000 | Bulgaria\sofia\100  sofia\Bulgaria\200  stop | Bulgaria -> 2  sofia -> 200 |

|  |  |
| --- | --- |
| **Input** | **Output** |
| G$er&m@an@y\berlin\1234333  pa$r###is\F&r&a&n&c&e\30000000  Bulg@aria\varn@a@#$#\32145535  Bulgaria\pom$#or$ie\3131231  l$#ond$32on\U$#434565K43\98686644  ham$#bu4300r43g\Ger$man2@y\1324  stop | Bulgaria -> 2  France -> 1  Germany -> 2  UK -> 1  london -> 98686644  varna -> 32145535  paris -> 30000000 |