# Problem 4 - Ranking

Here comes the final and the most interesting part – the Final ranking of the candidate-interns. The final ranking is determined by the points of the interview tasks and from the exams in SoftUni. Here is your final task. You will receive some lines of input in the format **“{contest}:{password for contest}”** until you receive **“end of contests”**. Save that data because **you will need it later**. After that you will receive other type of inputs in format **“{contest}=>{password}=>{username}=>{points}”** until you receive **“end of submissions”**. Here is what you need to do.

* Check if the **contest is valid (if you received it in the first type of input)**
* Check if the **password is correct for the given contest**
* Save the user with the contest they take part in **(a user can take part in many contests)** and the points the user has in the given contest. If you receive the **same contest and the same user update the points only if the new ones are more than the older ones.**

At the end you have to print the info for the user with the **most points** in the format **“Best candidate is {user} with total {total points} points.”**. After that print **all students ordered by their names**. **For each user print each contest with the points in descending order**. See the examples.

## Input

* strings in format **“{contest}:{password for contest}”** until the **“end of contests”** command. There will be no case with two equal contests
* strings in format **“{contest}=>{password}=>{username}=>{points}”** until the **“end of submissions”** command.
* **There will be no case with 2 or more users with same total points!**

## Output

* On the first line print the best user in format **“Best candidate is {user} with total {total points} points.”.**
* Then print all students ordered as mentioned above in format:

**{user1 name}**

**# {contest1} -> {points}**

**# {contest2} -> {points}**

**{user2 name}**

**…**

## Constraints

* the strings may contain any ASCII character except from **(:, =, >)**
* the numbers will be in range **[0 - 10000]**
* second input is always valid

## Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Part One Interview:success  Js Fundamentals:Pesho  C# Fundamentals:fundPass  Algorithms:fun  end of contests  C# Fundamentals=>fundPass=>Tanya=>350  Algorithms=>fun=>Tanya=>380  Part One Interview=>success=>Nikola=>120  Java Basics Exam=>pesho=>Petkan=>400  Part One Interview=>success=>Tanya=>220  OOP Advanced=>password123=>BaiIvan=>231  C# Fundamentals=>fundPass=>Tanya=>250  C# Fundamentals=>fundPass=>Nikola=>200  Js Fundamentals=>Pesho=>Tanya=>400  end of submissions | Best candidate is Tanya with total 1350 points.  Ranking:  Nikola  # C# Fundamentals -> 200  # Part One Interview -> 120  Tanya  # Js Fundamentals -> 400  # Algorithms -> 380  # C# Fundamentals -> 350  # Part One Interview -> 220 |
| Java Advanced:funpass  Part Two Interview:success  Math Concept:asdasd  Java Web Basics:forrF  end of contests  Math Concept=>ispass=>Monika=>290  Java Advanced=>funpass=>Simona=>400  Part Two Interview=>success=>Drago=>120  Java Advanced=>funpass=>Petyr=>90  Java Web Basics=>forrF=>Simona=>280  Part Two Interview=>success=>Petyr=>0  Math Concept=>asdasd=>Drago=>250  Part Two Interview=>success=>Simona=>200  end of submissions | Best candidate is Simona with total 880 points.  Ranking:  Drago  # Math Concept -> 250  # Part Two Interview -> 120  Petyr  # Java Advanced -> 90  # Part Two Interview -> 0  Simona  # Java Advanced -> 400  # Java Web Basics -> 280  # Part Two Interview -> 200 |