# Problem 3 – Student Exams

You are given a sequence of exam results. Your task is to sort the exams by students and their results. Each exam holds an array of **students**. Each **student** has a **result**.

You should check whether the exam **result** value is **valid**. A **valid** **result** value is between **0** and **400** points. If a **student** has an invalid result value, you should **ignore** that line.

If a student has attended an exam **more** **than** **once**, it means the student went to a **makeup** exam and you should take into consideration the **higher** result. You should also **count** **how many makeup exams** the student went to.

In the output you should print all **exams** sorted **in order of appearance**. In each exam the **students** should be sorted by their **score** in **descending** order. If two students have the **same** **score** you should then sort them by **how** **many** **makeup** exams did **they** **take** in **ascending** order. If the **makeup exams are equal** then you should sort the **students** by **name alphabetically**.

### Input

The input will be read from an **HTTP GET** **request** holding parameter named **string**. Entries will be comma separated and each entry holds an exam result description in format "**student – exam : result**". The input data will always be valid and in the format described. There is no need to check it explicitly.

### Output

### Print a table that holds the exams sorted in order of appearance, and for each student – his result and the number of makeup exams taken. The students should be sorted by several criteria:

### result in descending order: the students with the highest results should be first.

### makeup exams taken in ascending order *(if there are several players with the same result)* – if two players have the same results, you should sort them by makeup exams taken.

### Their names sorted alphabetically in ascending order in case several students have the same results and number of makeup exams taken.

### Please follow exactly the table format from the example.

### Constraints

* The number of entries will be between 1 and 1000.
* Student names will consist of Latin letters and a single space. Its length is between 3 and 50 characters. Leading and trailing whitespaces should be removed.
* Exam names will consist of Latin letters only.
* Exam and student names will be unique.
* The result of each student will be in the range [0 … 1000].
* Whitespaces may be found or missing around the separators "-" and ":".
* Allowed working time for your program: 0.2 seconds. Allowed memory: 16 MB.

### Example

Input data is given on new rows for clarity. You will receive single line with comma separated values for processing.

|  |  |
| --- | --- |
| **Input** | |
| string | Georgi Petrov - Java : 360,  Marina - JavaScript : 200,  Marina - JavaScript : 300,  Vasil Dimitrov - PHP : 120,  Vasil Dimitrov-PHP: 550,  Vasil Dimitrov - PHP : 250 |
| **Output** | |
| <table>  <tr><th>Subject</th><th>Name</th><th>Result</th><th>MakeUpExams</th>  <tr><td>Java</td><td>Georgi Petrov</td><td>360</td><td>0</td></tr>  <tr><td>JavaScript</td><td>Marina</td><td>300</td><td>1</td></tr>  <tr><td>PHP</td><td>Vasil Dimitrov</td><td>250</td><td>1</td></tr>  </table> | |

|  |  |
| --- | --- |
| **Input** | |
| string | Johnny Bravo - PHP : 300,  Johnny Bravo-PHP: 600,  Nikola Ivanov - PHP: 350,  Johnny Bravo - PHP : 400 |
| **Output** | |
| <table>  <tr><th>Subject</th><th>Name</th><th>Result</th><th>MakeUpExams</th>  <tr><td>PHP</td><td>Johnny Bravo</td><td>400</td><td>1</td></tr>  <tr><td>PHP</td><td>Nikola Ivanov</td><td>350</td><td>0</td></tr>  </table> | |