# Problem 4 – softuniTunes

Have you ever wanted to download songs for free? I mean, legally… Well, SoftUni is developing a tracker called suTunes and they need a skillful PHP programmer who knows how to get things done.

Your task is to write a PHP program that receives **information about several songs**, **filters the songs according to the artist selected** and **sorts the songs in given order**. For example we are given the following string:

Krali Marko | Rock | Nakov, Joro, Pesho | 1250 | 4.8

Sladkarnica Malinka | Jazz | Pencho, Nakov, Gero | 728 | 4.3

Hvani me Minke | Rock | Kaloqn, Tosho | 1930 | 4.4

V starite Karpati | Rock | Nakov | 1029 | 4.1

Rakiichice | Rock | Ceco, Joro, Nakov, Preslav, Petya | 453 | 4.2

The songs' properties are **Name**, **Genre**, **Artists**, **Downloads** and **Rating**. They are separated by **|**. Artists are separated by comma. As a table, it should look as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Genre | Artists | Downloads | Rating |
| Krali Marko | Rock | Nakov, Joro, Pesho | 1250 | 4.8 |
| Sladkarnica Malinka | Jazz | Pencho, Nakov, Gero | 728 | 4.3 |
| Hvani me Minke | Rock | Kaloqn, Tosho | 1930 | 4.4 |
| V starite Karpati | Rock | Nakov | 1029 | 4.1 |
| Rakiichice | Rock | Ceco, Joro, Nakov, Preslav, Petya | 453 | 4.2 |

You can sort a song by **name**, **genre**, **downloads** and **rating**. The order in which they are sorted is either **ascending** or **descending**. Example: we are given the artist **'Nakov'**, sort property **'rating'** and order **'ascending'**. The only songs which contains 'Nakov' as an artist are **Krali Marko**, **Sladkarnica Malinka**, **V starite Karpati** and **Rakiichice** (marked in gray). We sort those songs by **rating** in **ascending order**. Additionally, the **artists** should always be sorted in **alphabetical order**. The result should be the following table (printed as HTML table like in the examples below):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Genre | Artists | Downloads | Rating |
| V starite Karpati | Rock | Nakov | 1029 | 4.1 |
| Rakiichice | Rock | Ceco, Joro, Nakov, Petya, Preslav | 453 | 4.2 |
| Sladkarnica Malinka | Jazz | Gero, Nakov, Pencho | 728 | 4.3 |
| Krali Marko | Rock | Joro, Nakov, Pesho | 1250 | 4.8 |

If the sort property has equal values (e.g. genre 'Rock' 3 times), a **secondary sorting by song name in ascending order** should be performed.

### Input

The input will be read from an **HTTP GET request**. The **text** will be received from a **textarea with name 'text'**, holding **each song on a separate line**. The **artist** will be received from an **input text field with name 'artist'** – the artists will be separated by a comma and space **", "**. The **sort property** will be received from a **select field with name 'property'**. The **order** will be received from a **select field with name 'order'**.

### Output

The songs should be printed as an HTML table inside **<table></table>** tags. Each song should be printed in **<tr></tr>** tags. Each song property should be printed in **<td></td>** tags. There should be a heading row with **<th></th>** tags containing each song property – **Name**, **Genre**, **Artists**, **Downloads** and **Rating**.

Ensure all your output data is correctly encoded as HTML (use the **htmlspecialchars()** function).

### Constraints

* All songs will be in the format above: properties separated by **|** (with possible whitespace around them).
* Song names will always be **unique**.
* **Downloads** will be an integer number in the range [0…100000].
* **Rating** will be a floating-point number in the range [0…5].
* **Artist** will be a string value.
* **Property** will be one of the values: **name**, **genre**, **downloads**, **rating**.
* **Order** will be one of the values: **ascending**, **descending**.

### Examples

|  |  |
| --- | --- |
| **Input** | |
| text | Krali Marko | Rock | Nakov, Joro, Pesho | 1250 | 4.8  Sladkarnica Malinka | Jazz | Pencho, Nakov, Gero | 728 | 4.3  Hvani me Minke | Rock | Kaloqn, Tosho | 1930 | 4.4  V starite Karpati | Rock | Nakov | 1029 | 4.1  Rakiichice | Rock | Ceco, Joro, Nakov, Preslav, Petya | 453 | 4.2 |
| artist | Nakov |
| property | rating |
| order | ascending |
| **Output** | |
| <table>  <tr><th>Name</th><th>Genre</th><th>Artists</th><th>Downloads</th><th>Rating</th></tr>  <tr><td>V starite Karpati</td><td>Rock</td><td>Nakov</td><td>1029</td><td>4.1</td></tr>  <tr><td>Rakiichice</td><td>Rock</td><td>Ceco, Joro, Nakov, Petya, Preslav</td><td>453</td><td>4.2</td></tr>  <tr><td>Sladkarnica Malinka</td><td>Jazz</td><td>Gero, Nakov, Pencho</td><td>728</td><td>4.3</td></tr>  <tr><td>Krali Marko</td><td>Rock</td><td>Joro, Nakov, Pesho</td><td>1250</td><td>4.8</td></tr>  </table> | |

|  |  |
| --- | --- |
| **Input** | |
| text | Krali Marko | Rock | Nakov, Joro, Pesho | 1250 | 4.8  Sladkarnica Malinka | Jazz | Pencho, Nakov, Gero | 728 | 4.3  Hvani me Minke | Rock | Kaloqn, Tosho | 1930 | 4.4  V starite Karpati | Rock | Nakov | 1029 | 4.1  Rakiichice | Rock | Ceco, Joro, Nakov, Preslav, Petya | 453 | 4.2 |
| artist | Nakov |
| property | genre |
| order | ascending |
| **Output** | |
| <table>  <tr><th>Name</th><th>Genre</th><th>Artists</th><th>Downloads</th><th>Rating</th></tr>  <tr><td>Sladkarnica Malinka</td><td>Jazz</td><td>Gero, Nakov, Pencho</td><td>728</td><td>4.3</td></tr>  <tr><td>Krali Marko</td><td>Rock</td><td>Joro, Nakov, Pesho</td><td>1250</td><td>4.8</td></tr>  <tr><td>Rakiichice</td><td>Rock</td><td>Ceco, Joro, Nakov, Petya, Preslav</td><td>453</td><td>4.2</td></tr>  <tr><td>V starite Karpati</td><td>Rock</td><td>Nakov</td><td>1029</td><td>4.1</td></tr>  </table> | |