## Problem 3 – Sort Table

You are given a **HTML table** with 3 columns: **product**, **price** and **votes**. Write a JavaScript function to sort the table rows by **price** (as number, increasingly).

### Input

The input is passed to the first JavaScript function found in your code as **array of strings** in the format of the examples below. The HTML table will always have a header row and 3 columns: product, price and votes. **No whitespace** will be found between the tags and between the tags and the tags values.

The input data will always be valid and in the format described. There is no need to check it explicitly.

### Output

Print at the console the **HTML table** sorted by column "**Price**" (as number, increasingly). Please don't change anything in the table, just rearrange its data rows. When several rows hold **equal prices**, use the **product name as** **second sort criteria** (sort by product name alphabetically).

### Constraints

* The **number of rows** in the table is in the range [1…10 000].
* All **prices** are number in the range [0…100 000].
* Allowed working time: 0.2 seconds. Allowed memory: 16 MB.

**function** *solve*(input]) {  
  
 **var *rows*** = [];  
 **for** (**var *i*** = 2; ***i*** < input.length - 1; ***i***++) {  
 **var *rowData*** = input[***i***];  
 **var *regex*** = /<td>.\*?<\/td><td>(.\*?)<\/td>/g;  
 **var *match*** = ***regex***.exec(***rowData***);  
 **var *price*** = Number(***match***[1]);  
 **var *row*** = { **price**: ***price***, data: ***rowData*** };  
 ***rows***.push(***row***);  
 }  
 ***rows***.sort(**function** (a, b) {  
 **if** (a.***price*** != b.***price***) {  
 **return** a.***price*** - b.***price***;  
 } **else** {  
 **return** a.data == b.data ? 0 : a.data < b.data ? -1 : 1;  
 }  
 });  
 **console**.log(input[0]);  
 **console**.log(input[1]);  
 **for** (**var *i*** = 0; ***i*** < ***rows***.**length**; ***i***++) {  
 **console**.log(***rows***[***i***].data);  
 }  
 **console**.log(input[input.length - 1]);  
  
  
}

### Examples

|  |
| --- |
| **Input** |
| <table>  <tr><th>Product</th><th>Price</th><th>Votes</th></tr>  <tr><td>Vodka Finlandia 1 l</td><td>**19.35**</td><td>+12</td></tr>  <tr><td>Ariana Radler 0.5 l</td><td>**1.19**</td><td>+33</td></tr>  <tr><td>Laptop HP 250 G2</td><td>**629**</td><td>+1</td></tr>  <tr><td>Kamenitza Grapefruit 1 l</td><td>**1.85**</td><td>+7</td></tr>  <tr><td>Ariana Grapefruit 1.5 l</td><td>**1.85**</td><td>+7</td></tr>  <tr><td>Coffee Davidoff 250 gr.</td><td>**11.99**</td><td>+11</td></tr>  </table> |
| **Output** |
| <table>  <tr><th>Product</th><th>Price</th><th>Votes</th></tr>  <tr><td>Ariana Radler 0.5 l</td><td>**1.19**</td><td>+33</td></tr>  <tr><td>Ariana Grapefruit 1.5 l</td><td>**1.85**</td><td>+7</td></tr>  <tr><td>Kamenitza Grapefruit 1 l</td><td>**1.85**</td><td>+7</td></tr>  <tr><td>Coffee Davidoff 250 gr.</td><td>**11.99**</td><td>+11</td></tr>  <tr><td>Vodka Finlandia 1 l</td><td>**19.35**</td><td>+12</td></tr>  <tr><td>Laptop HP 250 G2</td><td>**629**</td><td>+1</td></tr>  </table> |