# 3. SoftUni Students



*Unlocking boundless potential through code.*

Write a function called **softuni\_students** that **receives information** about students, their course id, and course name and **returns the result**. The function will receive a **different number of arguments** and **keyword arguments**. The arguments will be passed as follows:

* The first arguments will be **tuple** **key-value pairs** – the **first** **one** is the **course** **id** **(string)** and the **second** one is the **username** (every username will be **unique**) of the **student** **(string);**
* The following **data** will be the **keyword arguments** - the **first** one is the **course id (string)**, and the **second** one is the **course** **name** **(string)**;

After receiving the information and calling the function:

* You should **link** the **student** to their **course**.
* Sort the data by **student’s username (alphabetically)**, and if a **student's course ID is not linked** to a certain course, **consider the student invalid**.
* Keepin mind that **one course could be linked to one or more students**.

In the end:

* Return all **valid** **students** eachone **on a new line**:

**"\*\*\* A student with the username {student username} has successfully finished the course {course\_name}!"**

* Afterward, if there are **invalid** students **return** them with the following message:

**"!!! Invalid course students: {student\_1, student\_2, student\_N}"**

***Note: Submit only the function in the judge system.***

### Input

* There will be **no input from the console**, just parameters passed to your function.

### Output

* Return **the** **string** as described above.

### Constraints

* The **arguments** willbealways **before the keyword arguments**.
* Each **tuple** given will always contain the **course id** of a **student**.
* You will always receive at least one **tuple** with **course** id and **student**.
* You will always receive at least one valid **course** **id** and **course** **name**.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| print(softuni\_students(  ('id\_1', 'Kaloyan9905'),  id\_1='Python Web Framework', )) | \*\*\* A student with the username Kaloyan9905 has successfully finished the course Python Web Framework! |
| print(softuni\_students(  ('id\_7', 'Silvester1'),  ('id\_32', 'Katq21'),  ('id\_7', 'The programmer'),  id\_76='Spring Fundamentals',  id\_7='Spring Advanced', )) | \*\*\* A student with the username Silvester1 has successfully finished the course Spring Advanced! \*\*\* A student with the username The programmer has successfully finished the course Spring Advanced! !!! Invalid course students: Katq21 |
| print(softuni\_students(  ('id\_22', 'Programmingkitten'),  ('id\_11', 'MitkoTheDark'),  ('id\_321', 'Bobosa253'),  ('id\_08', 'KrasimirAtanasov'),  ('id\_32', 'DaniBG'),  id\_321='HTML & CSS',  id\_22='Machine Learning',  id\_08='JS Advanced', )) | \*\*\* A student with the username Bobosa253 has successfully finished the course HTML & CSS! \*\*\* A student with the username KrasimirAtanasov has successfully finished the course JS Advanced! \*\*\* A student with the username Programmingkitten has successfully finished the course Machine Learning! !!! Invalid course students: DaniBG, MitkoTheDark |