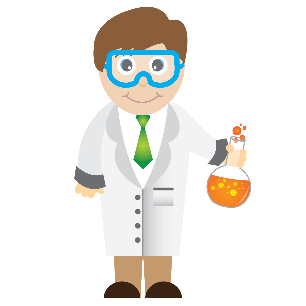
# Scientists



*You can finally call yourself a real scientist. And since you do your job so good, you are now promoted as Science Manager. Your very final task is to manage all the other scientists and their researches.*

## Classes

#### class Biologist

The Biologist class should receive a **name** and an **age**. Upon instance creation there should also be a variable **researched\_species** that stores the researched species (**list**). The class should also have a **method** called **add\_species**. If the passed value is **not** from the **Species** class, raise: **'This is not a species'**

The **string representation** of a **Biologist** should be:

**{name} is {age} years old Biologist with {researched\_species\_cont} researched species:**

If the biologist **has researched species** add the following:

**{string representation of each species on separate lines}**

#### class Chemist

The Chemist class should receive a **name** and an **age**. Upon instance creation there should also be a variable **researched\_elements** that stores the researched elements. The class should also have a **method** called **add\_element**. If the passed value is **not** from the **Element class**, raise: **'This is not an element'**.

The **string representation** of a **Chemist** should be:

**{name} is {age} years old Chemist with {researched\_elements\_cont} researched elements:**

If the chemist **has researched elements** add the following:

**{string representations of each element on separate lines}**

### Elements/Species

#### class Species

The species can **only** be **researched** by **Biologist**. A species class should receive **name** and **type**

There should also be a method that returns the **string representation** of the species in the following format: **"- {name} ({type})"**

#### class Element

The elements can **only** be **researched by Chemists**. An element class should receive a **name** and **type**

There should also be a method that returns the **string representation** of the element in the following format: **"- {name} ({type})"**

## Input

On the first line you will receive a **number** (**n**). On the next **'n'** lines you will receive **commands** to add **biologists/chemists/elements/species**. After that you will receive another **number** (**m**). On the next **'m'** lines you will have to **test some methods** on some scientists (use the validations from above).

Error messages (**check** them in that **order**):

* If the **scientist** does **not exist**, print: **"No such scientist"**
* If the **species/element** does **not exist**, print: **"No such element or species"**
* If the **method** does **not exist** in the class of the given scientist, print: **"Scientist has no such method"**

## Output

Print the output as described in each class above. The scientists should be **sorted** by **age in descending** then by **name and ascending**. **Species/Elements** should be sorted by **name in ascending**.

## Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5  Biologist('Peter', 21)  Chemist('Marry', 35)  Species('Lion', 'mammal')  Element('Oxygen', 'gas')  Element('Hydrogen', 'gas')  4  Peter-add\_species-Lion  Marry-add\_species-Lion  Marry-add\_element-Oxygen  Marry-add\_element-Hydrogen | Scientist has no such method  Marry is 35 years old Chemist with 2 researched elements:  - Hydrogen (gas)  - Oxygen (gas)  Peter is 21 years old Biologist with 1 researched species:  - Lion (mammal) |
| **Comment** | |
| First we add all scientists, species and elements  We add the Lion to the species of Peter  Since Marry is a Chemist and she does not have add\_species method, we print an error message  The other two commands are valid, so we execute them  We sort the output and we print it | |

*Where do hippos go to university? ... Hippocampus*