

Warehouse Machine

Here is your final task. You have to manage the storage of a coffee warehouse. You need to be able to add and sell coffee products, and do some revisions and inspections.

You need to **store coffee brands** and for **each brand** you store different **coffee (name, expire date and quantity)**. You can use strings to compare the date, not the Date class. The input is an **array of strings**. Each string contains information about what you need to do. The **first part** will be the **'command'**. It can be some of the following:

- **'IN'** – you need to **add coffee products**
- **'OUT'** – you need to **unload coffee products**
- **'REPORT'** – print **all the products in the warehouse as they were passed in the input**
- **'INSPECTION'** – print all the **products SORTED**

'REPORT' command:

Print **all brands with all of the coffee** in the following format:

```
'>>>> REPORT! <<<<<'
```

```
'Brand: {nameOfBrand}:'
```

```
'-> {coffeeName} -> {expireDate} -> {quantity}.'
```

'INSPECTION' command:

Same as the 'REPORT' command, but the **brands** should be **sorted alphabetically** and for each brand the **coffee should be sorted by quantity in descending order**. Print in the following format:

```
'>>>> INSPECTION! <<<<<'
```

```
'Brand: {nameOfBrand}:'
```

```
'-> {coffeeName} -> {expireDate} -> {quantity}.'
```

'IN' command:

Example: **'IN, Lavazza, Crema e Gusto, 2023-05-01, 5'**. You get the **coffee brand**, next you get the **name of the coffee**, then the **expire date** and the **quantity**.

- If this is the **first time you receive this brand** of coffee, **add it in the storage**, then **add the coffee with all of the other info**

- If the **brand already exists in the storage**, but the **coffee does NOT**, **add the coffee to the brand with all of its info**

- If the **brand and the coffee exist**, check the date. If the **new coffee expires later than the old**

one, replace it. If they are equal, just **add the new quantity to the old one**. Otherwise, do nothing.

'OUT' command:

Example: ' OUT, Lavazza, Crema e Gusto, 2023-05-01, 2'. Check for the **brand and the coffee**. If you **have them in storage**, check if you have some that **expires after the given date** and check if you **have the quantity needed**. If you have, **remove that quantity** of that coffee from the storage.

Input

- An array of strings

Output

- When you get the command '**REPORT**' or '**INSPECTION**' print the right result

Example

Input	Output
["IN, Batdorf & Bronson, Espresso, 2025-05-25, 20", "IN, Folgers, Black Silk, 2023-03-01, 14", "IN, Lavazza, Crema e Gusto, 2023-05-01, 5", "IN, Lavazza, Crema e Gusto, 2023-05-02, 5", "IN, Folgers, Black Silk, 2022-01-01, 10", "IN, Lavazza, Intenso, 2022-07-19, 20", "OUT, Dallmayr, Espresso, 2022-07-19, 5", "OUT, Dallmayr, Crema, 2022-07-19, 5", "OUT, Lavazza, Crema e Gusto, 2020-01-28, 2", "REPORT", "INSPECTION",]	>>>>> REPORT! <<<<< Brand: Batdorf & Bronson: -> Espresso -> 2025-05-25 -> 20. Brand: Folgers: -> Black Silk -> 2023-03-01 -> 14. Brand: Lavazza: -> Crema e Gusto -> 2023-05-02 -> 3. -> Intenso -> 2022-07-19 -> 20. >>>>> INSPECTION! <<<<< Brand: Batdorf & Bronson: -> Espresso -> 2025-05-25 -> 20. Brand: Folgers: -> Black Silk -> 2023-03-01 -> 14. Brand: Lavazza: -> Intenso -> 2022-07-19 -> 20. -> Crema e Gusto -> 2023-05-02 -> 3.