# 3. Shopping Cart

*Peter has decided to invite some guests. He should go shopping, but he will need help because   
there are too many things he needs to remember. Would you assist him?*

Write a function called **shopping\_cart** that **adds products to a shopping cart** for the following three types of meals: **"Soup"**, **"Pizza"**, and **"Dessert"**. Every meal has a **limit of products** that can be added to it:

* Soup: **3**
* Pizza: **4**
* Dessert: **2**

Once you **reach the limit of a meal**, you should **stop adding products** to that meal.

The function will receive a **different number of arguments**. The arguments will be passed as **tuples with two elements** - the **first** one is the **type of meal**, and **the second** is the **product for the meal**. You need to take **each argument** and make a **dictionary** with the **meal's** **name** as a key and the **products** **as a** **value** of the corresponding key.

There are some additional requirements:

* If you receive the **same product** for the **same meal** **more than once**, you **must not add it** again.
* If you run into the **word** "**Stop**" **(not tuple) as an argument**, you must immediately **stop** **adding products** to the cart - just **sort** and **return** the desired result as described below.

In the end, sort the **meals** by the **number of bought products in descending order**. If **there are meals** with an **equal number of products**, sort them (the meals) by **their name** in **ascending order** (alphabetically). For **each meal** sort **its** **products** in **ascending** **order** (alphabetically).

**Return** an output as described below.

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

### Input

* There will be **no input**, just parameters passed to your function

### Output

* **Return a string** for **each of the 3 types of a meal** of the **sorted result** in the format**:**
  + **"{meal\_type}:"**

**" - {first\_product\_added}"**

**" - {second\_product\_added}"**

**…**

**" - {Nth\_product\_added}"**

* If there are **no products** given for a meal, return **just** **its name in the format shown above**.
* If there are **NO products** in the cart (at all), **return** the message: "**No products in the cart!"**

### Constrains

* Each **tuple** given will always contain the **type of meal** in the first position and a **product** in the second.
* There will be **no other meals** passed than **"Soup"**, **"Pizza"**, and **"Dessert"**.

### Examples

|  |  |
| --- | --- |
| **Test Code** | **Output** |
| print(shopping\_cart(  ('Pizza', 'ham'),  ('Soup', 'carrots'),  ('Pizza', 'cheese'),  ('Pizza', 'flour'),  ('Dessert', 'milk'),  ('Pizza', 'mushrooms'),  ('Pizza', 'tomatoes'),  'Stop',  )) | Pizza:  - cheese  - flour  - ham  - mushrooms  Dessert:  - milk  Soup:  - carrots |
| print(shopping\_cart(  ('Pizza', 'ham'),  ('Dessert', 'milk'),  ('Pizza', 'ham'),  'Stop',  )) | Dessert:  - milk  Pizza:  - ham  Soup: |
| print(shopping\_cart(  'Stop',  ('Pizza', 'ham'),  ('Pizza', 'mushrooms'),  )) | No products in the cart! |