# 3. Pets Hotel

*You are a pet hotel owner.   
You need to organize the accommodation for the new-coming pets.*

Write a function called **accommodate\_new\_pets** that **receives information** about the available capacity of the hotel, the maximum weight allowed per pet, the pet types, and their weight, and **returns the result after the accommodation**. The function will receive a **different number of arguments**. The arguments will be passed as follows:

* The first argument will be the available capacity of your hotel - an **integer** in the **range [0, 50];**
* The second argument will be the maximum weight limit - a **float number** representing the pet’s maximum allowed weight;
* The following arguments will be the **tuples with two elements** - the **first** one is the **pet type (string)**, and the **second** one is **the pet weight** **(float)**;

After receiving the information and calling the function, the program should **start tracking the accommodation process**:

* Take the **pet type** from each tuple **successively** and **if you have enough capacity**, **accommodate it**, and proceed to the next one. Keep in mind that you will also need to track the **total number of pets** for **each pet type** you accommodate.
* If a pet’s **weight** exceeds the maximum weight limit, **ignore it**, and proceed to the next one.
* If the available **capacity is 0 (zero)**, **STOP accommodating**!
  + You are not supposed to check the weight of the unaccommodated pets (if any) when you run out of space.

In the end:

* If you’ve managed to **accommodate all pets**, return the message: **"All pets are accommodated! Available capacity: {available\_capacity}."**
* **Otherwise**, return the message: **"You did not manage to accommodate all pets!"**
* On the following lines **return** the accommodated **pet types** and **number of pets, ordered ascending (alphabetically) by pet type.** Each on a new line:

**"Accommodated pets:**

**{pet\_type1}: {number}**

**{pet\_type2}: {number}**

**…**

**{pet\_typeN}: {number}"**

***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 one of the **strings** **shown above** depending on the result **and the details** about accommodated pets **as described**.

### Constraints

* The **first** argument will always be an **integer**.
* The **second** argument will always be a **float** **number**.
* Each **tuple** given will always contain the **pet type** and **pet weight**.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| print(accommodate\_new\_pets(  10,  15.0,  ("cat", 5.8),  ("dog", 10.0),  )) | All pets are accommodated! Available capacity: 8.  Accommodated pets:  cat: 1  dog: 1 |
| print(accommodate\_new\_pets(  10,  10.0,  ("cat", 5.8),  ("dog", 10.5),  ("parrot", 0.8),  ("cat", 3.1),  )) | All pets are accommodated! Available capacity: 7.  Accommodated pets:  cat: 2  parrot: 1 |
| print(accommodate\_new\_pets(  2,  15.0,  ("dog", 10.0),  ("cat", 5.8),  ("cat", 2.7),  )) | You did not manage to accommodate all pets!  Accommodated pets:  cat: 1  dog: 1 |