Exercises: Arrays Advanced

1. Train

You will be given an **array of strings**.

The **first** element will be **a string containing wagons** (numbers). Each number inside the string represents **the number of passengers that are currently in a wagon**.

The **second** element in the array will be **the max capacity of each wagon** (single number).

The **rest** of the elements will be **commands** in the following format:

- **Add** `{passengers}` – add a wagon to the end with the given number of passengers.
- `{passengers}` - find an existing wagon to **fit all the passengers** (**starting from the first wagon**)

At the end **print the final state** of the train (all the wagons **separated** by a space)

Example

Input	Output
['32 54 21 12 4 0 23', '75', 'Add 10', 'Add 0', '30', '10', '75']	72 54 21 12 4 75 23 10 0
['0 0 0 10 2 4', '10', 'Add 10', '10', '10', '10', '8', '6']	10 10 10 10 10 10 10

2. Distinct Array

You will be given an **array of integer numbers** on the first line of the input (**space-separated**).

Remove all **repeating elements** from the array.

Print the result elements **separated** by single space.

Examples

Input	Output	Comments
[1, 2, 3, 4]	1 2 3 4	No repeating elements
[7, 8, 9, 7, 2, 3, 4, 1, 2]	7 8 9 2 3 4 1	7 and 2 are already present in the array ➔remove them

[20, 8, 12, 13, 4, 4, 8, 5]	20 8 12 13 4 5	4 and 8 are already present in the array ➔remove them

3. House Party

Write a function that keeps track of **guests** that are going to a house party.

You will be given an **array of strings**. Each string will be one of the following:

- **"{name} is going!"**
- **"{name} is not going!"**

If you receive the **first type of input**, you have to **add** the person if he/she **is not** in the list (If he/she is in the list print: **"{name} is already in the list!"**).

If you receive the **second type of input**, you have to **remove** the person if he/she **is** in the list (if not print: **"{name} is not in the list!"**).

At the end print all the guests each on a **separate line**.

Examples

Input	Output
['Allie is going!', 'George is going!', 'John is not going!', 'George is not going!']	John is not in the list! Allie
['Tom is going!', 'Annie is going!', 'Tom is going!', 'Garry is going!', 'Jerry is going!']	Tom is already in the list! Tom Annie Garry Jerry

4. Sorting

Write a function that sorts an **array of numbers** so that the first element is the **biggest** one, the second is the **smallest** one, the third is the **second biggest** one, the fourth is the **second smallest** one and so on.

Print the elements on one row, **separated** by single space.

Examples

Input	Output
[1, 21, 3, 52, 69, 63, 31, 2, 18, 94]	94 1 69 2 63 3 52 18 31 21

5. Sort an Array by 2 Criteria

Write a function that orders an **array of strings**, by their **length** in **ascending order** as **primary criteria**, and by **alphabetical value** in **ascending order** as **second criteria**. The comparison should be **case-insensitive**.

The **input** comes as **array of strings**.

The **output** is the **ordered** array of strings.

Examples

Input	Output

`["alpha", "beta", "gamma"]`	`beta`
	`alpha`
	`gamma`

Input	Output

`["Isacc", "Theodor", "Jack", "Harrison", "George"]`	`Jack`
	`Isacc`
	`George`
	`Theodor`
	`Harrison`

Hints

- An array can be **sorted** by passing a comparing function to the **`Array.sort()`** function
- Creating a comparing function by 2 criteria can be achieved by first comparing by the **main criteria**, if the 2 items are different (the result of the compare is not 0) - return the result as the result of the comparing function. If the two items are the same by the **main criteria** (the result of the compare is 0), we need to compare by the **second criteria** and the result of that comparison is the result of the comparing function