# Exercise: Arrays

Problems for exercises and homework for the ["JavaScript Advanced" course @ SoftUni](https://softuni.bg/trainings/3588/js-advanced-january-2022). Submit your solutions in the SoftUni judge system <https://judge.softuni.bg/Contests/2753/Arrays-and-Nested-Arrays-Exercise>.

## Print an Array with a Given Delimiter

The **input** comes as two parameters – an **array of strings** and a **string**. The second parameter is the delimiter.

The **output** is the elements of the array, printed on the console, each element **separated** from the others by the **given delimiter**.

### Examples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| **['One',**  **'Two',**  **'Three',**  **'Four',**  **'Five'],**  **'-'** | **One-Two-Three-Four-Five** |  | **['How about no?',**  **'I',**  **'will',**  **'not',**  **'do',**  **'it!'],**  **'\_'** | **How about no?\_I\_will\_not\_do\_it!** |

## Print Every N-th Element from an Array

The **input** comes as two parameters – an **array of strings** and a **number**. The second parameter is **N** – **the step**.

The **output** is every element on the **N-th** step **starting from the first one**. If the step is 3, you need to return the **1-st**, the **4-th**, the **7-th** … and so on, until you reach the end of the array.

The **output** is the **return** value of your function and must be an **array**.

### Example

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| **['5',**  **'20',**  **'31',**  **'4',**  **'20'],**  **2** | **['5', '31', '20']** |  | **['dsa',**  **'asd',**  **'test',**  **'tset'],**  **2** | **['dsa', 'test']** | **['1',**  **'2',**  **'3',**  **'4',**  **'5'],**  **6** | ['1'] |

### Hints

* **Return all the elements** with for loop, **incrementing** the **loop variable** with the value of the step variable.

## Add and Remove Elements

Write a JS function that **adds** and **removes** numbers **to/from** an array. You will receive a command which can either be "add" or "remove".

The **initial number** is **1**. Each input command should **increase that number**, regardless of what it is.  
Upon receiving an "add" command you should add the current number to your array.   
Upon receiving the "remove" command you should remove the last entered number, currently existent in the array.

The **input** comes as an **array of strings**. Each element holds a **command**.

The **output** is the element of the array, each printed on a new line. In case of an empty array, just print "Empty".

### Examples

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| **['add',**  **'add',**  **'add',**  **'add']** | **1**  **2**  **3**  **4** |  | **['add',**  **'add',**  **'remove',**  **'add',**  **'add']** | **1**  **4**  **5** | **['remove',**  **'remove',**  **'remove']** | **Empty** |

## Rotate Array

Write a JS function that rotates an array. The array should be rotated **to the right side**, meaning that the last element should become the first, upon rotation.

The **input** comes as two parameters – an **array of strings** and a **number**. The **second parameter** is the amount of rotation you need to perform.

The **output** is the resulting array after the rotations. The elements should be printed on one line, separated by a **single space**.

### Examples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| **['1',**  **'2',**  **'3',**  **'4'],**  **2** | **3 4 1 2** |  | **['Banana',**  **'Orange',**  **'Coconut',**  **'Apple'],**  **15** | **Orange Coconut Apple Banana** |

### Hints

* Check if there is a **built-in function** for inserting elements **at the** **start** of the array.

## Extract Increasing Subset from Array

Write a function that extracts only those numbers that **form a** **non-decreasing subset**. In other words, you start from the **first element** and continue to **the end** of the **given array of numbers**. Any number which is **LESS THAN** the **current biggest one** is **ignored**, alternatively if it’s equal or higher than the **current biggest one** you set it as the **current biggest one** and you continue to the next number.

The **input** comes as an **array of numbers**.

The **output** is the processed array after the filtration, which should be a non-decreasing subset. Return the **array of numbers**.

### Examples

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| **[1,**  **3,**  **8,**  **4,**  **10,**  **12,**  **3,**  **2,**  **24]** | **[1, 3, 8, 10, 12, 24]** |  | **[1,**  **2,**  **3,**  **4]** | **[1, 2, 3, 4]** | **[20,**  **3,**  **2,**  **15,**  **6,**  **1]** | **[20]** |

### Hints

* The Array.reduce() built-in function might help you a lot with this problem.

## List of Names

You will receive an **array of names**. Sort them **alphabetically in ascending order** and print a numbered list of all the names, each on a new line.

**Example**

|  |  |
| --- | --- |
| **Input** | **Output** |
| **["John", "Bob", "Christina", "Ema"]** | **1.Bob**  **2.Christina**  **3.Ema**  **4.John** |

**Hints**

* The **sort function** rearranges the array in ascending order

## Sorting Numbers

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

**Return** the resulting array.

**Example**

|  |  |
| --- | --- |
| **Input** | **Output** |
| **[1, 65, 3, 52, 48, 63, 31, -3, 18, 56]** | **[-3, 65, 1, 63, 3, 56, 18, 52, 31, 48]** |

## Sort an Array by 2 Criteria

Write a function that orders a **given array of strings**, by a **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 an **array of strings**.

The **output** is the elements of the ordered array of strings, printed each on a new line.

### Examples

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| **['alpha',**  **'beta',**  **'gamma']** | **beta**  **alpha**  **gamma** |  | **['Isacc',**  **'Theodor',**  **'Jack',**  **'Harrison',**  **'George']** | **Jack**  **Isacc**  **George**  **Theodor**  **Harrison** |  | **Deny**  **omen**  **test**  **Default** |

### 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 comparison is 0), we need to compare by the **second criteria** and the result of that comparison is the result of the comparing function.
* You can check more about Array.sort() here - [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/Array/sort](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/sort%20)