# JS Arrays and Strings – Exercises

Problems with exercise and homework for the ["JS Front-End" Course @ SoftUni.](https://softuni.bg/modules/132)

## Array Rotation

Write a function that receives an **array** and the **number of rotations** you have to perform.

Note: Depending on the number of rotations, the first element goes to the end.

### Output

Print the resulting arrayelementsseparated by a single space.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| [51, 47, 32, 61, 21], 2 | 32 61 21 51 47 |
| [32, 21, 61, 1], 4 | 32 21 61 1 |
| [2, 4, 15, 31], 5 | 4 15 31 2 |

## 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.

## 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] |

## Reveal Words

Write a function, which receives **two parameters**.

The first parameter will be a string with some words **separated by ', '**.

The second parameter will be a string that contains **templates containing '**\***'**.

Find the word with the **same length** as the template and **replace** it.

**Example**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 'great',  'softuni is \*\*\*\*\* place for learning new programming languages' | softuni is great place for learning new programming languages |
| 'great, learning',  'softuni is \*\*\*\*\* place for \*\*\*\*\*\*\*\* new programming languages' | softuni is great place for learning new programming languages |

## Modern Times of #(HashTag)

The input will be a **single string.**

**Find all** special words **starting with #**. If the found special word does not consist only of letters, then it is invalid and should not be printed.

Finally, print out all the special words you found without the label **(#)** on a new line.

**Example**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 'Nowadays everyone uses # to tag a **#special** word in **#socialMedia**' | special  socialMedia |
| 'The symbol # is known **#variously** in English-speaking **#regions** as the #**number** sign' | variously  regions  number |

1. **String Substring**

The input will be given as **two** separated strings(a **word** as a first parameter and a **text** as a second).

Write a function that checks given text for containing a given word. The comparison should be **case insensitive.** Once you find a match, **print** the word and **stop** the program.

If you don't find the word print: **"{word} not found!"**

**Example**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 'javascript', 'JavaScript is the best programming language' | javascript |
| 'python',  'JavaScript is the best programming language' | python not found! |

## Pascal-Case Splitter

You will receive a **single** **string**.

This string is written in **PascalCase** format. Your task here is to split this string by **every word** in it.

Print them joined by **comma** and **space.**

**Examples**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 'SplitMeIfYouCanHaHaYouCantOrYouCan' | Split, Me, If, You, Can, Ha, Ha, You, Cant, Or, You, Can |
| 'HoldTheDoor' | Hold, The, Door |
| 'ThisIsSoAnnoyingToDo' | This, Is, So, Annoying, To, Do |