

# Exercises: Arrays and Matrices

## 1. Print an Array with a Given Delimiter

The **input** comes as an **array of strings**. The last element of the array is the delimiter.

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

### Examples

Input	Output
['One', 'Two', 'Three', 'Four', 'Five', '-']	One-Two-Three- Four-Five

Input	Output
['How about no?', 'I', 'will', 'not', 'do', 'it!', '_']	How about no?_I_will_not_do_it!

### What to submit?

Function Signature: `function main(strings)`

## 2. Print Every N-th Element from an Array

The **input** comes as an **array of strings**. The last element 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 print the **1-st**, the **4-th**, the **7-th** ... and so on, until you reach the end of the array. Each element must be printed on a new line.

### Example

Input	Output
['5', '20', '31', '4', '20', '2']	5 31 20

Input	Output
['dsa', 'asd', 'test', 'tset', '2']	dsa test

Input	Output
['1', '2', '3', '4', '5', '6']	1

### Hints

- Use what you've seen from the **previous problem** to **extract the last element** of the array.
- Create a **step** variable to hold the **given step** of the array. Then **print all the elements** with a **for** loop, **incrementing the loop variable** with the value of the **step** variable.

## What to submit?

Function Signature: `function main(strings)`

### 3. Add and Remove Elements from an Array

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 array itself, with each element 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

## What to submit?

Function Signature: `function main(commands)`

### 4. 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 an **array of strings**. The **last element** of the array is the number of rotations you need to make.

The **output** is the resultant 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 1 2	['Banana', 'Orange',	Orange Coconut Apple Banana

'3', '4', '2']	
----------------------	--

'Coconut', 'Apple', '15']	
---------------------------------	--

## Hints

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

## What to submit?

Function Signature: `function main(strings)`

## 5. Extract Increasing Subsequence from an Array

Write a function that extracts only those numbers that **form a non-decreasing subsequence**. 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 subsequence. Each element should be printed on a new line.

## Examples

Input	Output
[1, 3, 8, 4, 10, 12, 3, 2, 24]	1 3 8 10 12 24

Input	Output
[1, 2, 3, 4]	1 2 3 4

Input	Output
[20, 3, 2, 15, 6, 1]	20

## Hints

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

## What to submit?

Function Signature: `function main(numbers)`