1. You will be given an array of drinks, with each drink being an object with two properties: name and price. Create a function that has the drinks array as an argument and return the drinks objects sorted by price in ascending order.

Assume that the following array of drink objects needs to be sorted:

drinks = [

{name: "lemonade", price: 50},

{name: "lime", price: 10}

]

The output of the sorted drinks object will be:

sortDrinkByPrice(drinks) ➞ [{name: "lime", price: 10}, {name: "lemonade", price: 50}]

2. A word is on the loose and now has tried to hide amongst a crowd of tall letters! Help write a function to detect what the word is, knowing the following rules:

* The wanted word is in **lowercase**.
* The crowd of letters is all in **uppercase**.
* Note that the word will be spread out amongst the random letters, but their letters **remain in the same order**.

### Examples

detectWord("UcUNFYGaFYFYGtNUH") ➞ "cat"

detectWord("bEEFGBuFBRrHgUHlNFYaYr") ➞ "burglar"

detectWord("YFemHUFBbezFBYzFBYLleGBYEFGBMENTment") ➞ "embezzlement"

3. In this challenge you will be given a relation between two numbers, written as a string. Write a function that determines if the relation is true or false.

### Examples

isTrue("2=2") ➞ true

isTrue("8<7") ➞ false

isTrue("5=13") ➞ false

isTrue("15>4") ➞ true

### Notes

* Tests will only have three types of relations: =, >, and <
* Many approaches work here, but the eval() function is particularly useful!

4. Given an object of how many more pages each ink color can print, output the maximum number of pages the printer can print before any of the colors run out.

### Examples

inkLevels({

"cyan": 23,

"magenta": 12,

"yellow": 10

}) ➞ 10

inkLevels({

"cyan": 432,

"magenta": 543,

"yellow": 777

}) ➞ 432

inkLevels({

"cyan": 700,

"magenta": 700,

"yellow": 0

}) ➞ 0

### Notes

A single printed page requires each color once, so printing is not possible if any of the slots lack ink (see example #3).

5. A **set** is a collection of unique items. A **set** can be formed from an array from removing all duplicate items.

[1, 3, 3, 5, 5, 5]

// original array

[1, 3, 5]

// original array transformed into a set

Create a function that sorts an array and removes all duplicate items from it.

### Examples

set([1, 3, 3, 5, 5]) ➞ [1, 3, 5]

set([4, 4, 4, 4]) ➞ [4]

set([5, 7, 8, 9, 10, 15]) ➞ [5, 7, 8, 9, 10, 15]

set([3, 3, 3, 2, 1]) ➞ [1, 2, 3]

### Notes

* For this question, output **an array, not a set**. These are two distinctly different data structures in JavaScript (although they can be converted from one to the other).