Find Number of Digits in Number

Create a function that will return an integer number corresponding to the amount of digits in the given integer num.

```
num_of_digits(1000) → 4
```

Concatenate Variable Number of Input Arrays

Create a function that concatenates n input arrays, where n is variable.

```
concat([1, 2, 3], [4, 5], [6, 7]) → [1, 2, 3, 4, 5, 6, 7]
```

Converting Objects to Arrays

Write a function that converts an object into an array, where each element represents a key-value pair in the form of an array.

```
toArray({ a: 1, b: 2 }) → [["a", 1], ["b", 2]]
```

Array of Multiples

Create a function that takes two numbers as arguments (num, length) and returns an array of multiples of num until the array length reaches length.

```
arrayOfMultiples(7, 5) → [7, 14, 21, 28, 35]
```

Return the Objects Keys and Values

Create a function that takes an object and returns the keys and values as separate arrays. Return the keys sorted alphabetically, and their corresponding values in the same order.

```
keysAndValues({ a: 1, b: 2, c: 3 })
→ [["a", "b", "c"], [1, 2, 3]]
```

Reverse Words in a String

Given an input string, reverse the string word by word, the first word will be the last, and so on.

```
reverseWords(" the sky is blue") → "blue is sky the"
```

Integer in Range?

Create a function that validates whether a number n is within the bounds of lower and upper. Return false if n is not an integer.

intWithinBounds(3, 1, 9) → true

Remove Trailing and Leading Zeros

Create a function that takes in a number as a string n and returns the number without trailing and leading zeros.

- Trailing Zeros are the zeros after a decimal point which don't affect the value (e.g. the last three zeros in 3.4000 and 3.04000).
- Leading Zeros are the zeros before a whole number which don't affect the value (e.g. the first three zeros in 000234 and 000230).

```
removeLeadingTrailing("230.000") → "230"
```

Fix the Error: Value vs. Reference Types

Create a function that returns true if two arrays contain identical values, and false otherwise.

```
checkEquals([1, 2], [1, 3]) → false

checkEquals([1, 2], [1, 2]) → true
```

Flattening an Array

Write a function to flatten an array of subarrays into one array.

```
flatten([["a", "b"], ["c", "d"]]) → ["a", "b", "c", "d"]
```

Find the Second Largest Number

Create a function that takes an array of numbers and returns the second largest number.

```
secondLargest([10, 40, 30, 20, 50]) → 40
```

Basic Calculator

Create a function that takes two numbers and a mathematical operator + - / * and will perform a calculation with the given numbers.

```
calculator(2, "+", 2) → 4

calculator(2, "*", 2) → 4

calculator(4, "/", 2) → 2
```

Check if All Values Are True

Create a function that returns true if all parameters are truthy, and false otherwise.

```
allTruthy(true, true, true) → true

allTruthy(true, false, true) → false
```

Index Multiplier

Return the sum of all items in an array, where each item is multiplied by its index (zero-based). For empty arrays, return 0.

```
indexMultiplier([1, 2, 3, 4, 5]) → 40
```

Let's Sort This Array!

Create a function that takes an array of numbers arr, a string str and return an array of numbers as per the following rules:

- "Asc" returns a sorted array in ascending order.
- "Des" returns a sorted array in descending order.
- "None" returns an array without any modification.

```
ascDesNone([4, 3, 2, 1], "Asc" ) → [1, 2, 3, 4]
```

Mirror Array

Given an integer array, transform that array into a mirror.

```
mirror([0, 2, 4, 6]) → [0, 2, 4, 6, 4, 2, 0]

mirror([1, 2, 3, 4, 5]) → [1, 2, 3, 4, 5, 4, 3, 2, 1]
```

Reverse the Number

Create a function that takes an integer n and reverses it.

rev(5121) → "1215"

Match the Last Item

Create a function that takes an array of items and checks if the last item matches the rest of the array concatenated together.

```
matchLastItem(["rsq", "6hi", "g", "rsq6hig"]) → true
```

Remove Duplicates from an Array

Create a function that takes an array of items, removes all duplicate items and returns a new array in the same sequential order as the old array (minus duplicates).

```
removeDups(["John", "Taylor", "John"]) → ["John", "Taylor"]
```

Find the Missing Number

Create a function that takes an array of numbers between 1 and a max number and returns the missing numbers.

missingNum([1, 2, 3, 4, 6, 7, 8, 9, 10]) → 5

Seven Boom!

Create a function that takes an array of numbers and return "Boom!" if the digit 7 appears in the array. Otherwise, return "there is no 7 in the array".

```
sevenBoom([1, 2, 3, 4, 5, 6, 7]) → "Boom!"

sevenBoom([8, 6, 33, 100]) → "there is no 7 in the array"
```

How Many Days Between Two Dates

Create a function that takes two dates and returns the number of days between the first and second date.

```
getDays(
  new Date("June 14, 2019"),
  new Date("June 20, 2019")
) → 6
```

Length of a Nested Array

The .length property on an array will return the number of elements in the array. For example, the array below contains 2 elements: [1, [2, 3]] // 2 elements

Write a function that returns the total number of non-nested items in a nested array.

```
getLength([1, [2, 3]]) → 3

getLength([1, [2, [3, 4]]]) → 4

getLength([1, [2, [3, [4, [5, 6]]]]]) → 6
```

Finding Common Elements

Create a function that takes two "sorted" arrays of numbers and returns an array of numbers which are common to both the input arrays.

```
commonElements([-1, 3, 4, 6, 7, 9], [1, 3]) → [3]

commonElements([1, 3, 4, 6, 7, 9], [1, 2, 3, 4, 7, 10]) → [1, 3, 4, 7]
```