JS Practice Problems - 2

1. **Fizz Buzz Implementation :**

   Get an integer n. Print numbers from 1 to n with the condition that

   - **if the number is divisible by 3 and 5 print “FizzBuzz”**
   - **if the number is divisible by 3 print “Fizz”**
   - **if the number is divisible by 5 print “Buzz”**
   - **Otherwise print the number itself.**

TestCases :

i. **input : n = 3**
   Output : 1 2 Fizz

ii. **Input : n = 5**
   Output : 1 2 Fizz 4 Buzz

iii. ***Input : n = 20***
   Output : 1 2 Fizz 4 Buzz Fizz 7 8 Fizz Buzz 11 Fizz 13 14 FizzBuzz 16 17 Fizz 19 Buzz

2.

Array of Multiples

Published by **Raka Raka** in **JavaScript**

arrays loops math numbers

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`.

Examples

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

arrayOfMultiples(12, 10) → [12, 24, 36, 48, 60, 72, 84, 96, 108, 120]

arrayOfMultiples(17, 6) → [17, 34, 51, 68, 85, 102]
```

Notes

Notice that `num` is also included in the returned array.

3.

Positive Count / Negative Sum

Published by Matt in JavaScript

arrays conditions numbers

Create a function that takes an array of positive and negative numbers. Return an array where the first element is the **count** of positive numbers and the second element is the **sum** of negative numbers.

Examples

```
countPosSumNeg([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, -11, -12, -13, -14, -15])
// There are a total of 10 positive numbers.
// The sum of all negative numbers equals -65.

countPosSumNeg([92, 6, 73, -77, 81, -90, 99, 8, -85, 34]) → [7, -252]

countPosSumNeg([91, -4, 80, -73, -28]) → [2, -105]

countPosSumNeg([]) → []
```

Notes

- If given an empty array, return an empty array: `[]`
- 0 is not positive.

4.

Find the Missing Number

Published by Matt in JavaScript

algorithms interview math numbers sorting

Create a function that takes an array of numbers between 1 and 10 (excluding one number) and returns the missing number.

Examples

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

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

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

Notes

- The array of numbers will be **unsorted** (not in order).
- Only one number will be missing.

5.

Return the Highest and Lowest Numbers

Published by Matt in JavaScript

algorithms math numbers sorting

Create a function that accepts a string of space separated numbers and returns the highest and lowest number (as a string).

Examples

```
highLow("1 2 3 4 5") → "5 1"

highLow("1 2 -3 4 5") → "5 -3"

highLow("1 9 3 4 -5") → "9 -5"

highLow("13") → "13 13"
```

Notes

- All numbers are valid Int32, no need to validate them.
- There will always be at least one number in the input string.
- Output string must be two numbers separated by a single space, and highest number is first.