

**Try to solve all assignments on a separate branch of bit-pp repository. Don't forget to first pull changes done at class and push the new repository to remote when done with all the work at the end.**

1. Write a program to insert a string within a string at a particular position (default is 1, beginning of a string).

```
"My random string", "JS " -> "JS My random string"
"My random string", "JS ", 10 -> "My random JS string"
```



2. Write a program to join all elements of the array into a string skipping elements that are undefined, null, NaN or Infinity.

```
[NaN, 0, 15, false, -22, '', undefined, 47, null]
```



3. Write a program to filter out *falsy* values from the array .

```
[NaN, 0, 15, false, -22, '', undefined, 47, null] -> [15, -22, 47]
```



4. Write a function that reverses a number. The result must be a number.

```
12345 -> 54321 // Output must be a number
```

```
function reverseNumber(num) {
  num = num + "";
  var output = "";
  var lastIndex = num.length - 1;

  for (var index = 0; index < num.length; index++) {
    var element = num[lastIndex - index];
    output += element;
  }

  var reversedNum = parseFloat(output);
  return reversedNum;
}

var result = reverseNumber(98765)

console.log(typeof result);
console.log(result);
```

5. Write a function to get the last element of an array. Passing a parameter 'n' will return the last 'n' elements of the array.

```
[7, 9, 0, -2] -> -2  
[7, 9, 0, -2], 2 -> [0, -2]
```

6. Write a function to create a specified number of elements with pre-filled numeric value array.

```
6, 0 -> [0, 0, 0, 0, 0, 0]  
2, "none" -> ["none", "none"]  
2 -> [null, null]
```

7. Write a function that says whether a number is perfect.

```
28 -> 28 is a perfect number.
```

Note: According to Wikipedia: In number theory, a perfect number is a positive integer that is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors excluding the number itself (also known as its aliquot sum). Equivalently, a perfect number is a number that is half the sum of all of its positive divisors (including itself).

E.g.: The first perfect number is 6, because 1, 2 and 3 are its proper positive divisors, and  $1 + 2 + 3 = 6$ . Equivalently, the number 6 is equal to half the sum of all its positive divisors:  $(1 + 2 + 3 + 6) / 2 = 6$ . The next perfect number is  $28 = 1 + 2 + 4 + 7 + 14$ . This is followed by the perfect numbers 496 and 8128.

```
function isPerfect(number) {  
    var temp = 0;  
    for (var i = 1; i <= number / 2; i++) {  
        if (number % i === 0) {  
            temp += i;  
        }  
    }  
  
    if (temp === number && temp !== 0) {  
        console.log(number + " is a perfect number.");  
    } else {  
        console.log(num + " is not a perfect number.");  
    }  
}  
  
isPerfect(28);
```

8. Write a function to find a word within a string.

'The quick brown fox', 'fox' -> "'fox' was found 1 times"  
'aa, bb, cc, dd, aa', 'aa' -> "'aa' was found 2 times"

```
function searchWordCount(text, word) {  
  
    var x = 0;  
    var y = 0;  
  
    for (i = 0; i < text.length; i++) {  
        if (text[i] == word[0]) {  
            for (j = i; j < i + word.length; j++) {  
                if (text[j] == word[j - i]) {  
                    y++;  
                }  
                if (y == word.length) {  
                    x++;  
                }  
            }  
            y = 0;  
        }  
    }  
    return "'" + word + "' was found " + x + " times.";  
}  
  
console.log(searchWordCount('The quick brown fox', 'fox'));  
console.log(searchWordCount('aa, bb, cc, dd, aa', 'aa'));
```

9. Write a function to hide email address.

"myemailaddress@bgit.rs" -> "myemail...@bgit.rs"

10. Write a program to find the most frequent item of an array.

[3, 'a', 'a', 'a', 2, 3, 'a', 3, 'a', 2, 4, 9, 3]

```
function findFrequentItem(inputArr) {  
    var mostFrequent = 1;  
    var m = 0;  
    var item;  
    for (var i = 0; i < inputArr.length; i++) {  
        for (var j = i; j < inputArr.length; j++) {  
            if (inputArr[i] == inputArr[j]) {  
                m++;  
            }  
            if (mostFrequent < m) {  
                mostFrequent = m;  
            }  
        }  
    }  
}
```

```
        item = inputArr[i];
    }
}
m = 0;
}

// output: item (number of times)
return item + " ( " + mostFrequent + " times ) "
}

var arr1 = [3, 'a', 'a', 'a', 2, 3, 'a', 3, 'a', 2, 4, 9, 3];
console.log(findFrequentItem(arr1));
```