

Палиндромом

Написать функцию, которая проверяет, является ли фраза палиндромом
В решении использовать два указателя

```
function isPalindrome(input) {}  
console.log("роза", isPalindrome("А роза упала на лапу Азора!")); // true  
console.log("шалаш", isPalindrome("Ш4л4ш")); // true  
console.log("Eva", isPalindrome("Eva, can I see bees in a cave?")); // true
```

Поменять чётные с нечётными символы в строке местами

```
function pairRevert(str) {}  
console.log(pairRevert('apple') = 'palpe');  
console.log(pairRevert('foobar') = 'ofbora');
```

В массиве nums из целых чисел найти позиции двух чисел сумма которых будет равна target

```
function twoSum(nums, target) {}  
const nums = [11, 2, 17, 32, 7, 15];  
const target = 9;  
console.log(twoSum(nums, target)) // [1, 4]
```

Написать функцию сжатия RLE

```
function RLE(string) {}  
console.log(RLE('AAAABBBCCXYZDDDDDEEEFFFAAAAAABBBBBBBBBBBBBBBBBBBBBBBBBBBB  
B') === 'A4B3C2XYZD4E3F3A6B28')
```

Сжатие массива

```
function compressArray(array) {}  
console.log(compressArray([3, 2, 1, 5, 6, -1, 10])); // "-1,1-3,5-6,10"
```

Аналог join

```
function join(sep, ...args) {}  
console.log(join('*', 1, 2, 3, 4, 5, 6)) // "1*2*3*4*5*6"
```

Сумма чисел массива

```
function summaryArray(arr) {}  
console.log(summaryArray([1, [1, 2], 2])); //6
```

```
console.log(summaryArray([1, 2, "3x"])); //6
console.log(summaryArray([1, 2, "x3"])); //3
```

Проверка массива на монотонность

```
function isMonotone(arr) {}
console.log(isMonotone([0, 1, 5, 9, 15])); //true
console.log(isMonotone([0, 1, 1, 5, 9, 9, 15])); //true
console.log(isMonotone([15, 8, 4, 2, 1])); //true
console.log(isMonotone([1, 1, 1, 1])); //true
console.log(isMonotone([0, 1, 5, 15, 4])); //false
console.log(isMonotone([15, 8, 4, 2, 1, 3])); //false
```

Обращение к свойству объекта

```
const o = {
  a: {
    b: {
      f: 7
    },
    c: 42
  },
  z: null
};
function getItem(object, path) {}
console.log(getResult(o, "a.b.f")); //7
console.log(getResult(o, "a.d")); //undefined
```

Получить все простые числа до заданного числа

```
function getPrimes(count) {}
console.log(getPrimes(199));
```

Правильная скобочная последовательность

```
function correctBreakSequence(string) {}
console.log(correctBreakSequence("{}{}[]")); //true
console.log(correctBreakSequence("{}{}[]]]]]")); //false
```

Сортировка по анаграммам

Написать функцию getAnagrams

```
getAnagrams("нос", "сон", "снеть", "днесь")
```

Должна вернуть

```
[ ["нос", "сон"], ["днесь", "снedy"] ]
```

```
function getAnagrams(...strings) {}  
console.log(getAnagrams("нос", "сон", "снedy", "днесь"));
```

Посчитать сумму значений объекта

```
let o = {  
  value: 4,  
  next: [  
    {  
      value: 3,  
      next: [  
        {  
          value: 5,  
          next: null  
        },  
        {  
          value: 6,  
          next: null  
        }  
      ]  
    },  
    {  
      value: 2,  
      next: [  
        {  
          value: 2,  
          next: null  
        },  
        {  
          value: 3,  
          next: [  
            {  
              value: 7,  
              next: null  
            },  
            {  
              value: 8,  
              next: null  
            }  
          ]  
        }  
      ]  
    }  
  ]  
}
```

```
function getSum(object) {}  
console.log(getSum(o) === 40);
```

Из строки - вложенный объект

```
function strToObj(str) {}  
console.log(strToObj("a.b.c")); // a: { b: { c: {}}}
```

Форматирование массива в объект

```
// С бекенда приходит массив такого вида:  
var arr = [ { name: 'width', value: 10 }, { name: 'height', value: 20 }, // ...];  
// Нужно получить объект такого вида:  
{  
  width: 10,  
  height: 20,  
  ...  
}  
function objFromArr(arr) {}
```

Задача на классы

Реализовать класс Car с методами speed, color, doors, вызов которых будет работать так, как в примере ниже:

```
var car = new Car('Lada X-Ray');  
car.speed(100)  
  .color('blue')  
  .doors(5);  
  
car.speed() // '100 km/h'  
car.color() // 'blue'  
car.doors() // 5 doors
```

Подсчет символа в строке

Написать функцию, которая принимает на вход 2 строковых аргумента, подсчитывает количество вхождений строки из второго аргумента в строку из первого аргумента.

- первый аргумент может быть пустой строкой
- второй аргумент может быть только строкой из одного символа
- нельзя использовать регулярные выражения

```
function strCount(str, char) {}  
strCount('Hello', 'o') // => 1  
strCount('Hello', 'l') // => 2
```