**Задание 1**

Используя методы для работы с массивами получить требуемый результат.

|  |
| --- |
| let nums = [1, 2, 3, 5, 8, 13, 21, 34];  let str = ["this", "is", "a", "very", "long", "array", "which", "has", "absolutely", "no", "sense"]; |

|  |  |
| --- | --- |
| Исходный массив | Ожидаемый результат (в скобках указан тип ожидаемого значения) |
| nums | Сумма удвоенных значений каждого элемента  (Number) |
| nums | Узнать, есть ли в массиве четные числа  (Boolean) |
| str | Соединить элементы массива с одну строку, где слова разделены пробелами  (String) |
| str | Получить новый массив, в котором к каждому элементу (строке) будет добавлено число, равное количеству символов в этой строке. Например, [“this - 4”, “is - 2” ....]  (Array) |
| str | Получить новый массив, в котором все элементы (строки) содержат 4 или более символов  (Array) |
| nums | Получить массив, который будет содержать только нечетные числа  (Array) |
| nums | Сообщить, является ли сумма всех элементов больше 100  (Boolean) |
| str | Получить новый массив, в котором все элементы будут отсортированы по количеству символов в строке по возрастанию  (Array) |
| str | Найти индекс самого длинного слова в массиве  (Number) |
| nums, str | Получить строку, которая будет содержать все элементы двух массивов перечисленных через запятую  (String) |

**Задание 2**

Реализовать функцию-счетчик.

Используя prompt() просим пользователя ввести число, которым инициализируется (начинается) счетчик.

Вторым prompt() просим указать шаг отсчета (число, на которое счетчик будет увеличивать свое значение)

Третьим - пользователь вводит количество повторений (вызовов) счетчика.

В итоге выводи alert, в котором сообщаем об итоговом значении счетчика после указанного количества повторений.

**Задание 3**

Написать функцию, которая будет принимать строку (только буквы латинского алфавита) любой длины и возвращать ее, но удалив из нее все гласные буквы английского алфавита. (их всего 6: «A», «E», «I», «O», «U», «Y»)

**Задание 4**

Написать функцию, которая будет принимать произвольное количество аргументов и возвращать массив строк. Каждая строка - это тип аргумента, возвращаемый оператором typeof. Например, f(null) - > [“object”]