**Лабораторная работа №10. «JavaScript».**

**Рекомендации сдаче ЛР**

- рекомендуется использовать в качестве среды PhpStorm, Webstorm, так средство разработки будет исправлять код и давать рекомендацию. Для студентов можно получить бесплатные лицензии 1 год. (https://www.jetbrains.com/ru-ru/community/education/#students)

**-** использовать для оформления ГОСТ университета

**Требования к отчету:**

* Титульный лист
* ведение
* цели и задачи
* решение задач
* вывод

**Задачи:**

1. Напишите функцию, которая найдёт числа в массиве, которые делятся на заданное число[​](https://www.jscamp.app/ru/docs/javascript30/#%D0%BD%D0%B0%D0%B9%D0%B4%D0%B8%D1%82%D0%B5-%D1%87%D0%B8%D1%81%D0%BB%D0%B0-%D0%BA%D0%BE%D1%82%D0%BE%D1%80%D1%8B%D0%B5-%D0%B4%D0%B5%D0%BB%D1%8F%D1%82%D1%81%D1%8F-%D0%BD%D0%B0-%D0%B7%D0%B0%D0%B4%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5-%D1%87%D0%B8%D1%81%D0%BB%D0%BE).
2. Нужно написать функцию, которая проверяет, являются ли две строки анаграммами, причем регистр букв не имеет значения. Учитываются лишь символы; пробелы или знаки препинания в расчет не берутся.
3. Нужно написать функцию, принимающую строку в качестве аргумента и возвращающую количество гласных, которые содержатся в строке.

Гласными являются «a», «e», «i», «o», «u».

1. Треугольник. Напишите цикл,  выводит такой треугольник:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | #  ##  ###  ####  #####  ######  ####### |

1. FizzBuzz. Напишите программу, которая выводит через console.log все числа от 1 до 100, с двумя исключениями. Для чисел, нацело делящихся на 3, она должна выводить ‘Fizz’, а для чисел, делящихся на 5 (но не на 3) – ‘Buzz’.Когда сумеете – исправьте её так, чтобы она выводила «FizzBuzz» для всех чисел, которые делятся и на 3 и на 5.
2. Шахматная доска. Напишите программу, создающую строку, содержащую решётку 8х8, в которой линии разделяются символами новой строки. На каждой позиции либо пробел, либо #.
3. Минимум. Напишите функцию min, принимающую два аргумента, и возвращающую минимальный из них.
4. Рекурсия. Ноль чётный. Единица нечётная. У любого числа N чётность такая же, как у N-2. Напишите рекурсивную функцию isEven согласно этим правилам. Она должна принимать число и возвращать булевское значение. Потестируйте её на 50 и 75. Попробуйте задать ей -1. Почему она ведёт себя таким образом? Можно ли её как-то исправить?
5. Считаем бобы. Символ номер N строки можно получить, добавив к ней .charAt(N) ( “строчка”.charAt(5) ) – схожим образом с получением длины строки при помощи .length. Возвращаемое значение будет строковым, состоящим из одного символа (к примеру, “к”). У первого символа строки позиция 0, что означает, что у последнего символа позиция будет string.length – 1. Другими словами, у строки из двух символов длина 2, а позиции её символов будут 0 и 1.Напишите функцию countBs, которая принимает строку в качестве аргумента, и возвращает количество символов “B”, содержащихся в строке.Затем напишите функцию countChar, которая работает примерно как countBs, только принимает второй параметр — символ, который мы будем искать в строке (вместо того, чтобы просто считать количество символов “B”). Для этого переделайте функцию countBs.
6. Сумма диапазона. Напишите функцию range, принимающую два аргумента, начало и конец диапазона, и возвращающую массив, который содержит все числа из него, включая начальное и конечное.Затем напишите функцию sum, принимающую массив чисел и возвращающую их сумму. Запустите указанную выше инструкцию и убедитесь, что она возвращает 55.В качестве бонуса дополните функцию range, чтобы она могла принимать необязательный третий аргумент – шаг для построения массива. Если он не задан, шаг равен единице. Вызов функции range(1, 10, 2) должен будет вернуть [1, 3, 5, 7, 9]. Убедитесь, что она работает с отрицательным шагом так, что вызов range(5, 2, -1) возвращает [5, 4, 3, 2].
7. Обращаем массив вспять. Напишите две функции, reverseArray и reverseArrayInPlace. Первая получает массив как аргумент и выдаёт новый массив, с обратным порядком элементов. Вторая работает как оригинальный метод reverse – она меняет порядок элементов на обратный в том массиве, который был ей передан в качестве аргумента. Не используйте стандартный метод reverse.
8. Глубокое сравнение. Оператор == сравнивает переменные объектов, проверяя, ссылаются ли они на один объект. Но иногда полезно было бы сравнить объекты по содержимому. Напишите функцию deepEqual, которая принимает два значения и возвращает true, только если это два одинаковых значения или это объекты, свойства которых имеют одинаковые значения, если их сравнивать рекурсивным вызовом deepEqual. Чтобы узнать, когда сравнивать величины через ===, а когда – объекты по содержимому, используйте оператор typeof. Если он выдаёт “object” для обеих величин, значит нужно делать глубокое сравнение. Не забудьте об одном дурацком исключении, случившемся из-за исторических причин: “typeof null” тоже возвращает “object”.
9. Свертка. Используйте метод reduce в комбинации с concat для свёртки массива массивов в один массив, у которого есть все элементы входных массивов.