

Задания к лабораторным работам по блоку "Алгоритмы обработки данных на JavaScript"

Содержание

1	Общие указания	2
1.1	alert	2
1.2	prompt	2
1.3	console.log	2
2	Лабораторная работа №1	3
3	Лабораторная работа №2	4
4	Лабораторная работа №3	6
5	Лабораторная работа №4	8
5.1	Одномерные массивы	8
5.2	Многомерные массивы	8
6	Лабораторная работа №5	10
6.1	Пользовательские структуры данных	10
6.2	Функции высшего порядка	11
6.3	Функции как возвращаемые объекты	11
7	Лабораторная работа №6	12
8	Лабораторная работа №7	13

1 Общие указания

Для выполнения лабораторной работы необходимо написать исходный код требуемой программы.

Ввод параметров в программу осуществляется при помощи функции `prompt`, если не указано иное.

Результат работы программы должен выводиться при помощи функции `alert`, либо `console.log`.

Описание этих функций представлено ниже.

1.1 alert

Функция `alert(message)` показывает пользователю переданное сообщение *message*.

1.2 prompt

Функция `prompt(message)` показывает сообщение *message* и запрашивает ввод текста у пользователя. Возвращает напечатанный текст или `null`, если пользователь закрыл окно.

```
const name = prompt('Введите своё имя');  
alert(name); // Распечатает введённый текст
```

1.3 console.log

Функция `console.log(message)` выводит переданное ей сообщение *message* в консоль разработчика.

2 Лабораторная работа №1

В процессе написания лабораторной работы ознакомиться с основами языка JavaScript.

Варианты:

1. Программа должна вычислять площадь треугольника по трём сторонам.
Входные данные: длины сторон a, b, c .
Выходные данные: площадь треугольника S .
Формула для расчёта: $S = \sqrt{p(p-a)(p-b)(p-c)}$, где $p = \frac{a+b+c}{2}$.
2. Программа должна вычислять сумму заданного количества начальных элементов геометрической прогрессии.
Входные данные: первый элемент геометрической прогрессии b_1 , знаменатель прогрессии q , количество элементов прогрессии n .
Выходные данные: сумма заданного количества элементов заданной геометрической прогрессии S .
Формула для расчёта: $S = \frac{b_1(q^n - 1)}{q - 1}$.
3. Программа должна вычислять сумму заданного количества начальных элементов арифметической прогрессии.
Входные данные: первый элемент арифметической прогрессии a_1 , шаг прогрессии d , количество элементов прогрессии n .
Выходные данные: сумма заданного количества элементов заданной арифметической прогрессии S .
Формула для расчёта: $S = \frac{2a_1 + d(n-1)}{2} \cdot n$.
4. Программа должна вычислять радиус окружности, описанной около треугольника, заданного длинами сторон.
Входные данные: длины сторон a, b, c .
Выходные данные: радиус описанной окружности R .
Формула для расчёта: $R = \frac{abc}{4\sqrt{p(p-a)(p-b)(p-c)}}$, где $p = \frac{a+b+c}{2}$.
5. Программа должна вычислять радиус окружности, вписанной в треугольник, заданного длинами сторон.
Входные данные: длины сторон a, b, c .
Выходные данные: радиус вписанной окружности r .
Формула для расчёта: $r = \sqrt{\frac{(p-a)(p-b)(p-c)}{p}}$, где $p = \frac{a+b+c}{2}$.

3 Лабораторная работа №2

В процессе написания лабораторной работы ознакомиться с управляющими операторами и циклами в языке JavaScript.

Варианты:

1. Напишите исходный код программы, которая считывает единственное целое число n и выводит результат проверки этого числа на простоту. Например, числа 1, 2, 3, 5, 7, 11, 13... являются простыми, так как делятся только сами на себя и на единицу.
Входные данные: число n .
Выходные данные: Сообщение о том, является ли число простым.
2. Напишите исходный код программы, которая считывает единственное целое число n и проверяет, является ли это число автоморфным. Натуральное число называется автоморфным, если десятичная запись его квадрата оканчивается на десятичную запись этого числа. Например, число 5 является автоморфным так как $5^2 = 25$. Число 25 является автоморфным так как $25^2 = 625$.
Входные данные: число n .
Выходные данные: Сообщение о том, является ли число автоморфным.
3. Напишите исходный код программы, которая считывает единственное целое число n и проверяет, является ли оно совершенным. Совершенным называется число, равное сумме всех своих собственных делителей (отличных от n). Например, $6 = 1 + 2 + 3$; $28 = 1 + 2 + 4 + 7 + 14$.
Входные данные: число n .
Выходные данные: Сообщение о том, является ли число совершенным.
4. Напишите исходный код программы, которая считывает единственное целое число n и проверяет, является ли оно избыточным. Избыточным называется положительное целое число, сумма положительных собственных делителей (отличных от n) которого превышает n . Например, число 12 является избыточным, т.к. $1 + 2 + 3 + 4 + 6 = 16$. $16 > 12$.
Входные данные: число n .
Выходные данные: Сообщение о том, является ли число избыточным.
5. Напишите исходный код программы, которая считывает единственное целое число s и проверяет, является ли оно негипотенузным. Негипотенузное число - натуральное число, квадрат которого не может быть записан как сумма двух ненулевых квадратов чисел. Примеры негипотенузных чисел: 1, 2, 3, 4, 6, 7,

Входные данные: число c .

Выходные данные: Сообщение о том, является ли число негипотенузным.

6. Напишите исходный код программы, которая находит все числа Армстронга на заданном пользователем отрезке $[a, b]$.

Числа Армстронга – это натуральные числа, равные сумме своих цифр, возведённых в степень количества цифр в их записи.

Пример: $153 = 1^3 + 5^3 + 3^3$.

Найденные числа должны быть выведены в консоль.

Входные данные: левая граница отрезка a и правая граница отрезка b .

Выходные данные: Сообщения с найденными числами Армстронга. Числа должны быть выведены в консоль.

7. Напишите исходный код программы, которая находит первые n Пифагоровых троек. Число n вводится пользователем.

Числа a, b, c составляют Пифагорову тройку, если $a^2 + b^2 = c^2$.

Входные данные: Требуемое количество пифагоровых троек n .

Выходные данные: Сообщения с найденными Пифагоровыми тройками. Пифагоровы тройки должны быть выведены в консоль в формате (a,b,c).

4 Лабораторная работа №3

В процессе написания лабораторной работы ознакомиться с механизмом рекурсии в языке JavaScript.

Варианты:

1. Быстрое возведение числа b в степень n определяется следующими правилами:

$$\begin{cases} b^0 = 1 \\ b^1 = b \\ b^n = (b^{n/2})^2 & \text{— если } n \text{ четно} \\ b^n = b \times b^{n-1} & \text{— если } n \text{ нечетно} \end{cases} \quad (1)$$

Реализуйте рекурсивную функцию, находящую b^n .

Входные данные: основание b и степень n .

Выходные данные: Число b , возведённое в степень n .

2. Функция Аккермана определяется следующим образом:

$$A(m, n) = \begin{cases} n + 1, & m=0; \\ A(m-1, 1), & m > 0 \text{ } n = 0; \\ A(m-1, A(m, n-1)), & m > 0, n > 0. \end{cases} \quad (2)$$

Напишите функцию, которая вычисляет значение функции Аккермана по заданным n и m .

3. Реализуйте рекурсивный алгоритм вычисления суммы n первых членов ряда:

$$1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^{n-1}}{(n-1)!}. \quad (3)$$

Входные данные: значение x и количество членов ряда n .

Выходные данные: значение суммы n первых членов ряда в заданной точке x .

4. Реализуйте алгоритм вычисления последовательности n вложенных корней:

$$\sqrt{m + \sqrt{m + \dots + \sqrt{m}}}. \quad (4)$$

Например, для $n = 3$ в точке $m = 4$: $\sqrt{4 + \sqrt{4 + \sqrt{4}}}$.

Входные данные: значение m и количество вложенных корней n .

Выходные данные: значение последовательности n вложенных корней в точке m .

5. Реализуйте алгоритм вычисления n вложенных синусов с использованием рекурсии:

$$\sin \sin \dots \sin x. \quad (5)$$

Например, для $n = 3$ и $x = 0$: $\sin \sin \sin 0$.

Входные данные: значение x и количество вложенных синусов n .

Выходные данные: значение n вложенных синусов в точке x .

6. Синус угла (заданного в радианах) можно вычислить приближением $\sin x \approx x$ при малых значениях x и употребить тригонометрическое равенство

$$\sin x = 3 \sin \frac{x}{3} - 4 \sin^3 \frac{x}{3} \quad (6)$$

для уменьшения значения аргумента \sin . Будем считать угол достаточно малым, если он не больше 0.1 радиана.

Реализуйте рекурсивную функцию, находящую значение синуса в заданной точке x .

Входные данные: точка x .

Выходные данные: значение синуса в точке x .

5 Лабораторная работа №4

В процессе написания лабораторной работы ознакомиться с алгоритмами обработки массивов в JavaScript.

5.1 Одномерные массивы

Варианты:

1. Напишите функцию, принимающую на вход массив вещественных чисел и возвращающую сумму элементов массива, расположенных до минимального элемента.
2. Напишите функцию, принимающую на вход массив вещественных чисел и возвращающую сумму модулей элементов массива, расположенных после первого элемента равного нулю.
3. Напишите функцию, принимающую на вход массив вещественных чисел и возвращающую сумму элементов, расположенных после последнего элемента равного нулю.
4. Напишите функцию, принимающую на вход массив вещественных чисел и возвращающую произведение элементов, расположенных между максимальным и минимальным элементами.
5. Напишите функцию, принимающую на вход массив вещественных чисел и возвращающую количество элементов, равных предыдущему элементу.
6. Напишите функцию, принимающую на вход массив вещественных чисел и возвращающую среднее арифметическое нечётных элементов.
7. Напишите функцию, принимающую на вход массив вещественных чисел и возвращающую количество элементов, неравных своему предыдущему.

Создайте массив при помощи литерала массива и запустите функцию на выполнение.

5.2 Многомерные массивы

Варианты:

1. Напишите функцию, принимающую на вход вещественную прямоугольную матрицу и возвращающую номера строк и столбцов всех седловых точек матрицы.
Матрица A имеет седловую точку $A_{i,j}$, если $A_{i,j}$ является минимальным элементом в i -й строке и максимальным элементом в j -м столбце.

2. Напишите функцию, принимающую на вход вещественную прямоугольную матрицу и возвращающую минимум среди сумм элементов диагоналей, параллельных главной диагонали матрицы.
3. Напишите функцию, принимающую на вход вещественную прямоугольную матрицу и возвращающую сумму элементов в тех строках, которые содержат хотя бы один отрицательный элемент.
4. Напишите функцию, принимающую на вход вещественную прямоугольную матрицу и возвращающую одномерный массив, состоящий из минимальных четных элементов строк матрицы.
5. Напишите функцию, принимающую на вход вещественную прямоугольную матрицу и возвращающую одномерный массив, состоящий из сумм отрицательных элементов соответствующих столбцов матрицы.
6. Напишите функцию, принимающую на вход вещественную прямоугольную матрицу и возвращающую одномерный массив, элементами которого будут номера последних отрицательных элементов строк матрицы.
7. Напишите функцию, принимающую на вход вещественную прямоугольную матрицу и возвращающую одномерный массив, состоящий из минимальных элементов соответствующих столбцов матрицы.

Создайте многомерный массив при помощи литерала массива и запустите функцию на выполнение.

6 Лабораторная работа №5

В процессе написания лабораторной работы ознакомиться с объектами JavaScript и функциями высшего порядка.

6.1 Пользовательские структуры данных

Создайте класс Student. Данный класс должен содержать:

- свойство, хранящее фамилию студента;
- свойство, хранящее имя студента;
- свойство marks, содержащее массив объектов следующего вида:
 - subjectName - название предмета;
 - mark - оценка по предмету.
- добавьте в класс метод, возвращающий среднюю оценку студента по всем предметам;
- добавьте в класс метод, возвращающий все оценки по переданному предмету;
- добавьте в класс метод добавления оценки по предмету.
- добавьте в класс метод, удаляющий все оценки по переданному предмету.

Создайте класс Group. Данный класс должен содержать:

- свойство имени группы;
- свойство массива студентов данной группы;
- добавьте метод, позволяющий добавить в группу студента;
- добавьте метод, позволяющий удалить студента по индексу;
- добавьте метод, возвращающий ассоциативный массив, в котором ключом является фамилия и имя студента, а значением - средняя оценка соответствующего студента.

Проверить работоспособность написанного кода. Для этого создайте группу, добавьте в неё пару студентов, а также вызовите реализованные методы.

6.2 Функции высшего порядка

Варианты:

1. Необходимо реализовать функцию *findLastIndex(array, predicate)*. Функция должна возвращать индекс последнего элемента в массиве, удовлетворяющего условию *predicate*. В противном случае возвращается -1.
2. Необходимо реализовать функцию *takeWhile(array, predicate)*. Функция должна создавать срез массива начиная с начала. Элементы массива берутся до тех пор пока *predicate* возвращает *false*.
3. Необходимо реализовать функцию *partition(array, predicate)*. Функция должна разделять исходный массив на две группы. Первая группа содержит элементы, для которых *predicate* возвращает *true*, а вторая - элементы, для которых *predicate* возвращает *false*.
4. Необходимо реализовать функцию *flatMap(array, iteratee)*. Функция должна применять к каждому элементу массива *array* функцию *iteratee*, а затем преобразовать полученный результат в плоскую структуру и помещать в результирующий массив.

6.3 Функции как возвращаемые объекты

Варианты:

1. Пусть *f* и *g* - две одноаргументные функции. Композиция функций *f* и *g* есть функция $x \rightarrow f(g(x))$. Определите функцию *compose*, которая реализует композицию.

Например, если *inc* - функция, добавляющая к своему аргументу один:

```
const addTwo = compose(inc, inc);
console.log(addTwo(3)); // 5
```

2. Пусть *f* - функция на численных аргументах, а *n* - положительное целое число. Мы можем построить *n*-кратное применение *f*, которое определяется как функция, значение которой в точке *x* равно $f(f(\dots(f(x))\dots))$.

Например, если *f* функция инкремента $x \rightarrow x + 1$, то *n* кратным применением будет функция $x \rightarrow x + n$.

Напишите функцию, которая принимает на вход функцию *f* и положительное целое число *n*, и возвращает функцию, вычисляющую *n*-кратное применение *f*.

7 Лабораторная работа №6

В процессе написания лабораторной работы ознакомиться с алгоритмами обработки строковых данных в JavaScript.

Варианты:

1. Необходимо реализовать функцию, которая находит первую строку текста, содержащую заданную подстроку. Считается, что строки текста оканчиваются переводами строки.
Функция должна возвращать саму строку и её номер. При поиске регистр символов должен игнорироваться.
2. Необходимо реализовать функцию, которая находит самое популярное слово в тексте. Слова состоят из идущих подряд символов русского алфавита в произвольном регистре и разделяются другими символами. При подсчёте регистр символов должен игнорироваться.
3. Необходимо реализовать функцию, которая подсчитывает количество гласных и согласных букв в тексте. Текст состоит из символов русского алфавита. При подсчёте регистр символов должен игнорироваться.
4. Необходимо реализовать функцию, которая заменяет первую букву в каждом слове текста на соответствующую заглавную букву. Слова состоят из идущих подряд символов русского алфавита в произвольном регистре и разделяются другими символами.
5. Напишите функцию, которая заменяет заданную подстроку в заданном тексте на другую заданную подстроку.
6. Напишите функцию, которая для каждого символа, встречающегося в переданном тексте, возвращает частоту, с которой этот символ встречается. При подсчёте регистр символов должен игнорироваться.
7. Напишите функцию, которая вычисляет количество предложений в переданном тексте. Предложением считается непустая последовательность печатных символов, оканчивающаяся точкой, вопросительным знаком или восклицательным знаком.
8. Напишите функцию, которая должна находить все числа, встречающиеся в переданном тексте. Числами можно считать последовательности из подряд идущих арабских цифр, отделённых друг от друга любыми другими символами.
9. Напишите функцию, которая должна находить в переданном тексте уникальные слова. Под уникальными словами понимаются слова, встречающиеся только один раз. Слова состоят из идущих подряд символов русского алфавита в произвольном регистре и разделяются другими символами. При подсчёте регистр символов должен игнорироваться.

8 Лабораторная работа №7

В процессе написания лабораторной работы ознакомиться с алгоритмами сортировки и поиска в JavaScript.

- Добавьте в класс Group метод, производящий сортировку по средней оценке. Метод должен реализовывать сортировку Шелла.
- Добавьте в класс Group метод, производящий бинарный поиск студентов по фамилии и имени. Сортировку массива произвести при помощи метода sort.