

Лабораторная работа №4.

Решения должны находиться в файле с названием Вашей лиги внутри методов с соответствующими номерами. Ваш код для решения задания необходимо писать в блоке, обозначенном границами `//code here` и `//end`. Код вне этого блока не изменять. При решении используйте переменные, которые даны по условию задания. Ответ на поставленную задачу должен быть присвоен переменной `answer` (или другой, указанной в шаблоне после `return`). Тип возвращаемой переменной не менять. Если в номере отсутствует переменная для возврата, то проверяться будет передаваемый в номер массив `array`. На вход **гарантируется** передача данных (не пустой массив и не `null`).

Общие граничные условия для всех заданий:

1. Проверка входных данных:

Если входные данные не позволяют выполнить условие согласно заданию, или данных недостаточно для выполнения работы, возвращать значение по умолчанию: 0 для значимых типов, `null` для ссылочных.

Если передаются индексы или номера элементов, которые выходят за границы массивов, возвращать значение по умолчанию: 0 для значимых типов, `null` для ссылочных.

Если по условию задания подразумевается изменение исходного массива (без возврата нового), то при невозможности выполнения оставить *исходный массив без изменений*.

2. Равные значения:

При поиске максимального/минимального/отрицательного/положительного/т.п. элемента в массиве, если условию соответствует несколько одинаковых элементов, по умолчанию брать первое встречаемое значение при прямом (слева направо) проходе (кроме случаев, когда в задании явно прописаны другие условия).

3. Возврат корректных пустых массивов:

Если ожидается массив, содержащий некий набор значений, но подходящие значения не были найдены в исходном массиве или же после выполнения задания все элементы массива были удалены, вернуть пустой массив, а не `null` (кроме случаев, когда по условию требуется `null`).

4. Сохранение порядка:

При необходимости сдвига/перемещения/сортировки элементов сохранять порядок элементов относительно друг друга. Если массив пуст или содержит один элемент, перестановка элементов не требуется.

При необходимости сортировки элементов следует использовать любую **устойчивую** сортировку.

Если все автоматические тесты пройдены успешно, Вы можете отправить лабораторную на заключительную проверку на GitHub. Более подробная инструкция описана в задании на Moodle. Если

Ваша работа принята, мы рекомендуем прорешать номера из других лиг в качестве подготовки к контрольной. Особенно те, которые отличаются от заданий из Вашей лиге.

В работе разрешены методы классов: **Console**, **Math**, **Random**, а также свойство массивов **Length**.

В решении задач разрешается использовать **не более 1го уровня вложенности циклов**.

Задания белой лиги.

1. В метод передается одномерный массив `vector`. Вычислить длину вектора, заданного массивом. Длина вектора вычисляется по формуле: $L = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$.
2. В метод передается одномерный массив `array` и целые числа `P`, `Q`. Определить, сколько элементов массива заключено строго между значениями `P` и `Q`.
3. В метод передается одномерный массив `array`. Поменять местами максимальный элемент массива и минимальный элемент из части массива, расположенной после максимального. Если после максимального элемента нет элементов или все элементы равны – изменений не производить.
4. В метод передается одномерный массив `array`. Максимальный элемент массива среди элементов с четными индексами заменить значением его индекса.
5. В метод передается одномерный массив `array` и число `P`. Найти индекс числа `P`. Вернуть `-1`, если число не найдено.
6. В метод передается одномерный массив `array`. Поменять местами соседние элементы (1-й со 2-м, 3-й с 4-м и т.д.), расположенные до максимального элемента массива. Если до максимального нет элементов – не выполнять перестановку.
7. В метод передается одномерный массив `array`. Удалить все отрицательные элементы массива.
8. В метод передается одномерный массив `array`. Отсортировать массив по убыванию значений элементов.
9. В метод передается одномерный массив `array`. Развернуть массив.
10. В метода передаются два одномерных массива `A` и `B`. Соединить их в новый массив `C`, добавляя элементы поочередно из массивов `A` и `B`, начиная с массива `A`. Если один массив “закончится” раньше, дозаполнить массив `C` элементами другого массива. Если один массив пуст – вернуть копию другого. Если оба пусты – вернуть пустой массив.
11. В метод передаются вещественные числа `a`, `b` и целое число `n`. Сформировать массив длины `n`, состоящий из равноудаленных элементов в диапазоне от `a` до `b`. Если $a > b$ – значения формировать в порядке уменьшения, если $a < b$ – в порядке увеличения. Если $a = b$ и $n = 1$, вернуть массив, состоящий из одного элемента. В остальных случаях вернуть `null`.
12. При обработке архивных глиняных дисков обнаружен массив закольцованных числовых данных, где часть информации утрачена. Значение `-1` указывает на поврежденные данные. В ходе исследования установлено, что пропущенные значения можно восстановить как среднее арифметическое соседних элементов. Если соседние элементы также повреждены, оставить значение без изменений. Пусть его расшифровывают математики. На вход подается поврежденный массив `raw`, вывести восстановленный массив `restored`. Если элементов меньше трех, прочитать данные невозможно. Если все элементы равны `-1` – вернуть копию исходного массива.

Задания зеленой лиги.

1. В метод передается одномерный массив `array`. Положительные элементы массива заменить средним арифметическим среди положительных элементов.
2. В метод передается одномерный массив `array`. Найти сумму квадратов элементов, расположенных до первого отрицательного элемента массива.
3. В метод передается одномерный массив `array`. Сформировать другой одномерный массив из отрицательных элементов, расположенных между максимальным и минимальным элементами массива. Если максимальный элемент встречается раньше минимального, считать диапазон от максимального до минимального, иначе – наоборот. Сами элементы `max` и `min` не включать в новый массив.
4. В метод передается одномерный массив `array`. Поменять местами максимальный и первый отрицательный элементы массива.
5. В метод передается одномерный массив `array`. Найти все максимальные элементы массива. Сформировать массив из их индексов.
6. В метод передается одномерный массив `array`. В массиве увеличить максимальные элементы на их порядковые номера. Нумерация максимальных элементов ведётся слева направо, начиная с 1.
7. В метод передается одномерный массив `array`. Максимальный элемент (или максимальные элементы, если их несколько) заменить суммой элементов массива, расположенных до него (до каждого из них, если их несколько).
8. В метод передается одномерный массив `array`. Найти максимальное количество следующих подряд упорядоченных по убыванию элементов. Под убыванием понимать строгое уменьшение (`array[i] > array[i+1]`). Если весь массив неубывающий, результат равен 1 (длина минимальной подпоследовательности).
9. В метод передается одномерный массив `array`. Упорядочить по возрастанию элементы массива с четными индексами (остальные элементы оставить на своих местах).
10. В метод передается одномерный массив `array`. Удалить повторяющиеся элементы массива. При наличии повторяющихся элементов в массиве сохранять первые встреченные (слева направо). Порядок остальных элементов должен сохраняться.
11. В метод передаются вещественные числа `a`, `b` и целое число `n`. Сформировать массив `A` длины `n`, состоящий из равноудаленных элементов в диапазоне от `a` до `b`. Положительные элементы массива `A`, которые превышают среднее арифметическое положительных элементов массива, выделить в массив `B`.
12. В таверне «Серебряный дракон» любят играть в кости на золото. Волшебные кубики Вальдара иногда дают критический выигрыш (если выпало 6 очков), но по легенде, они прокляты: каждая выпавшая шестерка уменьшает на 1 значение всех кубиков, выпавших после неё, но значения не могут стать меньше 1. После применения проклятия подсчитывается количество выигранных игр против шулера таверны, который выбрасывает шестерки на всех волшебных кубиках Вальдара. На вход подаётся массив `dices`, содержащий результаты ваших бросков. Сколько игр вы сможете выиграть с результатом кубиков `dices` у главного шулера таверны с кубиками, который всегда выбрасывает на всех кубиках шестерки? Победой считается результат, при котором значение кубика игрока строго больше значения шулера на соответствующем кубике.

Задания синей лиги.

1. В метод передается одномерный массив `array`. Первый отрицательный элемент массива заменить суммой элементов, расположенных после максимального элемента массива. Если максимальный элемент последний, сумма после него считается равной 0.
2. В метод передается одномерный массив `array` и число `P`. Вставить заданный элемент `P` после последнего положительного элемента массива и вернуть как новый массив. Если в массиве нет положительных элементов, вернуть копию исходного массива.
3. В метод передается одномерный массив `array`. Удалить минимальный среди положительных элементов массива. Если в массиве нет положительных элементов, вернуть копию исходного массива.
4. В метод передается одномерный массив `array`. Найти среднее значение элементов массива. Преобразовать элементы исходного массива, вычитая из каждого элемента полученное значение.
5. В метод передаются одномерные массивы `A` и `B`. Вычислить скалярное произведение массивов `A` и `B`. Скалярным произведением называется сумма попарных произведений соответствующих элементов массивов.
6. В метод передается одномерный массив `array`. Индексы элементов массива, меньших среднего, поместить в новый массив.
7. В метод передается одномерный массив `array`. Определить длину самой большой непрерывной упорядоченной (по возрастанию или по убыванию) последовательности. Последовательность считается непрерывной, если её элементы равны или их попарная разность одного знака.
8. В метод передается одномерный массив `array`. Продублировать все элементы с сохранением порядка следования. Например, передается `array = {3, 8, ...}`, получить `array = {3, 3, 8, 8, ...}`.
9. В метод передается одномерный массив `array`. Нормировать значения массива, чтобы его элементы принадлежали отрезку $[0, 1]$. Если все элементы массива равны, вернуть `null`.
10. В метод передается одномерный массив `array` и число `P`. Отсортировать массив по возрастанию и найти индекс числа `P` бинарным поиском в отсортированном массиве. Вернуть `-1`, если число не найдено.
11. В метод передаются положительные целые числа `a`, `b`, `c`. Сформировать массив из элементов, начинающийся с элемента `a`, увеличивая значение каждого последующего элемента на `b` до тех пор, пока элемент не превысит значение `c`. Число `b` должно быть положительным.
12. Группа археологов исследуют древнюю египетскую гробницу. Внутри они обнаружили несколько запечатанных комнат, в каждой из которых, по слухам, хранятся золотые слитки фараона. Однако не все комнаты сохранились нетронутыми — некоторые были разграблены еще в древности. Записи о количестве золотых слитков в каждой пронумерованной комнате были записаны в журнал `magazine`. Нулевые значения означают, что комната пуста. У археологов есть время посетить только 3 соседние комнаты и забрать оттуда золото. Помогите им выбрать, какие 3 комнаты нужно посетить, чтобы забрать как можно больше золотых слитков. Если комнат меньше 3-х, то посетить все, что есть. Ответ представить в виде массива индексов комнат.

Задания фиолетовой лиги.

1. В метод передается одномерный массив `array`. В массиве все элементы, расположенные после максимального, заменить средним значением элементов массива. Если максимальный элемент встречается несколько раз, заменить элементы после первого максимального.
2. В метод передается одномерный массив `array`. Сформировать два массива, включая в первый массив элементы исходного массива с четными индексами, во второй – с нечетными. Если длина исходного массива нечетная, первый выходной массив должен быть на 1 элемент больше второго.
3. В метод передается одномерный массив `array` и число `P`. Добавить элемент, равный `P`, после того элемента массива, который наиболее близок к среднему значению его элементов. Если несколько элементов одинаково близки, выбрать первый слева.
4. В метод передается одномерный массив `array`. Все отрицательные элементы переместить в конец массива с сохранением порядка их следования. Порядок положительных элементов также сохраняется.
5. В метод передаются одномерные массивы `A`, `B` и целое число `k`. Вставить массив `B` между `k`-м и `(k + 1)`-м по счету элементами массива `A`. Если `k` – это номер последнего элемента, вставить массив `B` в конец массива `A`.
6. В метод передаются одномерные массивы `A` и `B`. Вычислить сумму и разность массивов `A` и `B`. Суммой (разностью) двух массивов одинакового размера называется третий массив такого же размера, каждый элемент которого равен сумме (разности) соответствующих элементов исходных массивов.
7. В метод передается одномерный массив `array`. Преобразовать значения массива, чтобы его элементы принадлежали отрезку `[-1, 1]`. Если все элементы массива равны, вернуть `null`.
8. В метод передаются одномерные массивы `A` и `B`. Отсортировать их по убыванию и слить в новый массив `C`, сохраняя упорядочение элементов.
9. В метод передается одномерный массив `array`. Найти индекс его первого максимального элемента. Сдвинуть циклически все элементы массива вправо на величину найденного индекса.
10. В метод передается одномерный массив `array` и целое число `N`. Отзеркалить элементы массива, идущие до `N`-ого по счету элемента массива, после него. Элементы справа от `N`-го заменяются зеркальной копией элементов слева от `N`-го. Например, передается `array = {3, 1, 8, 4, 9, 5, 7 ...}` и `N = 3`, получить `array = {3, 1, 8, 1, 3, 5, 7 ...}`. Если слева больше, чем справа – заменяем столько, сколько есть справа. Если справа больше, чем слева – заменяем столько, сколько есть слева.
11. В метод передаются числа `a`, `b`, `n`. Вычислить значения функции $y = \cos x + x \cdot \sin x$ в `n` равноудаленных точках отрезка `[a, b]`. Результат получить в двух массивах `X` (аргумент), `Y` (функция). Используя сформированные массивы, определить все значения аргумента, при которых функция имеет максимум или минимум (включая локальные). Сформировать массивы аргументов и значений экстремумов `Xext` и `Yext` соответственно. Функция имеет локальный (или глобальный) максимум со значением `Yi` при значении аргумента `Xi`, если значения функции `Yi-1` и `Yi+1` в двух соседних точках меньше `Yi`. Аналогичные расчеты для поиска минимума. Границы массива не могут считаться локальными экстремумами.
12. Группа астрофизиков получила новые данные с орбитального телескопа, записанные в массив измерений яркости звезд. В данных обнаружены аномальные значения (артефакты съемки), которые необходимо отфильтровать перед анализом. Для очистки выборки выполнить следующие действия:

- a. Выделить в отдельные массивы слишком яркие значения — те, что более чем в 2 раза превышают среднее всех элементов, и слишком тусклые — те, что меньше половины среднего всех элементов.
- b. Найти среднее значение оставшихся элементов без учета удаленных значений (пропусков).
- c. В образовавшиеся пропуски вставить полусумму найденного среднего и значения соответствующего удаленного элемента из исходного массива.
- d. Отсортировать полученные «нормальные» значения по убыванию.

На выходе получить 3 массива: bright, normal, dim.