**Лабораторная работа 04.** Решить поставленную задачу программирования по разделу «Одномерные массивы». Продумать организацию удобного ввода элементов одномерного массива. Продумать схему корректного вывода элементов одномерного массива.

**Учесть все возможные ограничения**, накладываемые не только на переменные, но и на участвующие в расчёте функциональные зависимости (если они указаны в задании по варианту).

**Ввести контроль исходных данных**. Реализовать схему «ввод до победного» без возможности прерывания. При некорректном вводе исходных данных заставлять пользователя вводить сведения до тех пор, пока он не введёт их корректно. Продумать побуждающие сообщения-подсказки, направляющие пользователя ко вводу корректных значений.

Использовать для контроля исходных данных различные состояния метода «*TryParse*», входящего в перечень доступных компонентов интересующего значащего (*valuable*) типа данных, например: *int, byte, float, double*.

Подобрать корректные тестовые примеры **в необходимом количестве\***.

Помнить о точности производимых вычислений при работе с вещественными значениями, например:

**«При заданной точности  , но ,**

**то есть , а ».**

Код составленного программного обеспечения сопроводить комментариями.

Отчёт по решённой задаче составить по схеме, изложенной в методических указаниях [Сафронов А.И., Зольникова Н.Н., Новиков В.Г. Составление отчётной документации по решённым задачам алгоритмизации и программирования: Учебно-методическое пособие для проведения аудиторных занятий по Учебной практике. – М.: РУТ (МИИТ), 2018. – 83 с.]. Скачать можно на сайте Национальной Электронной Библиотеки (НЭБ) *elibrary.ru* : <https://www.elibrary.ru/item.asp?id=46271697>.

Для удобства работы обучающихся далее представлена памятка основных разделов отчёта:

0. Титульный лист

1. Цель работы

2. Формулировка задачи

3. Блок-схема алгоритма

4. Подбор тестовых примеров

5. Листинг (код) программы

6. Расчёт тестовых примеров на ПК

7. Вывод по работе

Блок-схему алгоритма оформлять согласно правилам, изложенным в разделе 8 того же учебно-методического пособия.

Для ввода и вывода элементов одномерных массивов может быть применена упрощённая схема иллюстрирования: типовой фрагмент «циклический ввод» – заменён одним блоком «Данные», типовой фрагмент «циклический вывод» – заменён одним блоком «Документ».

\*Пример корректного подбора тестовых примеров в нужном количестве представлен в разделе 7 Сборника задач [Сафронов А.И., Зольникова Н.Н., Новиков В.Г. Получение первичных профессиональных умений и навыков научно-исследовательской деятельности: Сборник задач для проведения аудиторных занятий по Учебной практике. – М.: РУТ (МИИТ), 2019. – 91 с.]. Скачать можно на сайте Национальной Электронной Библиотеки (НЭБ) *elibrary.ru* : <https://www.elibrary.ru/item.asp?id=46307421>.

**Принять во внимание тот факт, что все необходимые для тестирования значения могут быть занесены сразу в один единственный одномерный массив.**

Материалы, подготавливаемые к отправке по электронной почте, оформляются согласно Приложению 4 того же Сборника задач.

Настоятельно рекомендовано перед решением задачи ознакомиться с рекомендациями, собранными в Приложении 3 того же Сборника задач.

Не забывать давать значимые имена переменным, проектам, решениям, файлам и другим разрабатываемым компонентам. При нежелании использовать значимые имена переменных в отчёте необходимо составить таблицу соответствия вида:

|  |  |  |
| --- | --- | --- |
| № п/п | Наименование переменной в программе | Смысловое содержание используемой переменной |
| 1 | *a* | Ускорение |
| 2 | *v* | Скорость |
| 3 | *S* | Пройдённый путь |
| 4 | *t* | Затраченное время |
| 5 | *x* | Абсцисса точки на Декартовой плоскости |
| 6 | *y* | Ордината точки на Декартовой плоскости |
| 7 | *z* | Аппликата точки в пространстве |

Проанализировать исходные данные. Определить, какие их переменных являются аргументами функции, а которые поступают в формате констант для расчёта зависимости. На константы наложить ограничения на этапе ввода исходных данных, переменные значения контролировать на каждой итерации\*\* цикла (циклов).

\*\*Итерация в программировании – в широком смысле – организация обработки данных, при которой действия повторяются многократно, не приводя при этом к вызовам самих себя (в отличие от рекурсии). В узком смысле – один шаг итерационного, циклического вычислительного процесса.

**Внимание! При выявлении противоречивых условий среди вариантов индивидуального задания не предпринимать самостоятельных решений по устранению противоречий, а обратиться за консультацией к преподавателю, читающему курс «Программирования и основ алгоритмизации».**

***Корректный ввод элементов одномерного массива в общем виде программного обеспечения имеет формат:***

*«Введите количество элементов одномерного массива: [целое числовое значение]»*

*«Введите 1-й элемент одномерного массива: [числовое значение]»*

*«Введите 2-й элемент одномерного массива: [числовое значение]»*

*… … … …*

*«Введите [целое числовое значение]-й элемент одномерного массива: [числовое значение]»*

***Корректный вывод элементов одномерного массива в общем виде программного обеспечения имеет формат:***

*[имя одномерного массива] = (4.50, -0.01, 6.78, …, -15.39)*

**Задание состоит только из Индивидуальной части. Общая часть в нём отсутствует.**

**Вариант №1**

Определить количество различных чисел в одномерном массиве.

**Вариант №2**

Дан массив натуральных чисел. Вычислить сумму тех элементов, порядковые номера которых являются числами Фибоначчи.

**Вариант №3**

Переставить элементы одномерного массива таким образом, чтобы положительные элементы в нём предшествовали отрицательным элементам. При этом взаимный порядок как положительных, так и отрицательных элементов должен быть сохранён.

**Вариант №4**

Заполнить одномерный массив указанным количеством чисел Фибоначчи.

**Вариант №5**

Определить количество нулевых элементов в заданном одномерном массиве целых чисел.

**Вариант №6**

В одномерном массиве найти число, которое встречается в нём наибольшее количество раз. При этом ответ выдавать только для случаев, когда хотя бы одно из значений упоминается более одного раза. Иная ситуация инициирует выдачу сообщения об ошибке.

**Вариант №7**

В заданном одномерном массиве чисел определить позиции максимального и минимального значений. Счёт позиций начинается с единицы при индексации массива с нуля.

**Вариант №8**

Сформировать массив чисел, записанных в порядке возрастания, составленный из нечётных элементов заданного одномерного массива.

**Вариант №9**

Найти сумму положительных элементов одномерного массива.

**Вариант №10**

В одномерном массиве найти разность индексов наибольшего и наименьшего элементов.

**Вариант №11**

В заданном одномерном массиве чисел найти количество элементов, удовлетворяющих условию .

**Вариант №12**

Определить количество нулевых элементов одномерного массива, стоящих на нечётных позициях. Обратить внимание, что счёт позиций начинается с единицы, а индексируется массив с нуля.

**Вариант №13**

Сформировать массив чисел, упорядоченных по убыванию, только из нечётных элементов заданного одномерного массива.

**Вариант №14**

Найти сумму элементов одномерного массива, стоящих между первым и последним нулевыми элементами.

**Вариант №15**

Переставить элементы одномерного массива с чётными индексами в порядке возрастания квадратов их значений. Счёт позиций начинается с единицы, счёт индексации – с нуля.

**Вариант №16**

Определить количество нулевых элементов в заданном одномерном массиве целых чисел.

**Вариант №17**

Вычислить сумму чётных элементов одномерного массива с нечётными номерами индексов. Счёт позиций начинается с единицы, счёт индексации – с нуля.

**Вариант №18**

Найти среднее арифметическое элементов одномерного массива.

**Вариант №19**

Найти сумму косинусов элементов одномерного массива, синусы которых отрицательны.

**Вариант №20**

Дан массив целых чисел. Сформировать массив этих же чисел, упорядоченных по убыванию, и массив индексов этих чисел в исходном массиве.

**Вариант №21**

В заданном одномерном массиве определить тройку наибольших чисел и вывести их в порядке убывания.

**Вариант №22**

В одновременном массиве найти наибольшее число, синус которого отрицателен.

**Вариант №23**

Определить позицию наибольшего элемента одномерного массива. Счёт позиций начинается с единицы, счёт индексации – с нуля.

**Вариант №24**

Упорядочить одномерный массив в порядке возрастания синусов элементов.

**Вариант №25**

Заполнить одномерный массив факториалами индексов его элементов. Наложить ограничения на ввод размерности такого массива и в случае выхода за диапазон допустимых значений сообщать об ошибке с предложением повторения ввода.

**Вариант №26**

Найти сумму элементов одномерного массива согласно следующему правилу: находится первый попавшийся положительный элемент и учитывается в сумме, от него делается отступ на следующую позицию вправо и, если там оказывается отрицательный элемент, то начинается поиск ближайшего положительного элемента, найденный положительный элемент суммируется (положительный элемент суммируется и для случая, если он был найден сразу после отступа на нужную позицию) и от него делается отступ уже в два элемента вправо и так далее. Для тестирования рассматривать массивы с размерностями не меньше 20.

**Вариант №27**

В одномерном массиве найти разность позиций наименьшего и наибольшего элементов. Обратить внимание, что счёт позиций начинается с единицы.

**Вариант №28**

В заданном одномерном массиве чисел найти количество элементов, удовлетворяющих условию .

**Вариант №29**

Преобразовать заданный одномерный массив по следующему правилу: .

**Вариант №30**

Найти среднее геометрическое элементов одномерного массива.