**Задание 08.** 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**

В заданном одномерном массиве вещественных чисел длины **n** найти количество элементов, удовлетворяющих условию .

**Вариант №2**

Определить количество различных (уникальных) целых чисел в одномерном массиве длины **n**.

**Вариант №3**

В одномерном массиве длины **n** найти разность индексов наибольшего и наименьшего элементов. Счёт индексации вести с нуля.

**Вариант №4**

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

**Вариант №5**

Переставить целые элементы одномерного массива длины **n** таким образом, чтобы положительные элементы в нём предшествовали отрицательным элементам. При этом взаимный порядок как положительных, так и отрицательных элементов должен быть сохранён. Например, исходным является массив *А = (1, -7, 3, 0, -9, -1, 5)*, результирующим – массив *B = (1, 3, 0, 5, -7, -9, -1)*.

**Вариант №6**

Заполнить одномерный массив длины **n** соответствующим количеством чисел из последовательности Фибоначчи.

**Вариант №7**

Переставить элементы одномерного массива длины **n** из вещественных чисел таким образом, чтобы элементы в нём чередовались по схеме «отрицательный-положительный». При этом взаимный порядок как положительных, так и отрицательных элементов должен быть сохранён. Например, исходным является массив *А = (1.0, -7.0, 3.0, 0.0, -9.0, -1.0, 5.0)*, результирующим – массив *B = (-7.0, 1.0, -9.0, 3.0, -1.0, 0.0, 5.0)*.

**Вариант №8**

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

**Вариант №9**

В заданном одномерном массиве вещественных чисел длины **n** определить позиции максимального и минимального значений. Счёт позиций принудительно начинается с единицы даже при условии, что индексация элементов массивов на языке *Visual C#* начинается с нуля.

**Вариант №10**

Сформировать новый одномерный массив вещественных чисел. Числа в массиве расположить порядке возрастания. Решение принимать в отношении только нечётных значений исходного одномерного массива длины **n**.

**Вариант №11**

Найти сумму только положительных вещественных элементов в одномерном массиве длины **n**.

**Вариант №12**

В заданном одномерном массиве вещественных чисел длины **n** найти количество элементов, удовлетворяющих условию .

**Вариант №13**

Сформировать одномерный массив соответствующей, корректной длины из элементов, упорядоченных по убыванию. К рассмотрению взять только нечётные элементы заданного одномерного массива длины **n** из целых чисел.

**Вариант №14**

Определить количество нулевых элементов одномерного массива длины **n**, стоящих на нечётных позициях. Обратить внимание, что счёт позиций начинается с единицы, а индексируется массив в *Visual C#* с нуля.

**Вариант №15**

Найти сумму элементов одномерного массива длины **n**, стоящих между первым и последним нулевыми элементами. Учесть ситуации недостачи и отсутствия нулей в массиве. *Подсказка*: не самым удачным тестовым примером для задачи является постановка нулей на первую и последнюю позиции массива.

**Вариант №16**

Определить общее количество нулевых элементов в заданном одномерном массиве длины **n** целых чисел и перечислить в отдельной строке номера их позиций. Обратить внимание, что счёт позиций начинается с единицы, а индексируется массив в *Visual C#* с нуля.

**Вариант №17**

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

**Вариант №18**

Найти среднее арифметическое элементов одномерного массива длины **n**.

**Вариант №19**

Заполнить одномерный массив длины **n** соответствующими значениями факториалов индексов. Счёт индексов вести с нуля.

**Вариант №20**

В одномерном массиве длины **n** найти разность позиций наименьшего и наибольшего элементов. В случае отрицательного результата при вычислении разности упорядочить элементы массива, находящиеся между найденными позициями, по убыванию. Обратить внимание, что счёт позиций начинается с единицы.

**Вариант №21**

Дан массив целых чисел длины **n**. Сформировать новый массив этих же чисел, но упорядоченных по убыванию. На этапе вывода сформированного массива в строке под ним разместить ещё один массив, содержащий в ячейках индексы этих же чисел, какими они были в исходном массиве.

**Вариант №22**

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

**Вариант №23**

В одномерном массиве целых чисел длины **n** найти наибольшее число, синус которого отрицателен.

**Вариант №24**

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

**Вариант №25**

Упорядочить одномерный массив вещественных чисел длины **n** по мере возрастания синусов элементов.

**Вариант №26**

Заполнить одномерный массив длины **n** факториалами индексов его элементов (при нулевом индексе – 1!; при единичном индексе – 2! и т.д.). Наложить ограничения на пользовательский ввод размерности такого массива и в случае выхода за диапазон допустимых значений сообщать пользователю об ошибке с предложением повторения ввода.

**Вариант №27**

Найти сумму элементов одномерного массива длины **n** согласно следующему правилу: находится первый попавшийся положительный элемент и он 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 длинами **n**, не меньшими **20**.

**Вариант №28**

Найти для одномерного массива длины **n** сумму косинусов элементов, синусы которых отрицательны.

**Вариант №29**

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

**Вариант №30**

Преобразовать заданный одномерный массив длины **n** по следующему правилу: .

**Вариант №31**

Найти среднее геометрическое элементов одномерного массива длины **n**.

**Вариант №32**

Определить количество нулевых элементов в заданном одномерном массиве целых чисел длины **n**.

**Вариант №33**

Определить количество элементов, превышающих заданное пороговое значение **delta**, в заданном одномерном массиве вещественных чисел длины **n**.

**Вариант №34**

В заданном одномерном массиве вещественных чисел размерности **n** выявить элементы, удовлетворяющие условию , и записать их новый массив той же размерности.

**Вариант №35**

В заданном одномерном массиве вещественных чисел размерности **n** выявить элементы, удовлетворяющие условию , и записать их новый массив c размерностью, не превышающей количества таких элементов.