**Задание 10.**

Решить поставленную задачу программирования по разделу «Многомерные массивы». Продумать организацию удобного ввода элементов многомерного массива. Продумать схему корректного вывода элементов многомерного массива.

**Элементы рабочего задания:**

1. **Учесть все возможные ограничения**, накладываемые не только на переменные, но и на участвующие в расчёте функциональные зависимости**[[1]](#footnote-1)**.

2. **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*.

3. **Подобрать корректные тестовые примеры в необходимом количестве[[2]](#footnote-2).** Принять во внимание тот факт, что все необходимые для тестирования значения могут быть занесены сразу в один единственный многомерный массив (рациональная размерность массива для тестирования – не менее трёх элементов в каждом из направлений). Помнить, что для решения задачи в общем виде размерность не должна оказаться одинаковой в каждом из измерений, кроме случаев, когда с точки зрения математики принципиальны квадратные и кубические массивы.

4. **Помнить о точности производимых вычислений**, например:

|  |  |
| --- | --- |
| **«При заданной точности  , но ,**  **то есть , а ».** | (1) |

5. Код составленного программного обеспечения **сопроводить комментариями** (элементы программистской этики)**[[3]](#footnote-3)**.

6. **Отчёт по решённой задаче составить** по схеме, изложенной в методических указаниях**[[4]](#footnote-4)**. 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

7. **Схему алгоритма оформлять** согласно правилам**[[5]](#footnote-5)**.

8. Материалы, подготавливаемые **к отправке по электронной почте**, оформляются согласно **Приложению 4** Сборника задач**[[6]](#footnote-6)**. Если материалы направляются на проверку через платформу *GitHub*, то **Приложением 4** разрешается не руководствоваться.

9. Настоятельно рекомендовано перед решением задачи **ознакомиться с рекомендациями**, собранными в **Приложении 3** Сборника задач**[[7]](#footnote-7)**.

10. Не забывать **давать значимые имена переменным**, проектам, решениям, файлам и другим разрабатываемым компонентам**[[8]](#footnote-8)**. При нежелании использовать значимые имена переменных в отчёте необходимо составить в разделе, содержащем формулировку задачи, таблицу соответствия вида:

Таблица 1 – Пример таблицы соответствия переменных, используемых в программе

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

11.Вывод массивов количеством измерений, большим двух, **реализовывать постранично / послойно с остановками по «Console.ReadKey(true);»**. Возврат к предыдущей странице не реализовывать – двигаться только вперёд.

12. После ввода с клавиатуры данных в массив – **очищать окно консоли по «Console.Clear();»**.

13. **Вывод** исходных и модифицированных массивов вести **строго друг под другом** для обеспечения возможности их сопоставления.

14. Для привлечения внимания к наиболее существенным изменениям элементов **рекомендуется не забывать про возможности изменения цветов фона и шрифта консоли**.

15. **Использовать опорную схему ввода-вывода**:

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

*«Введите количество строк двумерного массива: [целое числовое значение]»*

*«Введите количество столбцов двумерного массива: [целое числовое значение]»*

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

*«Введите 2-й элемент 1-й строки массива: [числовое значение]»*

*…*

*«Введите 1-й элемент 3-й строки массива: [числовое значение]»*

*«Введите 2-й элемент 3-й строки массива: [числовое значение]»*

*… … … …*

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

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

|*1.40 -0.10 3.33 ... -9.37*|

*[имя одномерного массива] =* |*6.38 5.13 6.78 ... -10.29*|

| *... ... ... ... ...*|

|*4.50 -7.02, 2.64, ..., -0.09*|

**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».**

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

**Вариант №1**

Перемножить матрицы *E* и *P*, где *Е* – единичная матрица, *P* – пользовательская матрица. Перемножение матриц реализовать поэлементно.

**Вариант №2**

Реализовать «выкройку» недиагональных элементов одной матрицы и «вклеивание» их поверх элементов другой матрицы. «Выкройка» подразумевает выставление на позициях элементов значений, равных «-1».

**Вариант №3**

Вычислить определитель (детерминант) любой матрицы размерностью большей, чем [3x3]. Учесть ограничение на размерность матрицы при указании соответствующего значения в процессе исполнения программы.

**Вариант №4**

Разложить заданную матрицу размерностью большей, чем [3x3] на миноры всех её элементов.

**Вариант №5**

Вывести трёхмерный массив целых чисел (со значениями из диапазона от 1 до 9 включительно) по спирали. Последний элемент *C[Z-1, Y-1, X-1]* расположить в центре окна консоли, остальные элементы «раскручивать» вокруг него по часовой стрелке до достижения первого элемента *C[0, 0, 0]*.

**Вариант №6**

Дана матрица размерностью [*NxM*], состоящая только из натуральных чисел. Выбрать в каждой строке матрицы свой наименьший элемент и переставить его в первый столбец той же строки.

**Вариант №7**

Реализовать «выкройку» диагональных элементов одной матрицы и «вклеивание» их поверх элементов другой матрицы. «Выкройка» подразумевает обнуление элементов.

**Вариант №8**

Выполнить возведение заданной матрицы в заданную степень при условии, что все операции умножения заменяются операциями сложения, а все операции сложения – операциями поиска минимального элемента между парой значений. Операции реализовать алгоритмически без переопределения операторов языка *Visual C#*.

**Вариант №9**

Выполнить возведение заданной матрицы в заданную степень. Перемножение произвести поэлементно, а не по известным правилам перемножения из курса «Высшей математики».

**Вариант №10**

Дана целочисленная матрица из *N* строк и *M* столбцов (*1 < N <= 25*, *1 < M <= 30*). При попытке пользовательского ввода значений, находящихся за рамками указанного диапазона, запрашивать повторный ввод значений. Выполнить поиск количества различных (неповторяющихся, уникальных) элементов матрицы.

**Вариант №11**

Вывести трёхмерный массив по спирали. Первый элемент *S[0, 0, 0]* расположить в центре окна консоли и раскручиваться против часовой стрелки в направлении последнего элемента *S[U-1, V-1, W-1]*.

**Вариант №12**

Выполнить транспонирование введённой матрицы алгоритмически, не прибегая к использованию стандартных методов языка *Visual C#*.

**Вариант №13**

Реализовать перелистывание слоёв трёхмерного массива клавишами, являющимися первыми буквами направлений: «*L*» – «влево» и «*R*» – «вправо». Выход за диапазон по слоям обозначать надписями вида: «(пусто)». Сформулированная задача отменяет общее требование о движении «только вперёд» по слоям массива.

**Вариант №14**

Вывести трёхмерный массив «змейкой» в консоли. Первый элемент массива *B[0, 0, 0]* расположить в левом верхнем углу окна консоли. Вертикальными границами смены направления «змейки» слева является естественная граница консольного окна, а справа – половина ширины консольного окна.

**Вариант №15**

Упорядочить двумерный массив целых чисел по возрастанию. В полученном массиве записать остатки от деления элементов по модулю 3.

**Вариант №16**

Выполнить возведение заданной матрицы размерностью [6x6] с нулевыми элементами на главной диагонали в седьмую степень при условии, что все операции умножения заменяются операциями сложения, а все операции сложения операциями поиска максимального элемента между парой значений. Полученную матрицу умножить на исходную, в которой элементы главной диагонали заменены на 1000.

**Вариант №17**

Выполнить возведение заданной матрицы в заданную степень при условии, что все операции умножения заменяются операциями вычитания, а все операции сложения операцией поиска максимального элемента между парой значений.

**Вариант №18**

Получить расчёт обратной матрицы для заданной, не прибегая к использованию стандартных методов языка *Visual C#*.

**Вариант №19**

Заполнить трёхмерный массив с помощью датчика псевдослучайных целых чисел таким образом, чтобы все элементы в пределах слоя оказались различными (уникальными).

**Вариант №20**

В заданной матрице найти: сумму минимального и максимального элементов; количество отрицательных элементов, стоящих на чётных позициях. Обратить внимание на то, что индексация считается с нуля, а позиции – с единицы.

**Вариант №21**

Заполнить матрицу размерности *n* автоматически через сочетание значений индексов по нижеследующему шаблону:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | ... | n-2 | n-1 | n |
| 2 | 1 | 2 | ... | n-3 | n-2 | n-1 |
| 3 | 2 | 1 | ... | n-4 | n-3 | n-2 |
| ... | ... | ... | ... | ... | ... | ... |
| n-1 | n-2 | n-3 | ... | 2 | 1 | 2 |
| n | n-1 | n-2 | ... | 3 | 2 | 1 |

**Вариант №22**

Заполнить трёхмерный массив с помощью датчика псевдослучайных целых чисел таким образом, чтобы абсолютно все элементы в нём оказались различными (уникальными).

**Вариант №23**

Сложить матрицы *E* и *P*, где *E* – единичная матрица, *P* – пользовательская матрица целых чисел. Вывести результат «змейкой», начинающейся справа снизу. Границами установить: горизонтальную – в ¾ высоты окна консоль, вертикальную – в ¼ ширины окна консоли.

**Вариант №24**

Вывести трёхмерный массив символов (со значениями из диапазона от «а» до «я» включительно) по спирали. Первый элемент *C[0, 0, 0]* расположить в центре окна консоли, остальные элементы «раскручивать» вокруг него против часовой стрелки. Подобрать в качестве примера такие элементы массива, которые при наличии в нём 36 элементов выведут на экран прореженные 6 слов, состоящих из 6 букв, читаемых слева направо.

**Вариант №25**

Дана матрица размерностью [*NxM*], состоящая только из натуральных чисел. Выбрать в каждой строке матрицы свой наименьший элемент и переставить его в первый столбец строки, зеркальной к рассматриваемой относительно границ массива. Отмеченная зеркальность математически записывается как *x(i) = x(N-i)*.

**Вариант №26**

Заполнить матрицу размерности *n* автоматически через значения сочетания индексов по следующему шаблону:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | ... | n-2 | n-1 | n |
| 0 | 1 | 2 | ... | n-3 | n-2 | n-1 |
| 0 | 0 | 1 | ... | n-4 | n-3 | n-2 |
| ... | ... | ... | ... | ... | ... | ... |
| 0 | 0 | 0 | ... | 0 | 1 | 2 |
| 0 | 0 | 0 | ... | 0 | 0 | 1 |

**Вариант №27**

Упорядочить двумерный массив по убыванию. В полученном массиве заменить все элементы с чётными индексами значениями 1 / *x*, где *x* – значение элемента.

**Вариант №28**

Заполнить трёхмерный массив с помощью датчика псевдослучайных чисел таким образом, чтобы все элементы в пределах только центрального слоя оказались различными (уникальными). Если ярко выраженный центральный слой при указанной размерности отсутствует, то уникальными значениями заполнить пару срединных слоёв.

**Вариант №29**

Реализовать обмен элементами двух матриц по следующему правилу: «треугольник» над главной диагональю первой матрицы переносится в «треугольник» под главной диагональю второй матрицы и «треугольник» над главной диагональю второй матрицы переносится в «треугольник» под главной диагональю первой матрицы.

**Вариант №30**

Вывести трёхмерный массив целых чисел (со значениями из диапазона от -99 включительно до 100 не включительно) по спирали. Последний элемент *Z[A-1, B-1, C-1]* расположить в центре окна консоли, остальные элементы «раскручивать» вокруг него по часовой стрелке.

1. Пример корректного подбора тестовых примеров в нужном количестве представлен в разделе 7 Сборника задач [Сафронов, А. И. Получение первичных профессиональных умений и навыков научно-исследовательской деятельности: Сборник задач для проведения аудиторных занятий по учебной практике / А. И. Сафронов, Н. Н. Зольникова, В. Г. Новиков. – Москва: РУТ. – 2019. – 91 с. – *EDN SXMWOD*]. Скачать можно на сайте Национальной Электронной Библиотеки (НЭБ) *elibrary.ru*: [*https://www.elibrary.ru/item.asp?id=46307421*](https://www.elibrary.ru/item.asp?id=46307421). [↑](#footnote-ref-1)
2. То же, что и сноска 1. [↑](#footnote-ref-2)
3. Пункт 6 на странице 391 [Сафронов, А. И. Способы проектирования эргономичных графических пользовательских интерфейсов для интеллектуальных транспортных систем / А. И. Сафронов // Интеллектуальные транспортные системы: Материалы III Международной научно-практической конференции, Москва, 30 мая 2024 года. – Москва: РУТ. – 2024. – С. 385-394. – *DOI 10.30932/9785002446094-2024-385-394*. – *EDN TGVERU*]. Скачать можно на сайте Национальной Электронной Библиотеки (НЭБ) *elibrary.ru*: [*https://www.elibrary.ru/item.asp?id=68533620*](https://www.elibrary.ru/item.asp?id=68533620). [↑](#footnote-ref-3)
4. Пример оформления отчётной документации [Сафронов, А. И. Составление отчётной документации по решённым задачам алгоритмизации и программирования: Учебно-методическое пособие для проведения аудиторных занятий по учебной практике / А. И. Сафронов, Н. Н. Зольникова, В. Г. Новиков. – Москва: РУТ. – 2018. – 83 с. – *EDN WNAIFP*]. Скачать можно на сайте Национальной Электронной Библиотеки (НЭБ) *elibrary.ru*: [*https://www.elibrary.ru/item.asp?id=46271697*](https://www.elibrary.ru/item.asp?id=46271697). [↑](#footnote-ref-4)
5. Правила изложены в разделе 8 того же учебно-методического пособия, что указано по сноске 4. [↑](#footnote-ref-5)
6. То же, что и сноска 1. [↑](#footnote-ref-6)
7. То же, что и сноска 1. [↑](#footnote-ref-7)
8. То же, что и сноска 3. [↑](#footnote-ref-8)