

Контрольное домашнее задание, модуль 1

Контрольное домашнее задание предполагает самостоятельную домашнюю работу. Вам потребуется:

1. Изучить предложенные теоретические материалы самостоятельно.
2. Самостоятельно поработать с документацией по языку C#, в т.ч. осуществлять информационный поиск.
3. Разработать программы, определённые основной задачей и индивидуальным вариантом.
4. Сдать заархивированное решение, содержащее два проекта с решениями задач в SmartLMS вовремя.

Формат сдачи работы

Для проверки предоставляется решение, содержащие два проекта консольных приложений. Решение должно быть заархивировано и приложено к заданию в SmartLMS.

Срок выполнения и загрузки работы

Две недели (фактический дедлайн смотреть по SmartLMS)

Дедлайн является мягким и еще на протяжении суток работу можно будет отправить на проверку, с учётом штрафов. Если работа направлена на проверку не более чем через час после дедлайна, то максимальная оценка за нее 8 баллов. Работы, присланные позже, чем через час после дедлайна штрафуются баллом (при максимуме 8) от возможной оценки за каждые 2 часа просрочки.

Основная задача

Обязательно ознакомьтесь с теоретическими материалами для выполнения данной работы.

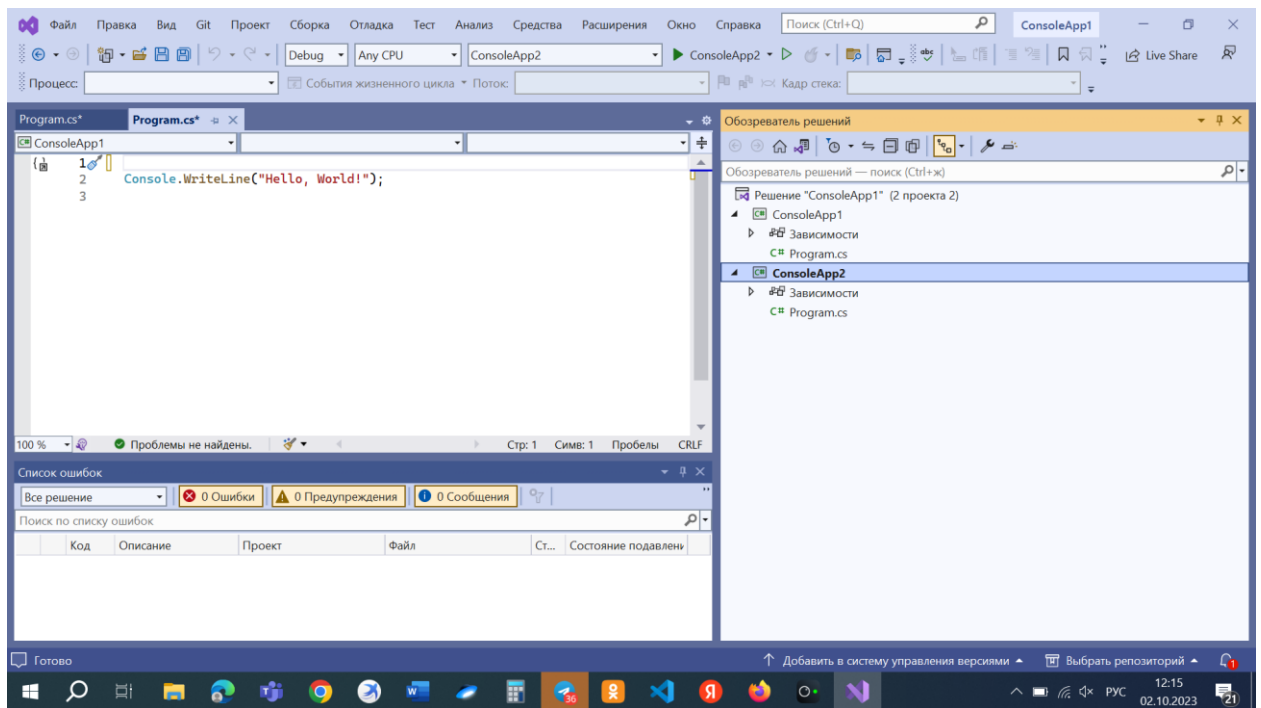
В одном решении разработать два проекта консольных приложений

Для добавления в текущее решение, содержащего Проект 1, нового Проекта 2 выполняем действия (для IDE Visual Studio):

1. В меню Файл выбрать пункт Добавить -> Новый проект и создать его.
2. В файле Program.cs Проекта 2 разместить код программы.
3. Для запуска на трансляцию исходного модуля Проекта 2 необходимо назначить данный проект текущим (активным). Для этого кликнув на название проекта вызываем правой кнопкой мыши контекстное меню, пункт «Назначить в качестве запускаемого проекта».

В примере проект ConsoleApp2 активен и выделен в обозревателе решений жирным шрифтом.

Обратите внимание, что название решения и проектов должны быть осмысленными, а не автоматически назначаемыми средой разработки.



Первый проект (программа 1):

1. создаёт *структуру данных A*, определённую индивидуальным вариантом (таблица 1, столбец 2);
2. назначает элементам *A* значения, заданные правилом из индивидуального варианта (таблица 1, столбец 3), используя **метод**, в который структура данных *A* передаётся по правилу, определённому индивидуальным вариантом (таблица 1, столбец 4);
3. создаёт текстовый файл (правило именования файла определено индивидуальным вариантом (таблица 1, столбец 5);
4. размещает в файле, определённом п. 3 требований к программе 1, значения элементов *структуры данных A*, правило форматирования элементов в файле определено индивидуальным вариантом (таблица 1, столбец 5);

Второй проект (программа 2):

1. получает имена файлов для чтения от пользователя и читает файлы формата, определённого требованиями к формату выходных данных программы 1 (таблица 1, столбец 5);
2. если файл не обнаружен на диске / не открывается по произвольным причинам: программа выводит информационное сообщение пользователю (текст строки определите самостоятельно);
3. если нарушен формат данных в файле с данными: программа не извлекает данные и не размещает их в структурах данных программы 2, а выводит информационное сообщение для пользователя (текст строки определите самостоятельно);
4. если файл существует на диске, успешно открылся и содержит данные допустимого формата: программа создаёт *структуру данных B*, определённую индивидуальным вариантом (таблица 2, столбец 2) и загружает в неё элементы из файла;
5. элементы *структуры данных B*, программа обрабатывает по правилу, определённому индивидуальным вариантом (таблица 2, столбец 3), для чего в программе должен быть описан метод обработки, куда структура данных *B* передаётся по заданному индивидуальным вариантом (таблица 2, столбец 4) правилу, .
6. выводит данные до и после изменений на экран в отформатированном виде, то есть по строкам, разделяя элементы одним пробелом и форматируя с точностью, заданной при создании файла в проекте 1.

Требования по совместимости и качеству для основной задачи:

- весь программный код должен быть написан на языке программирования C# с учётом использования .net 6.0;
- исходный код должен содержать комментарии, объясняющие неочевидные фрагменты и решения, резюме кода, описание целей кода (см. материалы лекции 1);
- использованные в программе идентификаторы должны соответствовать правилам и соглашениям об именовании идентификаторов C# (<https://learn.microsoft.com/ru-ru/dotnet/csharp/fundamentals/coding-style/identifier-names>);
- представленный к проверке код должен отвечать общим соглашениям о коде C# Microsoft (<https://learn.microsoft.com/ru-ru/dotnet/csharp/fundamentals/coding-style/coding-conventions>);
- при перемещении папки проекта (копировании / переносе на другое устройство) файлы должны открываться программой также успешно, как и на компьютере создателя, т.е. вне зависимости от путей;
- если загрузка данных происходит из непрямоугольного представления в прямоугольное, то недостающие элементы справа достраиваются нулевыми вещественными значениями;
- программа не допускает пользователя до решения задач, пока вводятся некорректные данные с клавиатуры;
- программа обрабатывает исключительные ситуации, связанные (1) со вводом и преобразованием / приведением данных как с клавиатуры, так и из файлов; (2) с созданием, инициализацией, обращением к элементам массивов и строк.
- обе представленные к проверке программы должны решать все поставленные задачи, успешно компилироваться, не закрываться без пояснений при некорректных вариантах работы и не завешаться аварийно.

Теоретические материалы для самостоятельной работы

Для работы с файлами и проверки существования / возможностей доступа к файлу используйте класс File (<https://learn.microsoft.com/ru-ru/dotnet/api/system.io.file?view=net-6.0>)

Для записи данных в текстовый файл воспользуйтесь следующими материалами:

- <https://learn.microsoft.com/ru-ru/dotnet/api/system.io.file.writealllines?view=net-6.0>
- <https://learn.microsoft.com/ru-ru/dotnet/api/system.io.file.writealltext?view=net-6.0>

Для организации чтения данных из файла воспользуйтесь следующими материалами:

- <https://learn.microsoft.com/ru-ru/dotnet/api/system.io.file.readalllines?view=net-6.0>

Для разделения строк на подстроки воспользуйтесь методом Split():

- <https://learn.microsoft.com/ru-ru/dotnet/csharp/how-to/parse-strings-using-split>

Индивидуальные варианты

Таблица 1. Индивидуальные варианты для проекта консольного приложения 1 (программа 1)

№	Структура данных А	Правило назначения элементов структуры данных А	Правило передачи структуры данных А в метод	Требования к выходному файлу и формату представления данных из А в файле						
1	2	3	4	5						
1	Двумерный массив NxM вещественных значений	<p>$0 < N \leq 15; 0 < M \leq 10$ – целые, вводит с клавиатуры пользователь. Значения элементов А назначаются подряд (без учета переходов по измерениям) по формуле:</p> $\frac{2n + 10}{3n - 10}, n \geq 1, \text{ целое}$ <p>Например, для N = 2 и M = 3 сформированный двумерный массив (пример с округлением данных для удобства представления в задании) может выглядеть примерно так:</p> <table><tr><td>-1,71</td><td>-3,5</td><td>-16</td></tr><tr><td>9</td><td>4</td><td>3,14</td></tr></table>	-1,71	-3,5	-16	9	4	3,14	<p>Метод возвращает тип void, А передана в метод по ссылке с модификатором out; N и M передаются в метод в качестве параметров</p> <p>(Справочник по C#. Параметры методов - C# Microsoft Learn)</p>	<p>Имя файла – строка, заданная пользователем с клавиатуры.</p> <p>Место размещение файла – папка с исполнимым файлом консольного приложения 1.</p> <p>При повторных запусках решения на новых входных данных создаётся новый файл для данных; имя запрашивается у пользователя повторно. Если файл уже существует, данные в нем перезаписываются.</p> <p>Данные об одном двумерном массиве пишутся в одну строку; элементы каждой строки двумерного массива записываются, разделяясь ровно одним пробелом, числа отформатированы с точностью 2 знака после десятичного разделителя; строки разделены символом точка с запятой.</p> <p>Например, для примера из столбца 3: -1,71 -3,50 -16,00; 9,00 4,00 3,14;</p>
-1,71	-3,5	-16								
9	4	3,14								
2	Двумерный массив NxM вещественных значений	<p>$0 < N \leq 15; 0 < M \leq 10$ – целые, вводит с клавиатуры пользователь. Значения элементов А назначаются подряд (без учета переходов по измерениям) по формуле:</p> $\frac{\sqrt{n^2 - 5n + 6}}{n - 2}, n > 5 \text{ целое}$	<p>Метод возвращает ссылку на структуру А, N и M – параметры метода</p> <p>(Справочник по C#. Параметры методов - C# Microsoft Learn).</p>	<p>Имя файла – строка, заданная пользователем с клавиатуры.</p> <p>Место размещение файла – папка с исполнимым файлом консольного приложения 1.</p> <p>При повторных запусках решения на новых входных данных создаётся новый файл для данных; имя запрашивается у пользователя повторно. Если файл уже существует, данные в нем перезаписываются.</p>						

		<p>Например, для $N = 2$ и $M = 3$ сформированный двумерный массив (пример с округлением данных для удобства представления в задании) может выглядеть примерно так:</p> <p>2,11 1,95 1,82 1,71 1,63 1,57</p>		<p>Данные об одном двумерном массиве пишутся по одному элементу на каждой строке файла; элементы каждой строки двумерного массива разделяются ровно одной пустой строкой, числа отформатированы с точностью 2 знака после десятичного разделителя;</p> <p>Например, для примера из столбца 3:</p> <p>2,11 1,95 1,82</p> <p>1,71 1,63 1,57</p>
3	Массив из N одномерных вещественных массивов	<p>$1 \leq N < 21$ – целое, вводит с клавиатуры пользователь. Длины измерений одномерных массивов, составляющих A – случайные значения из диапазона $[1;15]$. Значения элементов A назначаются подряд (без учета переходов по массивам, составляющим массив) по формуле:</p> $\frac{2\sqrt{n+3}}{3n-1}, n \geq 1 \text{ целое}$ <p>Например, $N = 3$ сформированные массивы массивов (пример с округлением данных для удобства представления в задании) могут выглядеть примерно так:</p>	<p>Метод возвращает тип void, ссылка на A передается в него с модификатором out; N – параметр метода.</p> <p>(Справочник по C#. Параметры методов - C# Microsoft Learn)</p>	<p>Имя файла – строка, заданная пользователем с клавиатуры. Место размещение файла – папка с исполнимым файлом консольного приложения 1. При повторных запусках решения на новых входных данных создаётся новый файл для данных; имя запрашивается у пользователя повторно. Если файл уже существует, данные в нем перезаписываются. Данные об одном массиве массивов пишутся по правилу: На первой строке целое число элементов массива массивов (количество одномерных массивов). Со следующей строки: на каждой строке файла массив, на который указывает очередная ссылка массива ссылок; элементы разделяются ровно одним пробелом, числа отформатированы с точностью 2 знака после десятичного разделителя;</p> <p>Например, для примера из столбца 3:</p>

		2,00 0,89 0,61 0,48 0,40 0,35 0,32 0,29 0,27		3 2,00 0,89 0,61 0,48 0,40 0,35 0,32 0,29 0,27
4	Массив из N одномерных вещественных массивов	<p>$1 \leq N < 21$ – целое, вводит с клавиатуры пользователь. Длины измерений одномерных массивов, составляющих A – случайные значения из диапазона [1;15). Значения элементов A назначаются подряд (без учета переходов по массивам, составляющим массив) по формуле:</p> $\frac{\sqrt{n^3 + 3}}{n^2 + 5}, n \geq 1 \text{ целое}$ <p>Например, N = 3 сформированные массивы массивов (пример с округлением данных для удобства представления в задании) могут выглядеть примерно так: 0,67 0,74 0,78 0,78 0,75 0,72 0,69 0,66 0,63</p>	<p>Метод возвращает тип void, ссылка на A передана в метод по ссылке out; N – параметр метода. (Справочник по C#. Параметры методов - C# Microsoft Learn)</p>	<p>Имя файла – строка, заданная пользователем с клавиатуры. Место размещение файла – папка с исполнимым файлом консольного приложения 1. При повторных запусках решения на новых входных данных создается новый файл для данных; имя запрашивается у пользователя повторно. Если файл уже существует, данные в нем перезаписываются. Данные об одном массиве массивов пишутся по правилу: На первой строке целое число (размерность массива ссылок). Со следующей строки: на каждой строке файла массив, на который указывает очередная ссылка массива ссылок; элементы разделяются ровно двумя символами ?? без пробелов, в конце строки символ ?; числа отформатированы с точностью 3 знака после десятичного разделителя; Например, для примера из столбца 3: 3 0,667??0,737??0,782? 0,780??0,754??0,722??0,689? 0,658??0,629?</p>
5	Массив из N вещественных массивов размера M элементов каждый	<p>$0 < N \leq 15$; $0 < M \leq 10$ – целые, вводит с клавиатуры пользователь Значения элементов A назначаются подряд (без учета переходов по измерениям) по формуле:</p>	<p>Метод возвращает ссылку на структуру A, N и M – параметры метода (Справочник по C#. Параметры методов - C# Microsoft Learn)</p>	<p>Имя файла – строка, заданная пользователем с клавиатуры. Место размещение файла – папка с исполнимым файлом консольного приложения 1. При повторных запусках решения на новых входных данных создается новый файл для данных; имя</p>

		$\frac{2n^2 + n - 3}{n^3 - n^2}, n > 1 \text{ целое}$ <p>Например, N = 3 и M = 3 сформированные массивы массивов (пример с округлением данных для удобства представления в задании) могут выглядеть примерно так: 1,75 1,00 0,69 0,52 0,42 0,35 0,30 0,26 0,23</p>		<p>запрашивается у пользователя повторно. Если файл уже существует, данные в нем перезаписываются. На каждой строке файла подряд сохраняются элементы одномерных массивов, входящих в массив массивов. При этом элементы одномерных массивов разделяются ровно двумя символами ;; без пробелов , в данных об одномерном массиве размещён один символ % без перевода строки; числа отформатированы с точностью 2 знака после десятичного разделителя.</p> <p>Например, для примера из столбца 3: 1,75;;1,00;;0,69%0,52;;0,42;;0,35%0,30;;0,26;;0,23%</p>
6	Массив из N вещественных массивов размера M элементов каждый	<p>$0 < N \leq 15; 0 < M \leq 10$ – целые, вводит с клавиатуры пользователь Значения элементов A назначаются подряд (без учета переходов по измерениям) по формуле:</p> $\frac{n^3}{n^2 + 1} - \frac{3n^2}{3n - 1}, n \geq 1 \text{ целое}$ <p>Например, N = 3 и M = 3 сформированные массивы массивов (пример с округлением данных для удобства представления в задании) могут выглядеть примерно так: -0,50 -0,40 -0,30 -0,24 -0,19 -0,16 -0,14 -0,12 -0,11</p>	Метод возвращает ссылку на структуру A, N и M – параметры метода (Справочник по C# . Параметры методов - C# Microsoft Learn)	<p>Имя файла – строка, заданная пользователем с клавиатуры. Место размещение файла – папка с решением, выше уровня обоих папок проектов. При повторных запусках решения на новых входных данных создаётся новый файл для данных; имя запрашивается у пользователя повторно. Если файл уже существует, данные в нем перезаписываются. На первой строке целое число (размерность массива ссылок). Со следующей строки: на каждой строке файла массив, на который указывает очередная ссылка массива ссылок; элементы разделяются ровно двумя символами ?? без пробелов, в конце строки единственный символ ?; числа отформатированы с точностью 3 знака после десятичного разделителя; Между строками массива массивов пустая строка. Например, для примера из столбца 3: 3 -0,500??-0,400??-0,300? -0,235??-0,192??-0,162?</p>

				-0,140??-0,123??-0,110?
7	Двумерный массив NxN вещественных значений	<p>$0 < N \leq 15$ – целое, вводит с клавиатуры пользователь. Значения элементов A назначаются подряд (без учета переходов по измерениям) по формуле:</p> $\frac{(2 - n)^3}{(n + 1)^2 - (n + 1)^3}, n > 1 \text{ целое}$ <p>Например, для $N = 3$ двумерный массив (пример с округлением данных для удобства представления в задании) может выглядеть примерно так:</p> <pre>-0,25 -0,00 0,02 0,08 0,15 0,22 0,28 0,33 0,38</pre>	Метод возвращает ссылку на структуру A, N передается в метод в качестве параметра (Справочник по C# . Параметры методов - C# Microsoft Learn)	<p>Имя файла – строка, заданная пользователем с клавиатуры.</p> <p>Место размещение файла – папка с решением, выше уровня обоих папок проектов.</p> <p>При повторных запусках решения на новых входных данных создаётся новый файл для данных; имя запрашивается у пользователя повторно. Если файл уже существует, данные в нем перезаписываются.</p> <p>Данные об одном двумерном массиве пишутся по одному элементу на каждой строке файла; элементы каждой строки двумерного массива разделяются ровно одной пустой строкой, числа отформатированы с точностью 2 знака после десятичного разделителя; Например, для примера из столбца 3:</p> <pre>-0,25 -0,00 0,02 0,08 0,15 0,22 0,28 0,33 0,38</pre>
8	Массив из N вещественных одномерных массивов размера N элементов каждый	<p>$0 < N \leq 15$ – целое, вводит с клавиатуры пользователь. Значения элементов A назначаются подряд (без учета переходов по измерениям) по формуле</p>	Метод возвращает ссылку на структуру A, N передается в метод в качестве параметра (Справочник по C# . Параметры методов - C# Microsoft Learn)	<p>Имя файла – строка, заданная пользователем с клавиатуры.</p> <p>Место размещение файла – папка с исполнимым файлом консольного приложения 1.</p> <p>При повторных запусках решения на новых входных данных создаётся новый файл для данных; имя</p>

		$\frac{2n^2 + n - 3}{\sqrt[3]{n^3 - n^2}}, n > 1 \text{ целое}$ <p>Например, N = 3 сформированные массивы массивов (пример с округлением данных для удобства представления в задании) могут выглядеть примерно так: 4,41 6,87 9,08 11,20 13,28 15,34 17,38 19,41 21,44</p>		<p>запрашивается у пользователя повторно. Если файл уже существует, данные в нем перезаписываются. На каждой строке файла массив, на который указывает очередная ссылка массива ссылок; элементы разделяются ровно двумя символами ;; без пробелов , в конце строки символ %; числа отформатированы с точностью 2 знака после десятичного разделителя; между строками массива массивов пустая строка. Например, для примера из столбца 3: 4,41;;6,87;;9,08%</p> <p>11,20;;13,28;;15,34%</p> <p>17,38;;19,41;;21,44%</p>
9	Массив из N вещественных одномерных массивов размера N элементов каждый	<p>$0 < N \leq 31$ – целое, вводит с клавиатуры пользователь. Значения элементов A назначаются подряд (без учета переходов по измерениям) по формуле:</p> $\sqrt{2n + 3} - \sqrt{n + 1}, n \geq 1 \text{ целое}$ <p>Например, N = 3 сформированные массивы массивов (пример с округлением данных для удобства представления в задании) могут выглядеть примерно так: 0,91 1,00 1,08 1,16 1,23 1,29 1,36 1,42 1,48</p>	<p>Метод возвращает тип void, ссылка на A передана в метод по ссылке out, N – параметр метода (Справочник по C#, Параметры методов - C# Microsoft Learn)</p>	<p>Имя файла – строка, заданная пользователем с клавиатуры. Место размещение файла – папка с решением, выше уровня обоих папок проектов. При повторных запусках решения на новых входных данных создаётся новый файл для данных; имя запрашивается у пользователя повторно. Если файл уже существует, данные в нем перезаписываются. На каждой строке файла массив, на который указывает очередная ссылка массива ссылок; элементы разделяются ровно двумя символами ++ без пробелов , в конце строки символ %; числа отформатированы с точностью 4 знака после десятичного разделителя; Например, для примера из столбца 3: 0,9137++1,0000++1,0806% 1,1561++1,2272++1,2947% 1,3589++1,4203++1,4792%</p>
10	Двумерный массив NxN	<p>$0 < N \leq 13$ – целое, вводит с клавиатуры пользователь</p>	<p>Метод возвращает ссылку на структуру A, N передается в</p>	<p>Имя файла – строка, заданная пользователем с клавиатуры.</p>

	вещественных значений	<p>Значения элементов A назначаются подряд (без учета переходов по измерениям) по формуле:</p> $n + \sqrt[3]{3 - n^3}, n > 0 \text{ целое}$ <p>Например, для $N = 3$ двумерный массив (пример с округлением данных для удобства представления в задании) может выглядеть примерно так:</p> <p>2,26 0,29 0,12 0,06 0,04 0,03 0,02 0,02 0,01</p>	<p>метод в качестве параметра (Справочник по C#. Параметры методов - C# Microsoft Learn)</p>	<p>Место размещение файла – папка с исполнимым файлом консольного приложения 1.</p> <p>При повторных запусках решения на новых входных данных создаётся новый файл для данных; имя запрашивается у пользователя повторно. Если файл уже существует, данные в нем перезаписываются.</p> <p>Элементы строк многомерного массива размещаются в отдельных строках файла и разделяются ровно одним пробелом, каждая строка завершается одним знаком точка с запятой, числа отформатированы с точностью 2 знака после десятичного разделителя.</p> <p>Например, для примера из столбца 3:</p> <p>2,26 0,29 0,12; 0,06 0,04 0,03; 0,02 0,02 0,01;</p>
11	Двумерный массив NxN вещественных значений	<p>$0 < N \leq 14$ – целое, вводит с клавиатуры пользователь</p> <p>Значения элементов A назначаются подряд (без учета переходов по измерениям) по формуле:</p> $\left(\frac{n+5}{n+1}\right)^n, n > 0 \text{ целое}$ <p>Например, для $N = 3$ двумерный массив (пример с округлением данных для удобства представления в задании) может выглядеть примерно так:</p> <p>3,00 5,44 8,00 10,50 12,86 15,06 17,09 18,95 20,66</p>	<p>Метод возвращает тип void, ссылка на A передана в метод по ссылке out, N – параметр метода (Справочник по C#. Параметры методов - C# Microsoft Learn)</p>	<p>Имя файла – строка, заданная пользователем с клавиатуры.</p> <p>Место размещение файла – папка с решением, выше уровня обоих папок проектов.</p> <p>При повторных запусках решения на новых входных данных создаётся новый файл для данных; имя запрашивается у пользователя повторно. Если файл уже существует, данные в нем перезаписываются.</p> <p>На первой строке файла два целых числа, разделенные ровно одним пробелом (количество строк и количество столбцов структуры данных A). Данные о двумерном массиве пишутся со следующей строки по правилу: Одна строка файла – одна строка A, элементы разделяются ровно одним знаком '*', числа отформатированы с точностью 2 знака после десятичного разделителя;</p> <p>Например, для примера из столбца 3:</p>

				3 3 3,00*5,44*8,00 10,50*12,86*15,06 17,09*18,95*20,66
12	Массив из N одномерных вещественных массивов размера M каждый	<p>$0 < N \leq 13; 0 < M \leq 17$ – целые, вводит с клавиатуры пользователь</p> <p>Значения элементов A назначаются подряд (без учета переходов по измерениям) по формуле:</p> $\left(\frac{6n^2 + 3n + 2}{6n^2 + 3n + 3}\right)^n, n \geq 0 \text{ целое}$ <p>Например, N = 3 и M = 4 сформированные массивы массивов (пример с округлением данных для удобства представления в задании) могут выглядеть примерно так:</p> <p>1,00 0,92 0,94 0,96 0,96 0,97 0,97 0,98 0,98 0,98 0,98 0,99</p>	<p>Метод возвращает тип void, ссылка A передана в метод с модификатором out; N и M также передаются в метод в качестве параметров</p> <p>(Справочник по C#. Параметры методов - C# Microsoft Learn)</p>	<p>Имя файла – строка, заданная пользователем с клавиатуры.</p> <p>Место размещение файла – папка с решением, выше уровня обоих папок проектов.</p> <p>При повторных запусках решения на новых входных данных создаётся новый файл для данных; имя запрашивается у пользователя повторно. Если файл уже существует, данные в нем перезаписываются.</p> <p>На первой строке файла указана размерность массива ссылок (целое число)</p> <p>Далее на каждой строке файла массив, на который указывает очередная ссылка массива ссылок; элементы разделяются ровно двумя пробелами; числа отформатированы с точностью 3 знака после десятичного разделителя;</p> <p>Например, для примера из столбца 3:</p> <p>3 1,000 0,917 0,940 0,955 0,964 0,971 0,975 0,978 0,981 0,983 0,984 0,986</p>
13	Двумерный массив NxM вещественных значений	<p>$0 < N \leq 15; 0 < M \leq 10$ – случайные целые значения из указанных диапазонов.</p> <p>Значения элементов A назначаются подряд (без учета переходов по измерениям) по формуле:</p> $\sqrt{n^3 + 1} - \sqrt{n^2}, n \geq 1 \text{ целое}$	<p>Метод возвращает тип void, ссылка A передана в метод с модификатором out; N и M также передаются в метод в качестве параметров</p> <p>(Справочник по C#. Параметры методов - C# Microsoft Learn)</p>	<p>Имя файла – строка, заданная пользователем с клавиатуры.</p> <p>Место размещение файла – папка с исполнимым файлом консольного приложения 1.</p> <p>При повторных запусках решения на новых входных данных создаётся новый файл для данных; имя запрашивается у пользователя повторно. Если файл уже существует, данные в нем перезаписываются.</p>

		<p>Например, для $N = 2$ и $M = 3$ сформированный двумерный массив (пример с округлением данных для удобства представления в задании) может выглядеть примерно так:</p> <pre>0,41 1,59 3,56 6,06 8,99 12,28</pre>		<p>Элементы строк многомерного массива размещаются в отдельных строках файла и разделяются символом точка с запятой, каждая строка завершается одним знаком точка с запятой, числа отформатированы с точностью 3 знака после десятичного разделителя</p> <p>Например, для примера из столбца 3:</p> <pre>0,414;1,586;3,559; 6,062;8,989;12,281;</pre>
14	Двумерный массив NxN вещественных значений	<p>$0 < N \leq 15$ – целое, вводит с клавиатуры пользователь</p> <p>Значения элементов A назначаются подряд (без учета переходов по измерениям) по формуле:</p> $\frac{n \cdot \cos(n)}{n^2 + 1}, n \geq 1 \text{ целое}$ <p>Например, для $N = 3$ двумерный массив (пример с округлением данных для удобства представления в задании) может выглядеть примерно так:</p> <pre>0,27 -0,17 -0,30 -0,15 0,05 0,16 0,11 -0,02 -0,10</pre>	<p>Метод возвращает тип void, ссылка на A передана в метод по ссылке out; N параметр метода (Справочник по C#. Параметры методов - C# Microsoft Learn)</p>	<p>Имя файла – строка, заданная пользователем с клавиатуры.</p> <p>Место размещение файла – папка с исполнимым файлом консольного приложения 1.</p> <p>При повторных запусках решения на новых входных данных создаётся новый файл для данных; имя запрашивается у пользователя повторно. Если файл уже существует, данные в нем перезаписываются.</p> <p>На первой строке файла два целых числа, разделенные ровно одним пробелом (количество строк и количество столбцов матрицы). Данные об одном двумерном массиве пишутся со следующей строки по правилу: Одна строка файла– одна строка матрицы, элементы разделяются ровно одним знаком '*' без пробелов, в конце строки – два символа *, строка завершается также двумя символами *, числа отформатированы с точностью 2 знака после десятичного разделителя.</p> <p>Например, для примера из столбца 3:</p> <pre>3 3 0,27*-0,17*-0,30** -0,15*0,05*0,16** 0,11*-0,02*-0,10**</pre>

15	Массив из N массивов вещественных значений размера N элементов каждый	<p>$0 < N \leq 21$ – целое, вводит с клавиатуры пользователь. Значения элементов A назначаются подряд (без учета переходов по измерениям) по формуле:</p> $\frac{n^2 \cdot \sin(n)}{n^2 + 1}, n > 0 \text{ целое}$ <p>Например, N = 3 сформированные массивы массивов (пример с округлением данных для удобства представления в задании) могут выглядеть примерно так: 0,4207 0,7274 0,1270 -0,7123 -0,9220 -0,2719 0,6438 0,9741 0,4071</p>	<p>Метод возвращает тип void, ссылка на A передана в метод по ссылке out; N параметр метода (Справочник по C#. Параметры методов - C# Microsoft Learn)</p>	<p>Имя файла – строка, заданная пользователем с клавиатуры. Место размещение файла – папка с исполнимым файлом консольного приложения 1. При повторных запусках решения на новых входных данных создается новый файл для данных; имя запрашивается у пользователя повторно. Если файл уже существует, данные в нем перезаписываются. На первой строке файла указана размерность массива ссылок (целое число) Далее на каждой строке файла массив, на который указывает очередная ссылка массива ссылок; элементы разделяются ровно двумя пробелами; в конце строки ставится точка с запятой; числа отформатированы с точностью 4 знака после десятичного разделителя; Например, массив из двух ссылок на массивы длиной 2 и 3 элемента записывается в таком порядке: 3 0,4207 0,7274 0,1270; -0,7123 -0,9220 -0,2719; 0,6438 0,9741 0,4071;</p>
16	Двумерный массив NxN вещественных значений	<p>$0 < N \leq 17$ – целое, вводит с клавиатуры пользователь. Значения элементов A назначаются подряд (без учета переходов по измерениям) по формуле:</p> $\frac{(n-1)^3 - n^3}{2n^2 + n - 1}, n \geq 0 \text{ целое}$ <p>Например, для N = 3 двумерный массив (пример с округлением данных для удобства представления</p>	<p>Метод возвращает тип void, ссылка A передана в метод с модификатором out; N также передается в метод в качестве параметра (Справочник по C#. Параметры методов - C# Microsoft Learn)</p>	<p>Имя файла – строка, заданная пользователем с клавиатуры. Место размещение файла – папка с исполнимым файлом консольного приложения 1. При повторных запусках решения на новых входных данных создается новый файл для данных; имя запрашивается у пользователя повторно. Если файл уже существует, данные в нем перезаписываются. На первой строке файла два целых числа, разделенные ровно одним пробелом (количество строк и количество столбцов матрицы). Данные об одном двумерном массиве пишутся со следующей строки по правилу:</p>

		<p>в задании) может выглядеть примерно так:</p> <p>-0,50 -0,78 -0,95 -1,06 -1,13 -1,18 -1,22 -1,25 -1,28</p>		<p>Одна строка файла – одна строка матрицы, элементы разделяются ровно одним знаком '*' без пробелов, числа отформатированы с точностью 3 знака после десятичного разделителя.</p> <p>Например, для примера из столбца 3:</p> <p>3 3 -0,500*-0,778*-0,950 -1,057*-1,130*-1,182 -1,221*-1,252*-1,276</p>
17	Двумерный массив NxM вещественных значений	<p>$0 < N \leq 17; 0 < M \leq 17$ – целые, случайные целые значения из указанных диапазонов. Значения элементов A назначаются подряд (без учета переходов по измерениям) по формуле:</p> $\frac{\sqrt{2 - n^3}}{\sqrt[3]{3 - n^2 - 1}}, n \leq 1 \text{ целое}$ <p>Например, для N = 3 и M = 3 сформированный двумерный массив (пример с округлением данных для удобства представления в задании) может выглядеть примерно так:</p> <p>1,00 1,12 1,73 -2,51 -2,82 -3,37 -3,96 -4,56 -5,15</p>	<p>Метод возвращает ссылку на структуру A, N и M – параметры метода (Справочник по C#. Параметры методов - C# Microsoft Learn)</p>	<p>Имя файла – строка, заданная пользователем с клавиатуры.</p> <p>Место размещение файла – папка с решением, т.е. выше уровня обоих папок с проектами.</p> <p>При повторных запусках решения на новых входных данных создаётся новый файл для данных; имя запрашивается у пользователя повторно. Если файл уже существует, данные в нем перезаписываются.</p> <p>Данные об одном двумерном массиве пишутся по правилу: одна строка файла– одна строка матрицы, элементы разделяются ровно двумя знаками && без пробелов, в конце строки – один символ '*', числа отформатированы с точностью 3 знака после десятичного разделителя;</p> <p>Например, для примера из столбца 3:</p> <p>1,00&&1,12&&1,73* -2,51&&-2,82&&-3,37* -3,96&&-4,56&&-5,15*</p>
18	Двумерный массив NxM вещественных значений	<p>$0 < N \leq 10; 0 < M \leq 10$ – целые, вводятся с клавиатуры пользователь. Значения элементов A назначаются подряд (без учета переходов по измерениям) по формуле:</p>	<p>Метод возвращает тип void, ссылка A передана в метод с модификатором out; N и M также передаются в метод в качестве параметров</p>	<p>Имя файла – строка, заданная пользователем с клавиатуры.</p> <p>Место размещение файла – папка с исполнимым файлом консольного приложения 1.</p>

		$\frac{\cos(n)}{n+1} \sqrt{n^5}, n > 1 \text{ целое}$ <p>Например, для N = 3 и M = 3 сформированный двумерный массив (пример с округлением данных для удобства представления в задании) может выглядеть примерно так:</p> <pre>0,2702 -0,0245 -0,0159 -0,0041 0,0008 0,0016 0,0007 -0,0001 -0,0004</pre>	(Справочник по C#. Параметры методов - C# Microsoft Learn)	<p>При повторных запусках решения на новых входных данных создается новый файл для данных; имя запрашивается у пользователя повторно. Если файл уже существует, данные в нем перезаписываются. Данные об одном двумерном массиве пишутся по правилу:</p> <p>одна строка файла – одна строка матрицы, элементы разделяются ровно одним знаком '!' без пробелов, в конце строки – два символа &&, числа отформатированы с точностью 6 знаков после десятичного разделителя;</p> <p>Например, для примера из столбца 3:</p> <pre>0,270151! -0,024522! -0,015877&& -0,004085! 0,000846! 0,001556&& 0,000727! -0,000089! -0,000375&&</pre>
19	Двумерный массив NxN вещественных значений	<p>$0 < N \leq 17$ – целое, вводит с клавиатуры пользователь</p> <p>Значения элементов A назначаются подряд (без учета переходов по измерениям) по формуле:</p> $\frac{\sqrt{9n^2 - 9} - 2n}{2 - \sqrt[3]{n^3 + 5}}, n \geq 1 \text{ целое}$ <p>Например, для N = 3 двумерный массив (пример с округлением данных для удобства представления в задании) может выглядеть примерно так:</p> <pre>-10,94 -3,40 -2,12 -1,72 -1,53 -1,42 -1,35 -1,30 -1,26</pre>	<p>Метод возвращает тип void, ссылка A передана в метод с модификатором out; N также передается в метод в качестве параметра</p> <p>(Справочник по C#. Параметры методов - C# Microsoft Learn)</p>	<p>Имя файла – строка, заданная пользователем с клавиатуры.</p> <p>Место размещение файла – папка с исполнимым файлом консольного приложения 1.</p> <p>При повторных запусках решения на новых входных данных создается новый файл для данных; имя запрашивается у пользователя повторно. Если файл уже существует, данные в нем перезаписываются.</p> <p>На первой строке файла два целых числа, разделенные ровно одним пробелом (количество строк и количество столбцов матрицы). Данные об одном двумерном массиве пишутся со следующей строки по правилу:</p> <p>Одна строка файла – одна строка матрицы, элементы разделяются ровно одним знаком '*' без пробелов, в конце строки – два символа **, числа отформатированы с точностью 2 знака после десятичного разделителя;</p>

				<p>Например, для примера из столбца 3:</p> <p>3 3</p> <p>-10,94*-3,40*-2,12**</p> <p>-1,72*-1,53*-1,42**</p> <p>-1,35*-1,30*-1,26**</p>
20	Массив из N массивов вещественных значений размера N элементов каждый	<p>$0 < N \leq 21$ – целое, вводит с клавиатуры пользователь. Значения элементов A назначаются подряд (без учета переходов по измерениям) по формуле:</p> $\sqrt{n(n+1)} - \sqrt{n^2+1}, n \geq 2 \text{ целое}$ <p>Например, N = 3 сформированные массивы массивов (пример с округлением данных для удобства представления в задании) могут выглядеть примерно так:</p> <p>0,2134 0,3018 0,3490</p> <p>0,3782 0,3980 0,4122</p> <p>0,4230 0,4314 0,4382</p>	<p>Метод возвращает тип void, ссылка A передана в метод с модификатором out; N также передается в метод в качестве параметра (Справочник по C#, Параметры методов - C# Microsoft Learn)</p>	<p>Имя файла – строка, заданная пользователем с клавиатуры.</p> <p>Место размещение файла – папка с исполнимым файлом консольного приложения 1.</p> <p>При повторных запусках решения на новых входных данных создается новый файл для данных; имя запрашивается у пользователя повторно. Если файл уже существует, данные в нем перезаписываются.</p> <p>Данные об одном массиве массивов пишутся по правилу: один элемент массива, на который указывает ссылка, на строке. Строки отделяются двумя пустыми строками. Числа отформатированы с точностью до 4 знаков после десятичного разделителя</p> <p>Например, для примера из столбца 3:</p> <p>0,2134</p> <p>0,3018</p> <p>0,3490</p> <p>0,3782</p> <p>0,3980</p> <p>0,4122</p> <p>0,4230</p> <p>0,4314</p> <p>0,4382</p>

Таблица 2. Индивидуальные варианты для проекта консольного приложения 2 (программа 2)

№	Структура данных В	Правило обработки элементов структуры данных В	Правило передачи структуры данных В в метод
1	2	3	4
1	Массив вещественных одномерных массивов	<p>Элементы каждой строки циклически сдвигаются на $k \geq 0$ элементов влево. k вводит с клавиатуры пользователь, при $k = 0$ сдвиг не происходит.</p> <p>Например, $k = 2$ Данные из структуры В до обработки (пример с округлением данных для удобства представления в варианте): -1,71 -3,5 -16 9 4 3,14 После обработки: -3,5 -16 -1,71 4 3,14 9</p>	<p>Ссылка на В передаётся в метод по значению, обработка элементов происходит в исходной, доступной через ссылку структуре данных В (Справочник по C#. Параметры методов - C# Microsoft Learn).</p>
2	Двумерный массив вещественных значений	<p>Происходит обмен строк, содержащих минимальный и максимальный во всей структуре В элементы. Если минимум и максимум находятся в одной строке, то обмен не происходит.</p> <p>Например, данные из структуры В до обработки (пример с округлением данных для удобства представления в варианте): 2,11 1,95 1,82 1,71 1,63 1,57 После обработки: 1,71 1,63 1,57 2,11 1,95 1,82</p>	<p>По переданной в качестве параметра В метод формирует новую структуру данных с указанными изменениями и возвращает ссылку на нее; при этом в самой В изменений не происходит (Справочник по C#. Параметры методов - C# Microsoft Learn)</p>
3	Двумерный массив вещественных значений	<p>Все нулевые значения заменяются средним значением элементов, находящихся в данной строке и имеющих ненулевое значение.</p> <p>Например, данные из структуры В до обработки (пример с округлением данных для удобства представления в варианте): 2,00 0,89 0,61 0 0,48 0,40 0,35 0,32</p>	<p>Ссылка на В передаётся в метод по значению, обработка элементов происходит в исходной, доступной через ссылку структуре данных В (Справочник по C#. Параметры методов - C# Microsoft Learn)</p>

		<p>0,29 0,27 0 0</p> <p>После обработки:</p> <p>2,00 0,89 0,61 1,16</p> <p>0,48 0,40 0,35 0,32</p> <p>0,29 0,27 0,28 0,28</p>	
4	Массив вещественных одномерных массивов	<p>Элементы каждой строки циклически сдвигаются на $k \geq 0$ элементов вправо. k вводит с клавиатуры пользователь, при $k = 0$ сдвиг не происходит.</p> <p>Например,</p> <p>$k = 2$</p> <p>Данные из структуры В до обработки (пример с округлением данных для удобства представления в варианте):</p> <p>0,67 0,74 0,78</p> <p>0,78 0,75 0,72 0,69</p> <p>0,66 0,63</p> <p>После обработки:</p> <p>0,74 0,78 0,67</p> <p>0,72 0,69 0,78 0,75</p> <p>0,66 0,63</p>	<p>В метод по значению передается ссылка В и новая ссылка на аналогичную структуру. Результаты преобразований данных В записываются в эту новую структуру и становятся доступными вызывающему коду (Справочник по C#. Параметры методов - C# Microsoft Learn)</p>
5	Массив вещественных одномерных массивов	<p>Элементы, стоящие над главной диагональю – квадратной части матрицы, зеркально обмениваются значениями с элементами, стоящими под главной диагональю. Элементы главной диагонали из обменов исключаются. Квадратность части определяется наименьшим значением измерения.</p> <p>Например, из структуры В до обработки (пример с округлением данных для удобства представления в варианте):</p> <p>1,75 1,00 0,69</p> <p>0,52 0,42 0,35</p> <p>0,30 0,26 0,23</p> <p>После обработки:</p> <p>1,75 0,52 0,20</p> <p>1,00 0,42 0,26</p> <p>0,69 0,35 0,23</p>	<p>По переданной в качестве параметра В метод формирует новую структуру данных с указанными изменениями и возвращает ссылку на нее; при этом в самой В изменений не происходит (Справочник по C#. Параметры методов - C# Microsoft Learn)</p>

6	Массив вещественных одномерных массивов	<p>Результирующая структура данных содержит только модули дробных частей от данных исходной структуры, при чём только элементов, стоящих на нечётных позициях в исходных строках. Размер структуры данных может уменьшиться, если элементов в ней не будет, то ссылка получает значение null.</p> <p><i>Например, данные из структуры В до обработки (пример с округлением данных для удобства представления в варианте):</i></p> <p>-0,500 -0,400 -0,300 -0,235 -0,192 -0,162 -0,140 -0,123 -0,110</p> <p><i>После обработки:</i></p> <p>0,400 0,192 0,123</p>	<p>В метод по значению передается ссылка В и новая ссылка на аналогичную структуру. Результаты преобразований данных В записываются в эту новую структуру и становятся доступными вызывающему коду (Справочник по C#. Параметры методов - C# Microsoft Learn)</p>
7	Двумерный массив вещественных значений	<p>Назовём средней линией матрицы – строку, количество строк до и после которой одинаково. Требуется зеркально относительно средней линии произвести обмен значений элементов В, при этом равные по значению элементы не переприсваивать.</p> <p><i>Например, из структуры В до обработки (пример с округлением данных для удобства представления в варианте):</i></p> <p>-0,25 -0,00 0,02 0,08 0,15 0,22 0,28 0,33 0,38</p> <p><i>После обработки:</i></p> <p>0,28 0,33 0,38 0,08 0,15 0,22 -0,25 -0,00 0,02</p>	<p>В метод по значению передается ссылка В и новая ссылка на аналогичную структуру. Результаты преобразований данных В записываются в эту новую структуру и становятся доступными вызывающему коду (Справочник по C#. Параметры методов - C# Microsoft Learn)</p>
8	Массив вещественных одномерных массивов	<p>Элементы каждой строки НЕциклически сдвигаются на $-15 \leq k \leq 15$ элементов; направление сдвига определено знаком k, минус – сдвиг влево; плюс – сдвиг вправо. k вводит с клавиатуры пользователь, при $k = 0$ сдвиг не происходит.</p> <p><i>Например,</i></p>	<p>Ссылка на В передаётся в метод по значению, обработка элементов происходит в исходной, доступной через ссылку структуре данных В (Справочник по C#. Параметры методов - C# Microsoft Learn)</p>

		$k = -2$ Данные из структуры В до обработки (пример с округлением данных для удобства представления в варианте): 4,41 6,87 9,08 11,20 13,28 15,34 17,38 19,41 21,44 После обработки: 0 0 9,08 0 0 15,34 0 0 21,44	
9	Массив вещественных одномерных массивов	Элементы, стоящие над побочной диагональю, зеркально обмениваются значениями с элементами, стоящими под побочной диагональю. Элементы самой побочной диагонали из обменов исключаются. Например, из структуры В до обработки (пример с округлением данных для удобства представления в варианте): 0,9137 1,0000 1,0806 1,1561 1,2272 1,2947 1,3589 1,4203 1,4792 После обработки: 1,4792 1,2947 1,0806 1,4203 1,2272 1,0000 1,3589 1,1561 0,9137	Ссылка на В передаётся в метод по значению, обработка элементов происходит в исходной, доступной через ссылку структуре данных В (Справочник по C#. Параметры методов - C# Microsoft Learn)
10	Двумерный массив вещественных значений	Все значения элементов, стоящих под главной диагональю, заменяются средним значением элементов, стоящих на главной диагонали. Элементы самой главной диагонали после этого получают нулевое вещественное значение. Например, из структуры В до обработки (пример с округлением данных для удобства представления в варианте): 2,26 0,29 0,12 0,06 0,04 0,03 0,02 0,02 0,01	Ссылка на В передаётся в метод по значению, обработка элементов происходит в исходной, доступной через ссылку структуре данных В (Справочник по C#. Параметры методов - C# Microsoft Learn)

		<p>После обработки:</p> <pre>0 0,29 0,12 0,77 0 0,03 0,77 0,77 0</pre>	
11	Массив вещественных одномерных массивов	<p>Результирующая структура данных содержит только целые части от данных исходной структуры, при чём только с чётными значениями целых частей. Размер структуры данных может уменьшиться, если элементов в ней не будет, то ссылка получает значение null.</p> <p>Например,</p> <p>Данные из структуры В до обработки (пример с округлением данных для удобства представления в варианте):</p> <pre>3,00 5,44 8,00 10,50 12,86 15,06 17,09 18,95 20,66</pre> <p>После обработки:</p> <pre>8 10 12 18 20</pre>	<p>По переданной в качестве параметра В метод формирует новую структуру данных с указанными изменениями и возвращает ссылку на нее; при этом в самой В изменений не происходит (Справочник по C#. Параметры методов - C# Microsoft Learn)</p>
12	Массив вещественных одномерных массивов	<p>Произвести сжатие каждой строки на один элемент со сдвигом влево. Индекс элемента определяется индексом строки. Так в нулевой строке исчезнет нулевой элемент, в первой – первый и т.д. Размер каждого одномерного массива после сжатия уменьшается на единицу.</p> <p>Например, из структуры В до обработки (пример с округлением данных для удобства представления в варианте):</p> <pre>1,00 0,92 0,94 0,96 0,96 0,97 0,97 0,98 0,98 0,98 0,98 0,99</pre> <p>После обработки:</p> <pre>0,92 0,94 0,96 0,96 0,97 0,98 0,98 0,98 0,99</pre>	<p>В метод по значению передается ссылка В и новая ссылка на аналогичную структуру. Результаты преобразований данных В записываются в эту новую структуру и становятся доступными вызывающему коду (Справочник по C#. Параметры методов - C# Microsoft Learn)</p>

13	Двумерный массив вещественных значений	<p>Произвести сжатие каждой строки со сдвигом влево. Из каждой строки должны исчезнуть элементы, у которых до трансформации индексы были чётными, элемент с нулевым индексом считать чётным. Размер структуры данных уменьшается, если в ней не остаётся элементов, то ссылка на нее становится null.</p> <p><i>Например, из структуры В до обработки (пример с округлением данных для удобства представления в варианте):</i></p> <pre>0,41 1,59 3,56 6,06 8,99 12,28</pre> <p><i>После обработки:</i></p> <pre>1,59 8,99</pre>	<p>Ссылка на В передаётся в метод по ссылке, обработка элементов происходит в новой, аналогичной В структуре данных, под которую выделяется память и в случае успешной обработки ссылка на В связывается с этой новой памятью (Справочник по C#. Параметры методов - C# Microsoft Learn).</p>
14	Массив вещественных одномерных массивов	<p>Произвести сжатие каждой строки со сдвигом влево. Из каждой строки должны исчезнуть элементы, у которых до трансформации индексы были чётными, элемент с нулевым индексом считать чётным. Размер каждого одномерного массива после сжатия уменьшается на единицу.</p> <p><i>Например, из структуры В до обработки (пример с округлением данных для удобства представления в варианте):</i></p> <pre>0,27 -0,17 -0,30 -0,15 0,05 0,16 0,11 -0,02 -0,10</pre> <p><i>После обработки:</i></p> <pre>-0,17 0,05 -0,02</pre>	<p>Ссылка на В передаётся в метод по ссылке, обработка элементов происходит в новой, аналогичной В структуре данных, под которую выделяется память и в случае успешной обработки ссылка на В связывается с этой новой памятью (Справочник по C#. Параметры методов - C# Microsoft Learn)</p>
15	Двумерный массив вещественных значений	<p>Произвести сжатие каждой строки со сдвигом влево. Из каждой строки должны исчезнуть элементы, стоящие на главной диагонали. Общий размер структуры уменьшается.</p> <p><i>Например, из структуры В до обработки (пример с округлением данных для удобства представления в варианте):</i></p> <pre>0,4207 0,7274 0,1270</pre>	<p>В метод по значению передается ссылка В и новая ссылка на аналогичную структуру. Результаты преобразований данных В записываются в эту новую структуру и становятся доступными вызывающему коду (Справочник по C#. Параметры методов - C# Microsoft Learn)</p>

		<p>-0,7123 -0,9220 -0,2719 0,6438 0,9741 0,4071 <i>После обработки:</i> 0,7274 0,1270 -0,7123 -0,2719 0,6438 0,9741</p>	
16	Массив вещественных одномерных массивов	<p>Элементы каждого столбца циклически сдвигаются на $k \geq 0$ элементов вверх. k вводит с клавиатуры пользователь, при $k = 0$ сдвиг не происходит. <i>Например,</i> $k = 2$ <i>Данные из структуры В до обработки (пример с округлением данных для удобства представления в варианте):</i> -0,50 -0,78 -0,95 -1,06 -1,13 -1,18 -1,22 -1,25 -1,28 <i>После обработки:</i> -1,22 -1,25 -1,28 -0,50 -0,78 -0,95 -1,06 -1,13 -1,18</p>	<p>Ссылка на В передаётся в метод по ссылке, обработка элементов происходит в новой, аналогичной В структуре данных, под которую выделяется память и в случае успешной обработки ссылка на В связывается с этой новой памятью (Справочник по C#. Параметры методов - C# Microsoft Learn)</p>
17	Массив вещественных одномерных массивов	<p>Назовём вертикальной средней линией матрицы – столбец, количество столбцов до и после которого одинаково. Требуется зеркально относительно вертикальной средней линии произвести обмен значений элементов В, при этом равные по значению элементы не переприсваивать. <i>Например, из структуры В до обработки (пример с округлением данных для удобства представления в варианте):</i> 1,00 1,12 1,73 -2,51 -2,82 -3,37 -3,96 -4,56 -5,15 <i>После обработки:</i> 1,73 1,12 1,00</p>	<p>Ссылка на В передаётся в метод по значению, обработка элементов происходит в исходной, доступной через ссылку структуре данных В (Справочник по C#. Параметры методов - C# Microsoft Learn)</p>

		-3,37 -2,82 -2,51 -5,15 -4,56 -3,96	
18	Массив вещественных одномерных массивов	<p>Элементы каждого столбца циклически сдвигаются на $k \geq 0$ элементов вниз. k вводит с клавиатуры пользователь, при $k = 0$ сдвиг не происходит.</p> <p>Например, $k = 2$</p> <p>Данные из структуры В до обработки (пример с округлением данных для удобства представления в варианте):</p> <p>0,2702 -0,0245 -0,0159 -0,0041 0,0008 0,0016 0,0007 -0,0001 -0,0004</p> <p>После обработки:</p> <p>-0,0041 0,0008 0,0016 0,0007 -0,0001 -0,0004 0,2702 -0,0245 -0,0159</p>	<p>Ссылка на В передаётся в метод по ссылке, обработка элементов происходит в новой, аналогичной В структуре данных, под которую выделяется память и в случае успешной обработки ссылка на В связывается с этой новой памятью (Справочник по C#. Параметры методов - C# Microsoft Learn)</p>
19	Двумерный массив вещественных значений	<p>Произвести сжатие каждого столбца со сдвигом вверх. Из каждой столбца должны исчезнуть элементы, стоящие на главной диагонали. Общий размер структуры уменьшается, до «прямоугольности» элементы достраиваются нулями снизу. Если в структуре не остаётся элементов, то ссылка на нее получает значение null.</p> <p>Например, из структуры В до обработки (пример с округлением данных для удобства представления в варианте):</p> <p>-10,94 -3,40 -2,12 -1,53 -1,42 0 -1,35 -1,30 -1,26</p> <p>После обработки:</p> <p>-1,53 -3,40 -2,12 -1,35 -1,30 -1,26 0 0 0</p>	<p>В метод по значению передается ссылка В и новая ссылка на аналогичную структуру. Результаты преобразований данных В записываются в эту новую структуру и становятся доступными вызывающему коду (Справочник по C#. Параметры методов - C# Microsoft Learn)</p>
20	Двумерный массив	<p>Произвести сжатие каждой строки со сдвигом влево. Из каждой строки должны исчезнуть элементы, стоящие на главной и побочной диагоналях. Общий размер структуры уменьшается, до</p>	<p>По переданной в качестве параметра В метод формирует новую структуру данных с указанными изменениями и возвращает ссылку на нее; при этом в самой В</p>

	вещественных значений	<p>«прямоугольности» элементы достраиваются нулями справа. Если в структуре не остаётся элементов, то ссылка на нее получает значение null.</p> <p><i>Например, из структуры В до обработки (пример с округлением данных для удобства представления в варианте):</i></p> <p>0,2134 0,3018 0,3490 0,3782 0,3980 0,4122 0,4230 0,4314 0,4382</p> <p><i>После обработки:</i></p> <p>0,3018 0 0,3782 0,4122 0,4314 0</p>	<p>изменений не происходит (Справочник по C#. Параметры методов - C# Microsoft Learn)</p>
--	-----------------------	--	---