

Задача 8. Многомерные (двумерные) массивы

Входные данные для этой задачи должны читаться из текстового файла, а выходные – записываться в текстовый файл.

Данная задача должна быть оформлена двумя способами:

1) в виде консольного приложения с разбором параметров командной строки,

2) в виде оконного приложения, где двумерный массив можно задать в JTable. При этом должна быть возможность загрузить данный из файла в JTable (реализованная в виде двух функций: чтение данных из файла в двумерный массив и отображение двумерного массива в JTable), а также сохранить данные из JTable в файл (реализованная в виде двух функций: чтение данных из JTable в двумерный массив и запись двумерного массива в файл).

Функции, реализующие логику задачи и чтение / запись данных из файлов / в файлы, должны быть оформлены в виде отдельного модуля (в отдельном файле). Этот модуль без каких-либо изменений должен использоваться в двух программах: с консольным интерфесом (файлы для чтения / записи задаются в параметрах командной строки) и оконным интерфейсом.

Заранее придумать не менее 5 различных тестов, охватывающих как типичные, так и все возможные граничные (наиболее невероятные и показательные) ситуации. (Сохранить в текстовых файлах input01.txt, input02.txt и т. д.)

Решение, естественно, должно быть оформлено в виде отдельной функции. В реализации обязательно использовать вспомогательные функции (в этой задаче, какие именно вспомогательные функции использовать, не расписано, вы должны самостоятельно решить, какие именно будут удобны для вашего варианта).

В данных задачах, если явно не оговорено иное, предполагается, что все строки двумерного массива имеют одинаковую длину, т.е. массив является прямоугольным. В реализации разрешено также использовать вместо прямоугольных массивов ступенчатые массивы (т.е. массивы массивов или списки списков). В этом случае необходимо дополнительно реализовать функцию проверки «прямоугольности» ступенчатого массива, которую применять для проверки корректности данных перед обработкой.

Реализация в виде консольного приложения с разбором параметров командной строки

Имена файлов для чтения и записи должны передаваться в параметрах командной строки, например, так:

```
> java ru.vsu.cs.course1.Task8 .\input.txt .\output.txt
```

(Здесь ".." означает текущую директорию, т.е. "..\" можно опустить. Также при обращении к файлам можно использовать "..", что будет означать родительскую директорию относительно текущей, например, "..\..\input05.txt")

Еще лучше будет, если вы реализуете разбор параметров командной строки, чтобы имена входных и выходных файлов можно было задавать в виде именованных параметров, например, так:

```
> java ru.vsu.cs.course1.Task8 -i .\input.txt -o .\output.txt
```

или так (а лучше и так и так)

```
> java ru.vsu.cs.course1.Task8 --input-file=.\input.txt --output-file=.\output.txt
```

Если программа запускается без указания необходимых аргументов, либо отсутствует входной файл, то в поток ошибки (System.err.println()) должно печататься сообщение об ошибке и программа должна завершаться с кодом, отличным от 0 (код успешного

завершения).

Для разбора параметров командной строки реализовать функцию:

```
public static InputArgs parseCmdArgs(String[] args)
```

InputArgs – класс, в котором описаны поля inputFile и outputFile (и, возможно, какие-от другие параметры в зависимости от задачи).

Варианты:

1. Первый столбец с минимальной суммой элементов поменять местами с последним столбцом с максимальной суммой элементов.
2. Последний столбец, содержащий минимальный элемент, поменять местами с первым столбцом, содержащим максимальный элемент двумерного массива.
3. Создать новый двумерный массив элементов, развернув переданный массив вокруг главной диагонали (т.е. строки должны стать столбцами, а столбцы строками).
4. Создать новый массив, заменив несколько одинаковых подряд идущих строк одним вхождением данной строки. (Совпадающие строки – строки, у которых все соответствующие элементы равны друг другу).
5. Создать новый двумерный массив, исключив из переданного массива совпадающие столбцы. (Совпадающие столбцы – столбцы, у которых все соответствующие элементы равны друг другу). При формировании нового массива оставить только первый из каждого набора совпадающих столбцов.
6. Отсортировать столбцы переданного двумерного массива по возрастанию со следующим критерием сравнения столбцов. Один столбец считается большим второго столбца, если первый элемент (с индексом строки 0) первого столбца больше первого элемента второго столбца и наоборот в противном случае. Если первые элементы столбцов совпадают, то аналогичным образом для сравнения столбцов сравниваются вторые элементы столбцов и т.д.

Сортировку можно выполнить как методом пузырька, который рассматривался на лекциях, но лучше реализовать сортировку выбором (https://ru.wikipedia.org/wiki/Сортировка_выбором), для которой в среднем требуется меньше обменов.

Для сравнения столбцов обязательно реализовать дополнительную функцию, которая будет принимать в качестве аргументов двумерный массив и индексы сравниваемых столбцов, а возвращать целое число – положительное, если первый столбец больше второго по описанному критерию, отрицательное – если меньше, и 0 – если столбцы равны (т.е. одинаковые).

7. Строки, элементы которых не убывают (т.е. образуют неубывающую последовательность чисел) переместить в начало (вверх), сохранив при этом взаимное расположением перемещаемых строк.
8. Столбцы, сумма которых отрицательна, переместить в конец (вправо), сохранив при этом взаимное расположением перемещаемых столбцов.

9. Отсортировать строки переданного двумерного массива по убыванию со следующим критерием сравнения строк. Одна строка считается больше другой, если сумма ее элементов больше суммы второй строки, меньше – если сумма меньше, и равны – если суммы равны. При этом должна обеспечиваться устойчивость сортировки (т.е. «равные» строки должны сохранять взаимное расположение относительно друг друга).

Сортировку можно выполнить как методом пузырька, который рассматривался на лекциях, но лучше реализовать сортировку выбором (https://ru.wikipedia.org/wiki/Сортировка_выбором), для которой в среднем требуется меньше обменов.

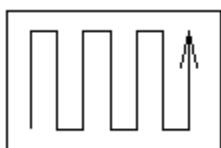
10. Осуществить циклический сдвиг диагоналей двумерного массива на n позиций. Пример для $n = 1$:

$$\begin{array}{ccc} \{\{ 1, 2, 3, 4, 5 \}, & & \{ \{ 12, 13, 14, 15, 11 \}, \\ \{ 6, 7, 8, 9, 10 \}, & \rightarrow & \{ 2, 3, 4, 5, 1 \}, \\ \{ 11, 12, 13, 14, 15 \} \}, 1) & & \{ 7, 8, 9, 10, 6 \} \} \end{array}$$

11. Написать функцию, которая создаст новый двумерный массив, «поворнутый» относительно переданного на 90° .

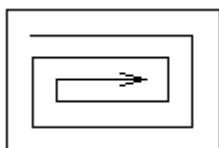
12. Осуществить циклический сдвиг столбцов или строк (указывается отдельно) двумерного массива на n позиций.

13. (*) Для переданного двумерного массива определить, образуют ли его элементы упорядоченную последовательность при просмотре элементов в следующем порядке:



Дополнительные массивы и списки не создавать.

14. (*) Для переданного двумерного массива определить, образуют ли его элементы упорядоченную последовательность при просмотре элементов в следующем порядке:

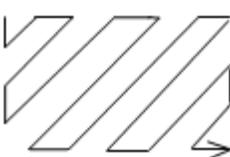


Дополнительные массивы и списки не создавать.

15. Создать новый двумерный массив, исключив из переданного двумерного массива строки и столбцы, состоящие из одинаковых элементов (т.е. строка, состоящая, предположим, из одних единиц, не должна попасть в новый двумерный массив).

16. Для переданного двумерного массива чисел создать новый двумерный массив чисел, где в соответствующей ячейке будет записано количество элементов, включая данный, на диагоналях для соответствующего элемента переданного массива, совпадающих со значением в данной ячейке переданного массива, например:

$$\begin{array}{ccc} \{ \{ 1, 2, 3, 4 \}, & & \{ \{ 2, 2, 1, 1 \}, \\ \{ 3, 2, 2, 0 \}, & \rightarrow & \{ 1, 1, 3, 1 \}, \\ \{ 1, 2, 1, 0 \} \} & & \{ 1, 2, 2, 1 \} \} \end{array}$$

17. Будем называть соседями элемента с индексами (r, c) в двумерном массиве такие элементы, индексы которых отличаются от (r, c) не более чем на единицу. Для переданного двумерного массива составить новый двумерный массив, где каждый элемент (r, c) нового массива будет содержать число соседей элемента (r, c) в переданном массиве, равных элементу (r, c) (в переданном массиве).
18. Элемент двумерного массива назовем седловой точкой, если он является наименьшим в своей строке и одновременно наибольшим в своем столбце или, наоборот, является наибольшим в своей строке и наименьшим в своем столбце. Для переданного двумерного массива составить новый двумерный массив такого же размера из логических значений, где истина будет означать, что соответствующий элемент является седловой точкой в переданном массиве.
19. (*) Создать новый двумерный массив, состоящий из ненулевых столбцов и строк переданного двумерного массива. (Нулевая строка или столбец – строка или столбец, содержащая одни нулевые элементы.)
20. Создать новый квадратный двумерный массив из переданного двумерного массива, обрезав от последнего лишние строки или столбцы, т.е. такие, для которых нет соответствующих столбцов или строк.
21. Для переданного двумерного массива найти все квадраты размером 2×2 , заполненные одинаковыми цифрами. Два квадрата отличаются, если отличаются координаты верхнего левого угла квадрата. Ответ получить в виде массива / списка координат (номер строки, номер столбца) верхних левых углов таких квадратов, например:
- $$\begin{array}{l} \{ \{ 1, 3, 3, 3, 5, 2 \}, \\ \{ 3, 3, 3, 3, 5, 3 \}, \rightarrow \{ (R: 0, C: 1), (R: 0, C: 2), (R: 1, C: 1), (R: 2, C: 3) \} \\ \{ 1, 3, 3, 4, 4, 3 \}, \\ \{ 4, 5, 4, 4, 4, 1 \} \} \end{array}$$
22. Для двух переданных прямоугольных массивов целых чисел определить, можно ли второй массив наложить на первый таким образом, чтобы значения в ячейках совпадали. Функция должна вернуть координаты ячейки первого массива, на который требуется поместить верхний левый угол второго массива.
23. (*) Для переданного двумерного массива определить, образуют ли его элементы упорядоченную последовательность при просмотре элементов в следующем порядке:
- 
- Дополнительные массивы и списки не создавать.
24. Отсортировать столбцы двумерного массива согласно возрастанию значений в указанной строке.
Сортировку можно выполнить как методом пузырька, который рассматривался на лекциях, но лучше реализовать сортировку выбором (https://ru.wikipedia.org/wiki/Сортировка_выбором), для которой в среднем требуется меньше обменов.

25. Для переданного двумерного массива чисел создать новый двумерный массив чисел, где в соответствующей ячейке будет записано количество элементов в строке и столбце переданного массива, включая рассматриваемый, совпадающих со значением в данной ячейке переданного массива, например:

$$\begin{array}{ll} \{ \{ 1, 2, 3, 4 \}, & \{ \{ 2, 3, 1, 1 \}, \\ \{ 3, 2, 2, 0 \}, & \rightarrow \quad \{ 1, 4, 2, 2 \}, \\ \{ 1, 2, 1, 0 \} \} & \{ 3, 3, 2, 2 \} \} \end{array}$$

26. Создать новый двумерный массив чисел, исключив из переданного двумерного массива все строки и столбцы, содержащие максимальные и минимальные элементы среди всех элементов переданного массива.

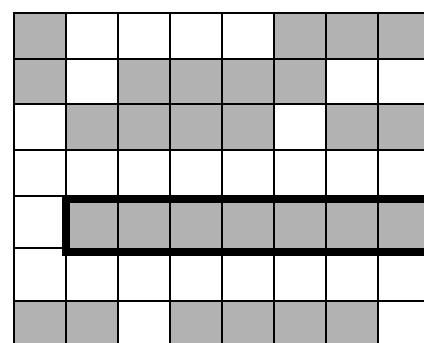
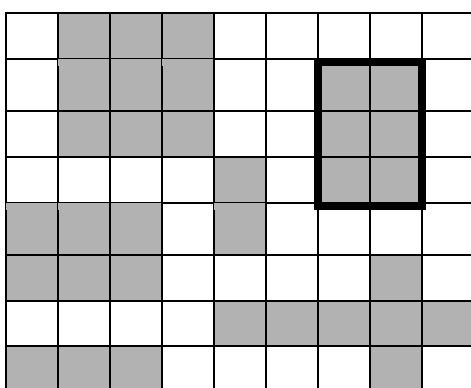
27. Выбрать из двумерного массива квадрат элементов с максимальной суммой элементов, который вернуть в виде двумерного массива.

Учесть, что если в переданном двумерном массиве есть отрицательные элементы, то квадрат с максимальной суммой не обязательно будет максимального размера. Обязательно придумать примеры под такой случай.

28. (*) Для переданного двумерного массива логических элементов, найти наибольший по площади прямоугольник, который полностью состоит из истинных значений, окруженный ложными значениями или границами массива.

Если существует несколько таких прямоугольников максимальной площади, вернуть самый верхний (по верхнему левому углу), затем самый левый.

На картинках ниже обведены такие прямоугольники для входного массива (закрашенные клетки – истина, белые – ложь):



Функция должна вернуть координату верхнего левого угла прямоугольника и высоту + ширину (для приведенных картинок это будет (1, 6, 3, 2) и (4, 1, 1, 7)). Если прямоугольник по условиям задачи найти не удалось, то функция должна вернуть (-1, -1, -1, -1).

29. (*) Вам передан прямоугольный массив целых чисел, описывающий поле для игры «5 в ряд» (<http://riddle-middle.ru/igry/krestiki-noliki>). Если в клетке поля 0, то в соответствующей ячейке массива будет 0, если 1 – 1, в противном случае считаем, что клетка поля пустая.

Вам необходимо определить, кто при текущем расположении 0 и 1 на поле выиграл. Функция должна вернуть 1, если выиграли 1-цы (есть 5 в ряд единиц, но нет 5 в ряд нулей), -1 – если 1-цы проиграли (есть 5 в ряд нулей, но нет 5 в ряд единиц), 0 – если нет победителя (если нет 5 в ряд и нулей и единиц или же есть и те и другие).

30. (*) Будем называть «дружественными» элементами для двумерного массива такие

элементы, которые соединяются с данным элементом одной из сторон и имеют одно и то же значение, а также все дружественные элементы для дружественных элементов данного элемента. Составить для переданного двумерного массива новый массив, в котором каждый элемент будет содержать информацию о количестве дружественных элементов для соответствующего элемента в переданном массиве. Пример:

$$\begin{array}{l} \{ \{ 1, 2, 3, 9, 9 \}, \\ \{ 2, 2, 2, 9, 2 \}, \\ \{ 1, 1, 2, 2, 4 \} \} \end{array} \rightarrow \begin{array}{l} \{ \{ 0, 5, 0, 2, 2 \}, \\ \{ 5, 5, 5, 2, 0 \}, \\ \{ 2, 2, 5, 5, 0 \} \} \end{array}$$

31. (*) Реализовать функцию, которая будет возвращать двумерный массив целых чисел в виде календаря на заданный месяц. Функция должна принимать два числа – номер и номер года, например:

$$(11, 2017) \rightarrow \begin{array}{l} \{ \{ 30, 31, 1, 2, 3, 4, 5 \}, \\ \{ 6, 7, 8, 9, 10, 11, 12 \}, \\ \{ 13, 14, 15, 16, 17, 18, 19 \}, \\ \{ 20, 21, 22, 23, 24, 25, 26 \}, \\ \{ 27, 28, 29, 30, 1, 2, 3 \} \} \end{array}$$

- 32.