# Тема 7: Указатели. Двумерные массивы.

*Цель: формирование навыков работы с указателями, двумерными массивами*.

## Задания для практических занятий.

### Задание 1.

В каждой функции обращение к элементам массива записать двумя способами: традиционно, используя индексы, и через указатели.

1. Напишите функцию inArray, имеющую три параметра – адрес матрицы (целые числа, **встроенный массив**), количество строк и количество столбцов, выполняющую ввод всех элементов массива из потока cin. Функция должна инициировать исключение при обнаружении ошибки ввода.
2. Напишите функцию inArray, имеющую три параметра – адрес матрицы (целые числа, **массив в динамической памяти**), количество строк и количество столбцов, выполняющую ввод всех элементов массива из потока cin. Функция должна инициировать исключение при обнаружении ошибки ввода.
3. Напишите функцию outArray, имеющую три параметра – адрес матрицы (целые числа, **встроенный массив**), количество строк и количество столбцов, выполняющую вывод всех элементов массива в поток cout. Каждую строку матрицы надо выводить с новой строки.
4. Напишите функцию outArray, имеющую три параметра – адрес матрицы (целые числа, **массив в динамической памяти**), количество строк и количество столбцов, выполняющую вывод всех элементов массива в поток cout. Каждую строку матрицы надо выводить с новой строки.
5. Напишите функцию numberOfRowsOrdered, имеющую три параметра – адрес матрицы (целые числа, **массив в динамической памяти**), количество строк и количество столбцов, возвращающую количество строк, элементы которых упорядочены по возрастанию. Для проверки, является ли строка упорядоченной, использовать функцию isOrderedArray.

Функция isOrderedArray имеет два параметра – адрес одномерного массива (целые числа) и количество элементов, возвращает значение true, если элементы массива упорядочены по возрастанию, и false в противном случае.

1. Напишите функцию main для тестирования функций, в которой

* опишите встроенный массив (размер задается поименованными константами) и массив в динамической памяти;
* введите количество строк и столбцов, выделите память для массива, предусмотрите обработку исключений, если количество задано некорректно и если память не выделена;
* вызовите функции и выведите ответы.

## Задания для лабораторных работ

### Индивидуальное задание. Встроенные массивы. Проект в Visual Studio

Напишите функцию (в соответствии с указанным вариантом) для работы со **встроенным массивом**. Функция, должна иметь три параметра – адрес матрицы (целые числа), количество строк и количество столбцов и возвращать найденное количество.

Напишите программу, выполняющую ввод и вывод матрицы и выполняющую действия в соответствии с указанным вариантом.

**Входные данные:** Значения элементов ввести из файла потока cin.

**Выходные данные**: в соответствии заданным с вариантом вывести в поток cout.

### Варианты***:***

1. Количество строк, не содержащих ни одного нулевого элемента.
2. Количество строк, содержащих хотя бы один нулевой элемент.
3. Количество строк, содержащих только положительные элементы.
4. Количество столбцов, не содержащих ни одного нулевого элемента.
5. Количество столбцов, содержащих хотя бы один нулевой элемент.
6. Количество столбцов, содержащих только положительные элементы.
7. Количество строк, содержащих только различные элементы.
8. Количество столбцов, содержащих только различные элементы.
9. Количество строк, содержащих только различные элементы.
10. Количество столбцов, содержащих повторяющиеся элементы.
11. Количество строк, содержащих повторяющиеся элементы.
12. Количество строк, элементы которых упорядочены по возрастанию элементов.
13. Количество столбцов, элементы которых упорядочены по возрастанию элементов.
14. Количество строк, элементы которых упорядочены по возрастанию элементов.
15. Количество столбцов, содержащих более трех повторяющихся элементов.
16. Количество строк, содержащих более трех повторяющихся элементов.
17. Количество столбцов, имеющих одинаковую сумму всех элементов.
18. Количество строк, имеющих одинаковую сумму всех элементов.

### Индивидуальное задание. Массивы в динамической памяти. Проект в Visual Studio

Напишите функцию (в соответствии с указанным вариантом) для работы с **двумерным массивом** – квадратной матрицей в динамической памяти. Функция, должна иметь два параметра – адрес матрицы (целые числа), размерность матрицы.

Напишите программу, выполняющую ввод и вывод матрицы и выполняющую действия в соответствии с указанным вариантом.

**Входные данные:** размерность матрицы, значения элементов (если требуется) ввести из файла или потока cin.

**Выходные данные**: в соответствии заданным с вариантом вывести в поток cout.

### Варианты:

1. Количество седловых точек матрицы. Матрица A имеет седловую точку A[i][j], если A[i][j] является минимальным элементом в i-ой строке и максимальным в j-ом столбце.
2. Число локальных минимумов. Элемент матрицы называется локальным минимумом, если он строго меньше всех имеющихся у него соседей.
3. Матрицу N\*N заполнить натуральными числами от 1 до N\*N по спирали, начинающейся в нижнем левом углу и закрученной по часовой стрелке.
4. Матрицу N\*N заполнить натуральными числами от 1 до N\*N по спирали, начинающейся в верхнем левом углу и закрученной по против часовой стрелки.
5. Матрицу N\*N заполнить натуральными числами от 1 до N\*N по спирали, начинающейся в верхнем левом углу и закрученной по часовой стрелке.
6. Матрицу N\*N заполнить натуральными числами от 1 до N\*N по спирали, начинающейся в верхнем левом углу и закрученной по против часовой стрелки.
7. Матрицу N\*N заполнить следующим способом. Элементам, находящимся на периферии (по периметру матрицы), присвоить значение 1, по периметру оставшейся подматрицы – значения 2 и так далее до заполнения матрицы.
8. Количество столбцов, в которых нет подряд идущих одинаковых элементов.
9. Количество строк, в которых нет подряд идущих повторяющихся элементов.
10. Номер столбца, в котором находится самая длинная серия подряд идущих равных элементов.
11. Номер строки, в которой находится самая длинная серия подряд идущих равных элементов.
12. Максимум среди сумм элементов диагоналей, параллельных побочной диагонали матрицы.
13. Минимум среди сумм элементов диагоналей параллельных главной диагонали матрицы.
14. Максимум среди сумм элементов диагоналей, параллельных главной диагонали матрицы.
15. Минимум среди сумм элементов диагоналей, параллельных побочной диагонали матрицы.
16. Для заданных k и m элементу a[k][m] присвоить значение 1, элементам окаймляющим его (соседним с ним по вертикали, горизонтали и диагоналям) – значения 2, элементам следующего окаймления - значение 3, и так далее до заполнения матрицы.
17. Сумму модулей элементов, расположенных ниже главной диагонали сглаженной матрицы. Соседями элемента A[i][j] в матрице назовем элементы A[k][l] c i-1 <= k <= i+1, j-1 <= l <= j+1, {k,l} != {i,j}. Операция сглаживания матрицы дает новую матрицу того же размера, каждый элемент которой получается, как среднее арифметическое имеющихся соседей соответствующего элемента исходной матрицы.
18. Сумму модулей элементов, расположенных выше главной диагонали сглаженной матрицы. Соседями элемента A[i][j] в матрице назовем элементы A[k][l] c i-1 <= k <= i+1, j-1 <= l <= j+1, {k,l} != {i,j}. Операция сглаживания матрицы дает новую матрицу того же размера, каждый элемент которой получается, как среднее арифметическое имеющихся соседей соответствующего элемента исходной матрицы.
19. Количество диагоналей, параллельных побочной диагонали, не содержащих ни одного нулевого элемента.
20. Проверить, является ли матрица нижней треугольной.