

## ЛАБОРАТОРНАЯ РАБОТА №3

### Курс «Основы программирования»

**Тема:** Массивы. Пользовательские функции.

**Цель:** Научиться работать с массивами в С#; научиться создавать и обращаться к собственным функциям.

**Темы для предварительной проработки** <sup>[устно]</sup>:

- Одномерные массивы в языке С#.
- Оператор foreach.
- Двумерные массивы.
- Функции. Параметры функций.
- Рекурсия.

**Общие задания** <sup>[код]</sup> :

1. Написать программу, которая запрашивает число элементов массива, после чего создает массив, заполняет его случайными целыми числами в диапазоне от -30 до 45 и выводит на экран строками по 10 элементов. Программа должна после этого вывести элементы массива в обратном направлении, начиная с последнего, игнорируя отрицательные элементы.
2. Написать программу поворота двумерного массива размерности 7x7 на 90 градусов вправо (без использования дополнительных массивов).
3. Написать программу циклического сдвига массива на  $k$  позиций влево.
4. Написать функции для поэлементного сложения и вычитания двумерных массивов 3x3. Функции должны принимать массивы в качестве параметров и выдавать результирующий массив в качестве возвращаемого значения. В *третьем параметре* функции необходимо вернуть среднее значение всех элементов входных массивов.
5. Написать программу перемножения двух матриц 5x5.
6. Написать и продемонстрировать работу следующих функций:
  - sumIterative – итерационно вычисляет сумму элементов массива;
  - sumRecursive – рекурсивно вычисляет сумму элементов массива;
  - minIterative – итерационно вычисляет минимальный элемент в массиве;
  - minRecursive – рекурсивно вычисляет минимальный элемент в массиве.
7. Написать рекурсивную функцию для нахождения  $n$ -ого члена ряда Фибоначчи по формулам, приведенным в лабораторной работе №2.
8. Написать программу, позволяющую рекурсивно вычислить определитель матрицы NxN по формуле:

$$\det A = \sum_{k=1}^N a_{1k} A_{1k} , \quad A_{ij} = (-1)^{i+j} \cdot M_{ij} ,$$

где  $M_{ij}$  – это дополнительный минор (определитель матрицы, получаемой из исходной вычеркиванием  $i$ -й строки и  $j$ -го столбца).

### **Индивидуальные задания** <sup>[КОД]</sup> :

1. Написать программу, заполняющую и отображающую на экране двумерный массив 9x9, в соответствии с вариантом (*приложение А*).
2. Написать программу, работающую с одномерным массивом, в соответствии с вариантом.

### **Контрольные вопросы** <sup>[ОТЧЕТ]</sup> :

1. Какими способами можно на языке С# объявить и проинициализировать одномерный массив?
2. Можно ли в С# узнать размер (количество элементов) массива?
3. Как работает оператор foreach?
4. Какими способами можно на языке С# объявить и проинициализировать двумерный массив? Какие есть типы двумерных массивов?
5. Синтаксис пользовательской функции. Параметры по умолчанию.
6. Передача параметров в функцию. Ключевые слова params, ref и out.
7. Что такое рекурсия? Особенности написания рекурсивных функций.

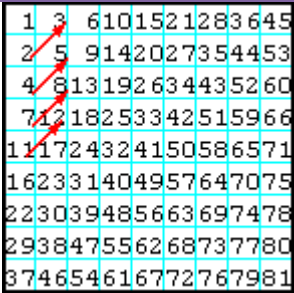
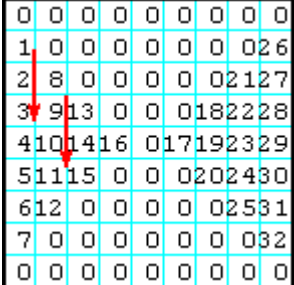
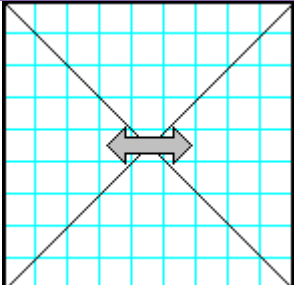
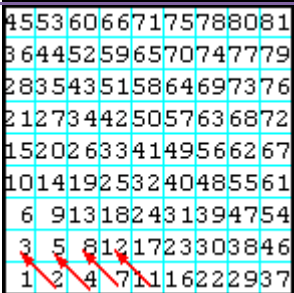
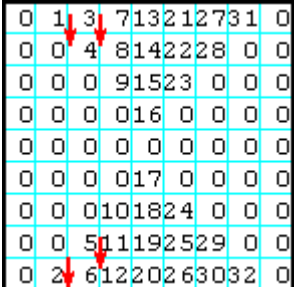
### **Рекомендуемые источники:**

- [1] Шилдт Г. С# 4.0. Полное руководство. – М.: Издательский дом «Вильямс», 2011. – 1056с.
- [2] Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: Построение и анализ. – М.: МЦНМО, 1999. – 960с.
- [3] Ахо А., Хопкрофт Д., Ульман Д. Структуры данных и алгоритмы. – М.: Издательский дом «Вильямс», 2001. – 384с.
- [4] Кнут Д. Искусство программирования, том 1. Основные алгоритмы. – М.: Издательский дом «Вильямс», 2000. – 720с.
- [5] Златопольский Д.М. Сборник задач по программированию. – СПб.: БХВ-Петербург, 2007. – 240с.

## Приложение А. Варианты индивидуальных заданий.

### Задание 1\*.

№ варианта	Содержание задания	Иллюстрация
1	Заполнить матрицу случайными числами. Отобразить главную и побочную диагонали симметрично относительно вертикальной оси.	
2	Заполнить матрицу случайными числами. Отобразить матрицу симметрично относительно главной диагонали	
3	Заполнить матрицу линейной последовательностью чисел, от левого верхнего угла по спирали: вправо - вниз - влево - вверх.	
4	Заполнить матрицу линейной последовательностью чисел, от центра по спирали: влево - вниз - вправо - вверх.	
5	Заполнить матрицу случайными числами. На главной диагонали разместить суммы элементов, которые лежат на той же строке и том же столбце.	

6	Заполнить матрицу линейной последовательностью чисел, от левого верхнего угла по диагонали: вправо - вверх.	
7	Заполнить секторы матрицы, которые лежат влево и вправо от главной и побочной диагоналей, линейной последовательностью чисел, от левого верхнего угла вниз - вправо. Остаток матрицы заполнить нулями.	
8	Заполнить матрицу случайными числами. Отобразить симметрично относительно вертикальной оси секторы матрицы, которые лежат влево и вправо от главной и побочной диагоналей.	
9	Заполнить матрицу линейной последовательностью чисел, от левого нижнего угла по диагонали: влево - вверх.	
10	Заполнить секторы матрицы, которые лежат выше и ниже главной и побочной диагоналей, линейной последовательностью чисел от левого верхнего угла вниз - вправо. Остаток матрицы заполнить нулями.	

## *Задание 2.*

*Примечание. Для всех вариантов программа сначала должна запрашивать у пользователя  $N$  – некоторое четное (!) число элементов массива, а затем и сами элементы массива.*

### Вариант 1.

Написать программу, которая выводит на экран TRUE, если элементы массива представляют собой возрастающую последовательность, иначе – FALSE.

### Вариант 2.

Написать программу, которая выводит на экран результат вычитания суммы элементов левой половины массива из суммы элементов правой половины массива.

### Вариант 3.

Написать программу, выводящую на экран количество пар элементов массива, которые по модулю отличаются друг от друга не более, чем на 5.

### Вариант 4.

Написать программу, выводящую на экран количество случаев в массиве, когда знак элемента отличается от предыдущего.

### Вариант 5.

Написать программу, которая выводит на экран максимальный элемент среди нечетных элементов массива, находящихся на четных позициях.

### Вариант 6.

Написать программу, которая проверяет, представляют ли элементы массива арифметическую прогрессию. Если да, то вывести на экран шаг прогрессии, иначе – вывести FALSE.

### Вариант 7.

Написать программу, которая выводит на экран TRUE, если все суммы симметричных элементов массива равны, иначе – FALSE.

### Вариант 8.

Написать программу, которая выводит на экран «1», если в первой половине массива отрицательных чисел больше, чем во второй, «2», если в первой половине массива отрицательных чисел меньше, и «0», если в первой половине массива столько же отрицательных чисел, сколько и во второй.

### Вариант 9.

Написать программу, которая выводит на экран TRUE, если положительные и отрицательные элементы в массиве чередуются строго друг за другом, по одному, иначе – FALSE.

### Вариант 10.

Написать программу, которая выводит на экран количество элементов массива, превышающих среднее арифметическое всех элементов.