

Лабораторная работа №6

Тема: Разработка программ с использованием массивов

Цель работы: Формирование умений и навыков в разработке сложных циклических алгоритмов обработки массивов с использованием процедур и функций пользователя.

Время на выполнение работы: 2 часа

Этапы работы:

- I.** Ознакомиться с теоретическими сведениями.
- II.** Выполнить задания, предложенные преподавателем.
- III.** Ответить на контрольные вопросы.

I. Краткие теоретические сведения

Тип Array предоставляет следующие члены.

Свойства

Имя	Описание
IsFixedSize	Получает значение, показывающее, имеет ли список Array фиксированный размер.
IsReadOnly	Получает значение, указывающее, доступен ли объект Array только для чтения.
IsSynchronized	Получает значение, показывающее, является ли доступ к коллекции Array синхронизированным (потокобезопасным).
Length	Получает 32-разрядное целое число, представляющее общее число элементов во всех измерениях массива Array.
LongLength	Получает 64-разрядное целое число, представляющее общее число элементов во всех измерениях массива Array.

Rank	Получает ранг (размерность) объекта Array. Например, одномерный массив возвращает 1, двухмерный массив возвращает 2 и т д
SyncRoot	Получает объект, который может использоваться для синхронизации доступа к объекту Array.

Методы

Имя	Описание
AsReadOnly <T>	Возвращает для заданного массива доступную только для чтения программу-оболочку.
BinarySearch h(Array, Object)	Выполняет поиск заданного элемента во всем отсортированном одномерном массиве, используя интерфейс IComparable , реализуемый каждым элементом массива и заданным объектом.
BinarySearch h(Array, Object, IComparer)	Выполняет поиск значения во всем отсортированном одномерном массиве, используя заданный интерфейс IComparer .
BinarySearch h(Array, Int32, Int32, Object)	Выполняет поиск значения в диапазоне элементов отсортированного одномерного массива, используя интерфейс IComparable , реализуемый каждым элементом массива и заданным значением.
BinarySearch h(Array, Int32, Int32, Object, IComparer)	Выполняет поиск значения в диапазоне элементов отсортированного одномерного массива, используя заданный интерфейс IComparer .
BinarySearch h<T>(T[], T)	Выполняет поиск заданного элемента во всем отсортированном одномерном массиве, используя универсальный интерфейс IComparable<T> , реализуемый каждым элементом массива Array и заданным объектом.
BinarySearch h<T>(T[], T,	Выполняет поиск значения во всем отсортированном одномерном

<code>IComparer<T></code>	массиве, используя заданный универсальный интерфейс <code>IComparer<T></code> .
<code>BinarySearch<T>(T[], Int32, Int32, T)</code>	Выполняет поиск значения в диапазоне элементов отсортированного одномерного массива, используя универсальный интерфейс <code>IComparable<T></code> , реализуемый каждым элементом массива <code>Array</code> и заданным значением.
<code>BinarySearch<T>(T[], Int32, Int32, T, IComparer<T>)</code>	Выполняет поиск значения в диапазоне элементов отсортированного одномерного массива, используя заданный универсальный интерфейс <code>IComparer<T></code> .
<code>Clear</code>	Задаёт диапазон элементов в массиве значение по умолчанию каждого типа элемента.
<code>Clone</code>	Создаёт неполную копию массива <code>Array</code> .
<code>Constrained Copy</code>	Копирует диапазон элементов из массива <code>Array</code> , начиная с заданного индекса источника, и вставляет его в другой массив <code>Array</code> , начиная с заданного индекса назначения. Гарантирует, что в случае невозможности успешно скопировать весь диапазон, все изменения будут отменены.
<code>ConvertAll<TInput, TOutput></code>	Преобразует массив одного типа в массив другого типа.
<code>Copy(Array, Array, Int32)</code>	Копирует диапазон элементов из массива <code>Array</code> , начиная с первого элемента, и вставляет его в другой массив <code>Array</code> , также начиная с первого элемента. Длина задаётся как 32-разрядное целое число.
<code>Copy(Array, Array, Int64)</code>	Копирует диапазон элементов из массива <code>Array</code> , начиная с первого элемента, и вставляет его в другой массив <code>Array</code> , также начиная с первого элемента. Длина задаётся как 64-разрядное целое число.
<code>Copy(Array, Int32, Array)</code>	Копирует диапазон элементов из массива <code>Array</code> , начиная с заданного индекса источника, и вставляет его в другой массив <code>Array</code> , начиная с заданного индекса назначения. Длина и индексы задаются как 32-

Int32, Int32)	разрядные целые числа.
Copy(Array, Int64, Array, Int64, Int64)	Копирует диапазон элементов из массива Array, начиная с заданного индекса источника, и вставляет его в другой массив Array, начиная с заданного индекса назначения. Длина и индексы задаются как 64-разрядные целые числа.
CopyTo(Array, Int32)	Копирует все элементы текущего одномерного массива в заданный одномерный массив начиная с указанного индекса в массиве назначения. Индекс задается как 32-разрядное целое число.
CopyTo(Array, Int64)	Копирует все элементы текущего одномерного массива в заданный одномерный массив начиная с указанного индекса в массиве назначения. Индекс задается как 64-разрядное целое число.
CreateInstance(Type, Int32)	Создает одномерный массив Array заданного типа Type и длины, индексация которого начинается с нуля.
CreateInstance(Type, Int32[])	Создает многомерный массив Array заданного типа Type с заданными длинами по измерениям и индексацией, начинающейся с нуля. Длины по измерениям задаются в массиве 32-разрядных целых чисел.
CreateInstance(Type, Int64[])	Создает многомерный массив Array заданного типа Type с заданными длинами по измерениям и индексацией, начинающейся с нуля. Длины по измерениям задаются в массиве 64-разрядных целых чисел.
CreateInstance(Type, Int32, Int32)	Создает двумерный массив Array заданного типа Type с заданными длинами по измерениям и индексацией, начинающейся с нуля.
CreateInstance(Type, Int32[], Int32[])	Создает многомерный массив Array заданного типа Type с заданными длинами по измерениям и нижними границами.
CreateInstance(Type, Int32, Int32, Int32)	Создает трехмерный массив Array заданного типа Type с заданными длинами по измерениям и индексацией, начинающейся с нуля.

<code>Equals(Object)</code>	Определяет, равен ли заданный объект текущему объекту. (Унаследовано от <code>Object</code> .)
<code>Exists<T></code>	Определяет, содержит ли заданный массив элементы, удовлетворяющие условиям указанного предиката.
<code>Finalize</code>	Позволяет объекту попытаться освободить ресурсы и выполнить другие операции очистки, перед тем как объект будет утилизирован в процессе сборки мусора. (Унаследовано от <code>Object</code> .)
<code>Find<T></code>	Выполняет поиск элемента, удовлетворяющего условиям указанного предиката, и возвращает первое найденное вхождение в пределах всего списка <code>Array</code> .
<code>FindAll<T></code>	Извлекает все элементы, удовлетворяющие условиям указанного предиката.
<code>FindIndex<T>(T[], Predicate<T>)</code>	Выполняет поиск элемента, удовлетворяющего условиям указанного предиката, и возвращает отсчитываемый от нуля индекс первого найденного вхождения в пределах всего списка <code>Array</code> .
<code>FindIndex<T>(T[], Int32, Predicate<T>)</code>	Выполняет поиск элемента, удовлетворяющего условиям указанного предиката, и возвращает отсчитываемый от нуля индекс первого вхождения в диапазоне элементов списка <code>Array</code> , начиная с заданного индекса и заканчивая последним элементом.
<code>FindIndex<T>(T[], Int32, Int32, Predicate<T>)</code>	Выполняет поиск элемента, удовлетворяющего условиям указанного предиката, и возвращает отсчитываемый от нуля индекс первого вхождения в диапазоне элементов списка <code>Array</code> , начинающемся с заданного индекса и содержащем указанное число элементов.
<code>FindLast<T></code>	Выполняет поиск элемента, удовлетворяющего условиям указанного предиката, и возвращает последнее найденное вхождение в пределах всего списка <code>Array</code> .
<code>FindLastIndex<T>(T[],</code>	Выполняет поиск элемента, удовлетворяющего условиям указанного предиката, и возвращает отсчитываемый от нуля индекс последнего

<code>Predicate<T></code>	найденного вхождения в пределах всего списка <code>Array</code> .
<code>FindLastIndex<T>(T[], Int32, Predicate<T>)</code>	Выполняет поиск элемента, удовлетворяющего условиям указанного предиката, и возвращает отсчитываемый от нуля индекс последнего вхождения в диапазоне элементов списка <code>Array</code> , начиная с первого элемента и заканчивая элементом с заданным индексом.
<code>FindLastIndex<T>(T[], Int32, Int32, Predicate<T>)</code>	Выполняет поиск элемента, удовлетворяющего условиям указанного предиката, и возвращает отсчитываемый от нуля индекс последнего вхождения в диапазоне элементов списка <code>Array</code> , содержащем указанное число элементов и заканчивающемся элементом с заданным индексом.
<code>ForEach<T></code>	Выполняет указанное действие с каждым элементом указанного массива.
<code>GetEnumerator</code>	Возвращает объект <code>IEnumerator</code> для массива <code>Array</code> .
<code>GetHashCode</code>	Играет роль хэш-функции для определённого типа. (Унаследовано от <code>Object</code> .)
<code>GetLength</code>	Получает 32-разрядное целое число, представляющее количество элементов в заданном измерении массива <code>Array</code> .
<code>GetLongLength</code>	Получает 64-разрядное целое число, представляющее количество элементов в заданном измерении массива <code>Array</code> .
<code>GetLowerBound</code>	Получает Индекс первого элемента заданного измерения в массиве.
<code>GetType</code>	Возвращает объект <code>Type</code> для текущего экземпляра. (Унаследовано от <code>Object</code> .)
<code>GetUpperBound</code>	Получает Индекс последнего элемента заданного измерения в массиве.

<code>GetValue(Int 32)</code>	Получает значение, хранящееся в указанной позиции одномерного массива <code>Array</code> . Индекс задается как 32-разрядное целое число.
<code>GetValue(Int 32[])</code>	Получает значение, хранящееся в указанной позиции многомерного массива <code>Array</code> . Индексы задаются как массив 32-разрядных целых чисел.
<code>GetValue(Int 64)</code>	Получает значение, хранящееся в указанной позиции одномерного массива <code>Array</code> . Индекс задается как 64-разрядное целое число.
<code>GetValue(Int 64[])</code>	Получает значение, хранящееся в указанной позиции многомерного массива <code>Array</code> . Индексы задаются как массив 64-разрядных целых чисел.
<code>GetValue(Int 32, Int32)</code>	Получает значение в заданной позиции двухмерного массива <code>Array</code> . Индексы задаются в виде 32-разрядных целых чисел.
<code>GetValue(Int 64, Int64)</code>	Получает значение в заданной позиции двухмерного массива <code>Array</code> . Индексы задаются в виде 64-разрядных целых чисел.
<code>GetValue(Int 32, Int32, Int32)</code>	Получает значение, хранящееся в указанной позиции трехмерного массива <code>Array</code> . Индексы задаются в виде 32-разрядных целых чисел.
<code>GetValue(Int 64, Int64, Int64)</code>	Получает значение, хранящееся в указанной позиции трехмерного массива <code>Array</code> . Индексы задаются в виде 64-разрядных целых чисел.
<code>IndexOf(Array, Object)</code>	Выполняет поиск заданного объекта внутри всего одномерного массива <code>Array</code> и возвращает индекс его первого вхождения.
<code>IndexOf(Array, Object, Int32)</code>	Выполняет поиск указанного объекта и возвращает индекс первого вхождения в диапазоне элементов одномерного массива <code>Array</code> , начинающемся с элемента с заданным индексом и заканчивающемся последним элементом.
<code>IndexOf(Array, Object, Int32, Int32)</code>	Выполняет поиск указанного объекта и возвращает индекс первого вхождения в диапазоне элементов одномерного массива <code>Array</code> , начинающемся с элемента с заданным индексом и содержащем указанное число элементов.

<code>IndexOf<T>(T[], T)</code>	Выполняет поиск указанного объекта и возвращает индекс первого вхождения во всем массиве <code>Array</code> .
<code>IndexOf<T>(T[], T, Int32)</code>	Выполняет поиск указанного объекта и возвращает индекс первого вхождения в диапазоне элементов массива <code>Array</code> , начиная с элемента с заданным индексом и заканчивая последним элементом.
<code>IndexOf<T>(T[], T, Int32, Int32)</code>	Выполняет поиск указанного объекта и возвращает индекс первого вхождения в диапазоне элементов массива <code>Array</code> , начинающемся с элемента с заданным индексом и содержащем указанное число элементов.
<code>Initialize</code>	Инициализирует каждый элемент массива <code>Array</code> типа значения путем вызова конструктора по умолчанию для типа значений.
<code>LastIndexOf(Array, Object)</code>	Выполняет поиск заданного объекта и возвращает индекс его последнего вхождения внутри всего одномерного массива <code>Array</code> .
<code>LastIndexOf(Array, Object, Int32)</code>	Выполняет поиск указанного объекта и возвращает индекс его последнего вхождения в диапазоне элементов одномерного массива <code>Array</code> , начинающемся с первого элемента и заканчивающемся элементом с заданным индексом.
<code>LastIndexOf(Array, Object, Int32, Int32)</code>	Выполняет поиск указанного объекта и возвращает индекс последнего вхождения в диапазоне элементов одномерного массива <code>Array</code> , содержащем указанное число элементов и заканчивающемся элементом с заданным индексом.
<code>LastIndexOf<T>(T[], T)</code>	Выполняет поиск указанного объекта и возвращает индекс последнего вхождения во всем массиве <code>Array</code> .
<code>LastIndexOf<T>(T[], T, Int32)</code>	Выполняет поиск указанного объекта и возвращает индекс последнего вхождения в диапазоне элементов массива <code>Array</code> , начиная с первого элемента и заканчивая элементом с заданным индексом.
<code>LastIndexOf<T>(T[], T, Int32, Int32)</code>	Выполняет поиск указанного объекта и возвращает индекс последнего вхождения в диапазоне элементов массива <code>Array</code> , содержащем указанное число элементов и заканчивающемся в позиции с указанным индексом.

Memberwise Clone	Создает неполную копию текущего объекта Object. (Унаследовано от Object.)
Resize<T>	Изменяет количество элементов в одномерном массиве до указанной величины.
Reverse(Array)	Изменяет порядок элементов во всем одномерном массиве Array на обратный.
Reverse(Array, Int32, Int32)	Изменяет последовательность элементов в диапазоне элементов одномерного массива Array на обратную.
SetValue(Object, Int32)	Присваивает значение элементу, находящемуся в указанной позиции одномерного массива Array. Индекс задается как 32-разрядное целое число.
SetValue(Object, Int32[])	Присваивает значение элементу, находящемуся в указанной позиции многомерного массива Array. Индексы задаются как массив 32-разрядных целых чисел.
SetValue(Object, Int64)	Присваивает значение элементу, находящемуся в указанной позиции одномерного массива Array. Индекс задается как 64-разрядное целое число.
SetValue(Object, Int64[])	Присваивает значение элементу, находящемуся в указанной позиции многомерного массива Array. Индексы задаются как массив 64-разрядных целых чисел.
SetValue(Object, Int32, Int32)	Присваивает значение элементу, находящемуся в указанной позиции двумерного массива Array. Индексы задаются в виде 32-разрядных целых чисел.
SetValue(Object, Int64, Int64)	Присваивает значение элементу, находящемуся в указанной позиции двумерного массива Array. Индексы задаются в виде 64-разрядных целых чисел.
SetValue(Object)	Присваивает значение элементу, находящемуся в указанной позиции

<code>ject, Int32, Int32, Int32)</code>	трехмерного массива <code>Array</code> . Индексы задаются в виде 32-разрядных целых чисел.
<code>SetValue(Object, Int64, Int64, Int64)</code>	Присваивает значение элементу, находящемуся в указанной позиции трехмерного массива <code>Array</code> . Индексы задаются в виде 64-разрядных целых чисел.
<code>Sort(Array)</code>	Сортирует элементы во всем одномерном массиве <code>Array</code> , используя реализацию интерфейса <code>IComparable</code> каждого элемента массива <code>Array</code> .
<code>Sort(Array, Array)</code>	Сортирует пару одномерных объектов <code>Array</code> (один содержит ключи, а другой — соответствующие элементы) по ключам в первом массиве <code>Array</code> , используя реализацию интерфейса <code>IComparable</code> каждого ключа.
<code>Sort(Array, IComparer)</code>	Сортирует элементы в одномерном массиве <code>Array</code> , используя заданный интерфейс <code>IComparer</code> .
<code>Sort(Array, Array, IComparer)</code>	Сортирует пару одномерных объектов <code>Array</code> (один содержит ключи, а другой — соответствующие элементы) по ключам в первом массиве <code>Array</code> , используя заданный интерфейс <code>IComparer</code> .
<code>Sort(Array, Int32, Int32)</code>	Сортирует элементы в диапазоне элементов одномерного массива <code>Array</code> с помощью реализации интерфейса <code>IComparable</code> каждого элемента массива <code>Array</code> .
<code>Sort(Array, Array, Int32, Int32)</code>	Сортирует диапазон элементов в паре одномерных объектов <code>Array</code> (один содержит ключи, а другой — соответствующие элементы) по ключам в первом массиве <code>Array</code> , используя реализацию интерфейса <code>IComparable</code> каждого ключа.
<code>Sort(Array, Int32, Int32, IComparer)</code>	Сортирует элементы в диапазоне элементов одномерного массива <code>Array</code> , используя заданный интерфейс <code>IComparer</code> .
<code>Sort(Array, Array, Int32, Int32, IComparer)</code>	Сортирует диапазон элементов в паре одномерных объектов <code>Array</code> (один содержит ключи, а другой — соответствующие элементы) по ключам в первом массиве <code>Array</code> , используя заданный интерфейс <code>IComparer</code> .

<code>Sort<T>(T[])</code>	Сортирует элементы во всем массиве <code>Array</code> , используя реализацию универсального интерфейса <code>IComparable<T></code> каждого элемента массива <code>Array</code> .
<code>Sort<T>(T[], IComparer<T>)</code>	Сортирует элементы в одномерном массиве <code>Array</code> , используя заданный универсальный интерфейс <code>IComparer<T></code> .
<code>Sort<T>(T[], Comparison<T>)</code>	Сортирует элементы массива <code>Array</code> с использованием указанного объекта <code>Comparison<T></code> .
<code>Sort<T>(T[], Int32, Int32)</code>	Сортирует элементы в диапазоне элементов массива <code>Array</code> , используя реализацию универсального интерфейса <code>IComparable<T></code> каждого элемента массива <code>Array</code> .
<code>Sort<T>(T[], Int32, Int32, IComparer<T>)</code>	Сортирует элементы в диапазоне элементов массива <code>Array</code> , используя заданный универсальный интерфейс <code>IComparer<T></code> .
<code>Sort<TKey, TValue>(TKey[], TValue[])</code>	Сортирует пару одномерных объектов <code>Array</code> (один содержит ключи, а другой — соответствующие элементы) по ключам в первом массиве <code>Array</code> , используя реализацию универсального интерфейса <code>IComparable<T></code> каждого ключа.
<code>Sort<TKey, TValue>(TKey[], TValue[], IComparer<TKey>)</code>	Сортирует пару одномерных объектов <code>Array</code> (один содержит ключи, а другой — соответствующие элементы) по ключам в первом массиве <code>Array</code> , используя заданный универсальный интерфейс <code>IComparer<T></code> .
<code>Sort<TKey, TValue>(TKey[], TValue[], Int32, Int32)</code>	Сортирует диапазон элементов в паре объектов <code>Array</code> (один содержит ключи, а другой — соответствующие элементы) по ключам в первом массиве <code>Array</code> , используя реализацию универсального интерфейса <code>IComparable<T></code> каждого ключа.
<code>Sort<TKey, TValue>(TKey[], TValue[], Int32, Int32, IComparer<TKey>)</code>	Сортирует диапазон элементов в паре объектов <code>Array</code> (один содержит ключи, а другой — соответствующие элементы) по ключам в первом массиве <code>Array</code> , используя заданный универсальный интерфейс <code>IComparer<T></code> .

<code>ey[], TValue[], Int32, Int32, IComparer< TKey>)</code>	массиве <code>Array</code> , используя заданный универсальный интерфейс <code>Comparer<T></code> .
<code>ToString</code>	Возвращает строку, представляющую текущий объект. (Унаследовано от <code>Object</code> .)
<code>TrueForAll< T></code>	Определяет, все ли элементы массива удовлетворяют условиям указанного предиката.

Пример

Создание и инициализация

```
class TestArraysClass
{
    static void Main()
    {
        // Declare a single-dimensional array
        int[] array1 = new int[5];

        // Declare and set array element values
        int[] array2 = new int[] { 1, 3, 5, 7, 9 };

        // Alternative syntax
        int[] array3 = { 1, 2, 3, 4, 5, 6 };

        // Declare a two dimensional array
        int[,] multiDimensionalArray1 = new int[2, 3];

        // Declare and set array element values
        int[,] multiDimensionalArray2 = { { 1, 2, 3 }, { 4, 5, 6 } };

        // Declare a jagged array
        int[][] jaggedArray = new int[6][];

        // Set the values of the first array in the jagged array structure
        jaggedArray[0] = new int[4] { 1, 2, 3, 4 };
    }
}
```

Действия при помощи класса `Array`

```
class Program
{
    static void Main(string[] args)
    {
        Random rand = new Random();

        int[] mas = new int[10] ;
    }
}
```

```

for(int i=0;i<10;i++)
    mas[i] = rand.Next(1,15);

Console.WriteLine("Исходный массив чисел: ");
foreach (int x in myArr)
    Console.Write("\t{0}",x);

// Реализуем сортировку массива
Console.WriteLine("\n\nОтсортированный массив:");
Array.Sort(myArr);
foreach (int x in myArr)
    Console.Write("\t{0}",x);

// Организуем поиск числа 12
Console.WriteLine("\n\nПоиск числа:");
int search = Array.BinarySearch(myArr, 12);
Console.WriteLine("Число 12 находится на {0} позиции",search+1);

Console.ReadLine();
}
}

```

Действия при помощи класса Array и использование лямба выражений

```

static void Main(string[] args)
{
    var mas = new[] { 5, 10, 15, 20, 18 };
    //вывод массива
    Array.ForEach(mas, x => Console.WriteLine(x));
    //Поиск максимума
    int max = mas.Max();
    //Кратные 3м элементы
    Console.WriteLine("Количество кратных 3м = {0}", mas.Count(x =>
x % 3 == 0));
    Console.WriteLine("max = {0}", max);
    //Сохранение в массив нечетных элементов
    int[] mas2 = Array.FindAll(mas, x => x % 2 == 1);
    Array.ForEach(mas2, x => Console.WriteLine("Нечетные {0}",x));
    Array.Sort(mas);
    Console.WriteLine("Кратный 5 = {0}", Array.Find<int>(mas2, x => x
% 5 == 0));
    Array.Reverse(mas);
    Array.ForEach(mas, x => Console.WriteLine(x));
}

```

ПРАКТИЧЕСКАЯ ЧАСТЬ1

Общее задание:

1. Функция для вывода элементов массива static void PrintAll(int/double [] array)

2. Функция для получения нового массива случайно по количеству элементов static int/double[] GetArray(int CountNumber);

Реализовать способ создания последовательностей (массивов) двумя способами на выбор пользователю: ввод ручками, или автоматическое (рандомное)

Для удобства можете создавать абсолютно любые классы, или функции, методы, которые смогут вам помочь в реализации программы

В самой же функции Main не должно быть больше 10 строк.

вариант	Задания
1	<ol style="list-style-type: none">1. Проверить является ли данная числовая последовательность a_1, a_2, \dots, a_n возрастающей.2. Имеется n итоговых оценок студента. Подсчитать количество шестерок, семерок, восьмерок, девяток и десятков (пользователь вводит оценки через пробел)3. Дана последовательность x_1, x_2, \dots, x_n и число y. Подсчитать количество элементов последовательности, которые меньше y.
2	<ol style="list-style-type: none">1. Информация о температуре воздуха за месяц задана в виде массива. Определить, сколько раз температура опускалась ниже 0°C. Пользователь вводит название месяца, и в зависимости от месяца, создается массив под количества дней.2. Дана последовательность x_1, x_2, \dots, x_n, упорядоченная в порядке возрастания и вещественное y. Найти такое k, что $x_k < y \leq x_{k+1}$.3. Проверить является ли данная числовая последовательность a_1, a_2, \dots, a_n убывающей
3	<ol style="list-style-type: none">1. Информация о среднесуточной температуре воздуха за месяц задана в виде массива. Определить, сколько дней температура была ниже среднемесячной. Пользователь вводит название месяца, и в зависимости от месяца, создается массив под количества дней.2. Имеется n итоговых оценок студента. Расположить эти оценки в следующем порядке: десятки, восьмерки, шестерки. Остальные оценки в произвольном порядке. Исходные оценки пользователь вводит с клавиатуры3. Проверить является ли данная числовая последовательность a_1, a_2, \dots, a_n невозрастающей
4	<ol style="list-style-type: none">1. Дан числовой массив A, состоящий из n-элементов (вводит пользователь). Найти среднее арифметическое положительных элементов этого массива.2. Дана последовательность x_1, x_2, \dots, x_n. Подсчитать количество таких троек, что $x_{i-1} < x_i < x_{i+1}$ ($i=2, n-1$)3. Проверить является ли данная числовая последовательность a_1, a_2, \dots, a_n неубывающей
5	<ol style="list-style-type: none">1. Дан числовой массив A, состоящий из n-элементов. Все положительные элементы этого массива уменьшить на 0,5.2. Дана последовательность x_1, x_2, \dots, x_n. Найти номер элемента, который отличается

	<p>от среднего арифметического значения элементов последовательности на минимальную величину.</p> <p>3. Проверить является ли данная числовая последовательность a_1, a_2, \dots, a_n постоянной (все члены равны между собой).</p>
6	<p>1. Дан числовой массив A, состоящий из n-натуральных чисел. Определить количество элементов массива, которые больше заданного числа.</p> <p>2. Дана последовательность x_1, x_2, \dots, x_n. Найти наибольшую сумму подряд идущих элементов (3 элемента) Вывести эти элементы на экран, вывести сумму.</p> <p>3. Дана последовательность целых чисел a_1, a_2, \dots, a_n. Найти и вывести значение и математический индекс наибольшего по модулю элемента последовательности</p>
7	<p>1. Дан числовой массив A, состоящий из n-натуральных чисел. Определить количество элементов массива, являющихся нечетными числами.</p> <p>2. Дана последовательность x_1, x_2, \dots, x_n. Подсчитать количество элементов последовательности, больших среднего арифметического значения положительных элементов последовательности.</p> <p>3. Дана последовательность целых чисел a_1, a_2, \dots, a_n. Найти модуль наибольшего элемента последовательности</p>
8	<p>1. Дан числовой массив A, состоящий из n-натуральных чисел. Определить количество элементов массива, являющихся кратными 7.</p> <p>2. Дана последовательность натуральных чисел x_1, x_2, \dots, x_n. Изменить данную последовательность так, чтобы в начале стояли все четные, а затем нечетные элементы последовательности.</p> <p>4. Дана последовательность целых чисел a_1, a_2, \dots, a_n. Найти и вывести значение и математический индекс наименьшего по модулю элемента последовательности</p>
9	<p>1. Дан числовой массив A, состоящий из n-натуральных чисел. Определить количество элементов массива кратных 3, но не кратных 5.</p> <p>2. Дана последовательность натуральных чисел x_1, x_2, \dots, x_n. Изменить данную последовательность так, чтобы в начале стояли все нулевые элементы, затем отрицательные, а затем положительные элементы последовательности</p> <p>3. Дана последовательность целых чисел a_1, a_2, \dots, a_n. Найти наименьший среди положительных элементов</p>
10	<p>1. Дан числовой массив A, состоящий из n-натуральных чисел. Определить количество элементов массива, удовлетворяющих условию $A_i < (A_{i-1} + A_{i+1})/2$. Вывести данные элементы и их индекс.</p> <p>2. Даны две последовательности чисел a_1, a_2, \dots, a_n и b_1, b_2, \dots, b_n. Подсчитать, какое число раз встречается ситуация, когда $a_i > b_i$ и $a_{i+1} < b_{i+1}$ ($i=1, n-1$). Вывести на экран данные ситуации</p> <p>3. Дана последовательность целых чисел a_1, a_2, \dots, a_n. Найти наибольший среди отрицательных элементов</p>
11	<p>1. Дан числовой массив A, состоящий из n-натуральных чисел. Определить количество элементов массива, удовлетворяющих условию $2k < A_k < 3k$. Вывести данные элементы и их индексы</p> <p>2. Даны две последовательности чисел a_1, a_2, \dots, a_n и b_1, b_2, \dots, b_n. Найти i, при котором $a_i + b_i$ наибольшая из всех таких пар</p> <p>3. Дана последовательность целых чисел a_1, a_2, \dots, a_n. Найти наибольший среди элементов, стоящих на четных местах</p>
12	<p>1. Дан числовой массив A, состоящий из n-натуральных чисел. Определить количество элементов массива, имеющих четные порядковые номера и являющихся нечетными числами. Вывести данные числа на экран</p> <p>2. Даны две последовательности чисел a_1, a_2, \dots, a_n и b_1, b_2, \dots, b_n. Найти количество пар a_i, b_i ($i=1, n$) таких, что оба числа в паре четные. Вывести полученные пары на экран.</p>

	3. Дана последовательность действительных чисел a_1, a_2, \dots, a_n . Найти сумму всех цифр данной последовательности.
13	<p>1. Дан числовой массив A, состоящий из n-натуральных чисел. Определить количество элементов массива, которые при делении на 7 дают остаток 1, 2 или 5. Вывести данные элементы.</p> <p>2. Дана последовательность x_1, x_2, \dots, x_n. Найти наибольший по модулю элемент последовательности среди отрицательных элементов последовательности.</p> <p>3. Дана последовательность действительных чисел a_1, a_2, \dots, a_n. Получить из заданной последовательности другую, в которой сначала бы шли отрицательные члены исходной последовательности, после нулевые, а затем неотрицательные.</p>
14	<p>1. Имеется n итоговых оценок студента. Определить является ли студент отличником (все оценки не ниже 9).</p> <p>2. Дана последовательность x_1, x_2, \dots, x_n. Поменять местами самый большой элемент с самым маленьким.</p> <p>3. Дана последовательность действительных чисел a_1, a_2, \dots, a_n. Найти сумму членов, удовлетворяющих условию $a_k < k$.</p>
15	<p>1. Имеется n итоговых оценок студента. Определить является ли студент неуспевающим (имеются оценки ниже 4), неуспевающий студент это студент у которого таких оценок больше двух</p> <p>2. Дана последовательность x_1, x_2, \dots, x_n. Найти самый большой по модулю отрицательный элемент</p> <p>3. Даны координаты x_i и y_i n-точек на плоскости. Подсчитать количество точек, лежащих в первой четверти.</p>
16	<p>1. Дан числовой массив A, состоящий из n-натуральных чисел. Определить количество элементов массива, имеющих четные порядковые номера и являющихся нечетными числами. Вывести данные числа на экран</p> <p>2. Дана последовательность x_1, x_2, \dots, x_n. Найти наибольшую сумму подряд идущих элементов.(3 элемента) Вывести эти элементы на экран, вывести сумму.</p> <p>3. Дана последовательность целых чисел a_1, a_2, \dots, a_n. Найти наибольший среди отрицательных элементов</p>
17	<p>1. Дан числовой массив A, состоящий из n-элементов. Все положительные элементы этого массива уменьшить на 0,5</p> <p>2. Даны две последовательности чисел a_1, a_2, \dots, a_n и b_1, b_2, \dots, b_n. Найти i, при котором $a_i + b_i$ наибольшая из всех таких пар</p> <p>3. Дана последовательность действительных чисел a_1, a_2, \dots, a_n. Найти сумму всех цифр данной последовательности</p>
18	<p>1. Дан числовой массив A, состоящий из n-натуральных чисел. Определить количество элементов массива, удовлетворяющих условию $A_i < (A_{i-1} + A_{i+1})/2$. Вывести данные элементы и их индекс</p> <p>2. Дана последовательность x_1, x_2, \dots, x_n. Поменять местами самый большой элемент с самым маленьким</p> <p>3. Проверить является ли данная числовая последовательность a_1, a_2, \dots, a_n неубывающей</p>

ПРАКТИЧЕСКАЯ ЧАСТЬ2

Создайте новый проект в текущем решении, создайте в нем класс Runner и выполните в нем проверку следующих заданий

Задание1.

Реализуйте **статические** функции (методы) ввода и простого/форматного вывода для одномерных массивов типа **double**. Разместить эти функции (методы) в классе **InOutArray**.

Сигнатура для ввода: static double[] GetArray(string line)

Сигнатура для вывода: static void PrintArray(double[] array)

Static void PrintArray(double[] array, string name, int countColumn)

Задание2.

Реализуйте **статические** функции (методы) для поиска минимального и максимального значения в одномерном массиве для типа **double**. Реализуйте **статические** функции (методы) для сортировки по возрастанию/убыванию одномерных массивов типа **double**. Реализуйте умножение матрицы на вектор и матрицы на матрицу. Разместить данные функции в классе **Helper**

static double FindMin(double[] array)

static double FindMax(double[] array)

static void Sort(double[] array, bool order) // сортировку можно любую от пузырька до быстрой, главное своя реализация

static double[] Mul(double[,] matrix, double array)

static double[,] Mul(double[,] matrix1, double[,] matrix1)

*Задание3.

Сформировать одномерный массив double array[31], значения элементов которого располанаются произвольным образом в диапазоне от 102 до 354 и округленны до целых. Сформировать из массива array новый отсортированный по возрастанию массив newArray, в который входят те элементы массива array, у которых 10% от значения больше индекса этого элемента в массиве array.

Вывести массивы array и newArray в виде таблиц с псевдографическими границами (количество столбцов не более 5, количество строк должно быть минимальным – определяется программно).

Индекс каждого элемента таблицы должен быть указан, например A[3]=5.6. и индекс в массиве array изменяется по строке, а в массиве newArray — по столбцам

Massiv M <index to rows>				
M[1]=7.0	M[2]=6.9	M[3]=4.4	M[4]=6.4	M[5]=6.2
M[6]=5.0	M[7]=6.0	M[8]=6.5	M[9]=5.9	M[10]=5.7
M[11]=6.8	M[12]=5.6	M[13]=6.8	M[14]=4.5	M[15]=6.8
M[16]=6.1	M[17]=4.3	M[18]=5.9	M[19]=4.7	M[20]=6.0
M[21]=4.9	M[22]=4.5	M[23]=4.8	M[24]=6.1	M[25]=6.5
M[26]=6.0	M[27]=5.6	M[28]=6.3	M[29]=4.5	
Massiv B <index to cols>				
B[1]=4.3	B[8]=5.6			
B[2]=4.5	B[9]=5.9			
B[3]=4.5	B[10]=6.0			
B[4]=4.5	B[11]=6.0			
B[5]=4.7	B[12]=6.1			
B[6]=4.8	B[13]=6.3			
B[7]=4.9	B[14]=6.5			

III. Контрольные вопросы

1. Какие массивы называются неявно типизированными?
2. Динамические методы класса Array
3. Статические методы класса Array

Литература

1. Полный справочник по C#. Г. Шилдт. Издательский дом «Вильямс», 2004.
2. C# в подлиннике. Наиболее полное руководство. Х.Дейтел.
3. C# в задачах и примерах. Культин. Н.Б.
4. C# учебный курс. Г.Шилдт. СПб.: Питер, 2002.
5. C# программирование на языке высокого уровня Павловская Т.А. СПб.:

БХВ-Петербург.