

Лабораторная работа №2

«Массивы и строки»

Цель: целью данной работы является получение опыта работы с массивами и строками на языке высокого уровня C#, в среде программирования Microsoft Visual Studio

Задание:

1. На вход программы подаётся массив целых чисел произвольной длины (из консоли, или заполняется циклом). Далее происходит инвертирование массива, после чего в консоль выводятся исходный и инвертированный массивы.

Пример:

Вход:

1	2	3	4	5
---	---	---	---	---

Выход:

5	4	3	2	1
---	---	---	---	---

Массив — это пронумерованная коллекция элементов одного типа. Массивы используются для хранения и обработки большого количества данных одного типа. Они имеют фиксированный размер и индексируются с нуля.

Инвертирование массива означает изменение порядка элементов массива так, чтобы первый элемент становился последним, второй элемент — вторым с конца и так далее.

Для выполнения 1 задания вам понадобятся операторы:

1. **Console.ReadLine** — для считывания числа из консоли
2. **Console.WriteLine** — для вывода на экран сообщений
3. **int** — Объявление массива целых чисел
4. **for** — цикл, в котором некая результирующая переменная будет умножаться на число N, P раз
5. **Array.Reverse** - эта функция меняет порядок элементов в массиве на обратный

2. На вход программы подаётся размерность двумерного массива $M \times N$, где M обозначает количество строк, а N — количество столбцов в массиве. (значения M и N заданы в коде). Далее двумерный массив заполняется числами от 1 до $M \times N$ “змейкой”, после чего заполненный массив выводится в консоль.

Пример:

Вход: $M = 3, N = 3$;

Выход:

1	2	3
6	5	4
7	8	9

Двумерный массив — это массив, состоящий из одномерных массивов (строк), которые расположены последовательно в памяти. Он представляет собой матрицу, состоящую из строк и столбцов.

Для выполнения 2 задания вам понадобятся операторы:

1. **Console.ReadLine** — для считывания числа из консоли
2. **Console.WriteLine** — для вывода на экран сообщений
3. **int** — Объявление массива целых чисел
4. **for** — цикл, в котором некая результирующая переменная будет умножаться на число N, P раз
5. **if, else** — используются для управления направлением движения "змейки" по массиву.

3. На вход программы подаётся размерность двумерного массива $M \times N$ и одномерный массив размерностью $M \times N$ (значения M и N заданы в коде, массив заполняется в коде, или из консоли). Далее создаётся двумерный массив размерностью $M \times N$ и элементы одномерного массива построчно копируются в двумерный массив, после чего оба массива выводятся в консоль.

Пример:

Вход: $M = 3, N = 3$;

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

Выход:

1	2	3
4	5	6
7	8	9

Для выполнения 3 задания вам понадобятся операторы:

1. **Console.ReadLine** — для считывания числа из консоли
2. **Console.WriteLine** — для вывода на экран сообщений
3. **int** — Объявление массива целых чисел
4. **for** — цикл, в котором некая результирующая переменная будет умножаться на число N, P раз

4. На вход программы подаётся двумерный массив размерностью $M \times N$ (значения M и N заданы в коде). Далее осуществляется поиск максимального и минимального элементов в массиве, после чего массив, а также значения

максимального и минимального элементов выводятся в консоль.

Пример:

1	2	3
4	5	6
7	8	9

Вход: $M = 3, N = 3$;

Выход: $MIN = 1, MAX = 9$;

Для выполнения 4 задания вам понадобятся операторы:

1. **Console.ReadLine** – для считывания числа из консоли
2. **Console.WriteLine** – для вывода на экран сообщений
3. **int** – Объявление массива целых чисел
4. **for** – цикл, в котором некая результирующая переменная будет умножаться на число N, P раз
5. **Random** – в работе этого класса заложен алгоритм, который генерирует последовательность чисел

5. На вход программы подаётся строка символов (задана в коде или из консоли). Далее определяется количество разных символов в строке, после чего в консоль выводится заданная строка и количество разных символов в ней.

Пример:

Вход: “Это тестовая строка.”

Выход: Symbols = 12;

Для выполнения 5 задания вам понадобятся операторы:

1. **Console.ReadLine** – для считывания числа из консоли
2. **Console.WriteLine** – для вывода на экран сообщений
3. **foreach** - используется для итерации по символам в строке

6. На вход программы подаётся строка символов (заполняется в коде, или из консоли). Далее определяется количество повторений для каждого символа, после чего в консоль выводится заданная, и уникальные символы этой строки с количеством их повторений в ней.

Пример:

Вход: “мама мыла раму”

Выход: $M = 4, A = 4, Ы = 1, Л = 1, Р = 1, У = 1$;

Для выполнения 6 задания вам понадобятся операторы:

1. **Console.ReadLine** – для считывания числа из консоли
2. **Console.WriteLine** – для вывода на экран сообщений
3. **int** – Объявление массива целых чисел
4. **foreach** – используется для итерации по символам в строке
5. **TryGetValue** – получает из словаря элемент по ключу key.

7. На вход программы подаётся строка символов (заполняется в коде, или из консоли). Далее пользователь вводит символ и введенный пользователем символ удаляется из строки, после чего получившаяся строка выводится в консоль.

Пример:

Вход: “исходная строка”, “а”

Выход: “исходня строк”

Для выполнения 7 задания вам понадобятся операторы:

1. **Console.ReadLine** – для считывания числа из консоли
2. **Console.WriteLine** – для вывода на экран сообщений
3. **String Replace** – эта функция заменяет все вхождения одного символа на другой

Справочная информация:

Объявление массивов могут выглядеть следующим образом:

```
int[] mas = new int[10];           //одномерный массив на 10 элементов
int[,] mas2D = new int[3,3];       //двумерный массив размером 3x3
```

Можно инициализировать массивы при объявлении:

```
int[] m1 = new int[5] { 1, 2, 3, 4, 5 };           //массив из 5 целых чисел
float[] m2 = { 1.0f, 2.0f, 3.0f };                 //массив из 3 вещественных чисел

int[,] m2d = { {1, 2}, { 3, 4 }, { 5, 6 } };       //объявление и инициализация двумерного массива
```

либо используя циклы:

```
int[] m = new int[10]; //массив целых чисел из 10 элементов

for (int i = 1; i <= 10; i++) //заполнение числами от 1 до 10
    m[i] = i;

int[,] m2d = new int[3,3]; //массив целых чисел из 9 элементов

for (int i = 0; i < 3; i++) //заполнение строк массива числами от 1 до 3
    for (int j = 0; j < 3; j++)
        m2d[i,j] = i + 1;
```

Вывод значений массива на экран можно реализовать при помощи цикла for:

```
int[] m = { 1, 2, 3, 4, 5 };

for (int i = 0; i < m.Length; i++)
    Console.Write("{0} ", m[i]);           //вывод на экран элемента массива и пробела
```

Вывод матрицы в консоль:

```
// Выводим матрицу на консоль
Console.WriteLine("Матрица:");
for (int i = 0; i < rows; i++)
{
    for (int j = 0; j < cols; j++)
    {
        Console.Write(matrix[i, j] + " ");
    }
    Console.WriteLine(); // Переход на новую строку
}
```

Строка может быть объявлена и инициализирована следующим образом:

```
string str;           //создание строковой переменной
str = "Hello World";  //запись текста в строковую переменную
```

Считать строку из консоли можно при помощи метода ReadLine:

```
string str;           //создание строковой переменной
str = Console.ReadLine(); //запись текста в строковую переменную
```

В сущности, строка является массивом символов, поэтому к ней

применимы операции поэлементного доступа:

```
Console.WriteLine(str);    //вывод на экран всей строки
Console.WriteLine(str[3]); //вывод на экран 4го символа строки
```

Так же, к строке можно добавлять символы. Например, функцию создания инвертированной версии строки можно реализовать как:

```
string str;                //создание строковой переменной
str = Console.ReadLine();  //запись текста в строковую переменную
string strInv = "";        //создание пустой строки

for(int i = str.Length-1; i >= 0 ; i--) //цикл, начинающийся в конце строки и идущий к её началу
{
    strInv += str[i];          //добавление символа к пустой строке
}
```

Ссылки на описания используемых функций/операторов:

1. **Console.ReadLine**, **Console.WriteLine**, **Console.ReadKey** – <https://devpractice.ru/c-sharp-lesson-3-console-io/#p33>
2. **if** – <https://devpractice.ru/c-sharp-lesson-6-if-and-for/>
3. **for** – <https://metanit.com/sharp/tutorial/2.6.php>
4. **int** – <https://csharp.net-tutorials.com/ru/380/введение/целые/>
5. **Array.Reverse** – <https://metanit.com/sharp/tutorial/20.5.php>
6. **else** – <https://metanit.com/sharp/tutorial/2.5.php>
7. **Random** – http://plssite.ru/csharp/csharp_random_article.html
8. **foreach** – <https://metanit.com/sharp/tutorial/4.9.php>
9. **TryGetValue** – <https://www.techiedelight.com/determine-key-exists-dictionary-csharp/>
10. **String Replace** – <https://www.iditect.com/guide/csharp/csharp-string-replace.html>