**Лабораторная работа №12**

**Работа с массивами: Вложенные циклы**

**Цели:**

1) Получить навыки работы с вложенными циклами.

2) Закрепить навыки разработки приложения по имеющейся блок-схеме

3) Познакомиться со средствами отладки программ.

**Приложение Lab12\_01. Обращение порядка следования элементов одномерного массива**

***Задание:***

Разработать консольное приложение, обращающее порядок следования элементов одномерного массива на противоположный.

***Константы:***

Отсутствуют

***Переменные:***

v, n, a

***Исходные данные:***

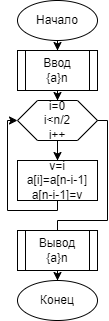
- a – массив данных

- n – размерность массива

***Результат:***

В окне консоли выводится обратный порядок массива.

**Блок-схема программы:**

****

***Код приложения:***

**Листинг 1 – Код консольного приложения Lab11\_01**

namespace GaivoronskayEA.OP.Lab12\_01

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Лабораторная работа №11");

Console.WriteLine("Приложение Lab12\_01");

Console.WriteLine("Разработчик: Гайворонская Е. А");

Console.WriteLine("-------------------------------");

Console.WriteLine("Обращение порядка массива");

Console.WriteLine("Введите размерность массива:");

int n = int.Parse(Console.ReadLine());

double[] a = new double[n];

Console.WriteLine("Заполните массив:");

for (int i = 0; i < n; i++)

{

a[i] = int.Parse(Console.ReadLine());

}

double v;

for (int i = 0; i < n / 2; i++)

{

v = a[i];

a[i] = a[n - i - 1];

a[n - i - 1] = v;

}

Console.WriteLine("Перевернутый массив:");

for (int i = 0; i < n; i++)

{

Console.WriteLine(a[i]);

}

Console.ReadLine();

}

}

}

**Тестирование:**

Пример 1)

**Исходные данные:**

Длинна массива 3

Массив [1, 2, 3]

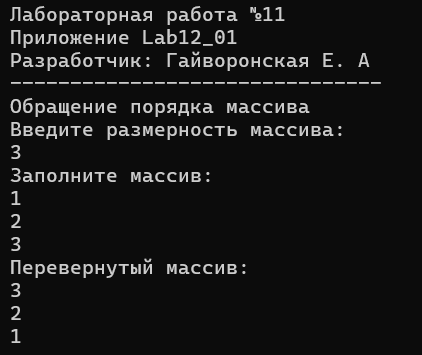


Рисунок 1 – Консольное приложение Lab12\_01

**Приложение Lab12\_02. Знакомство со средствами отладки**

***Задание:***

Познакомиться со средствами отладки кода на примере приложения, выполняющего обращение порядка следования элементов одномерного массива на противоположный

1. Для запуска пошагового выполнения воспользуемся пунктом главного меню «Отладка / Шаг с обходом» на рисунке 12.2.

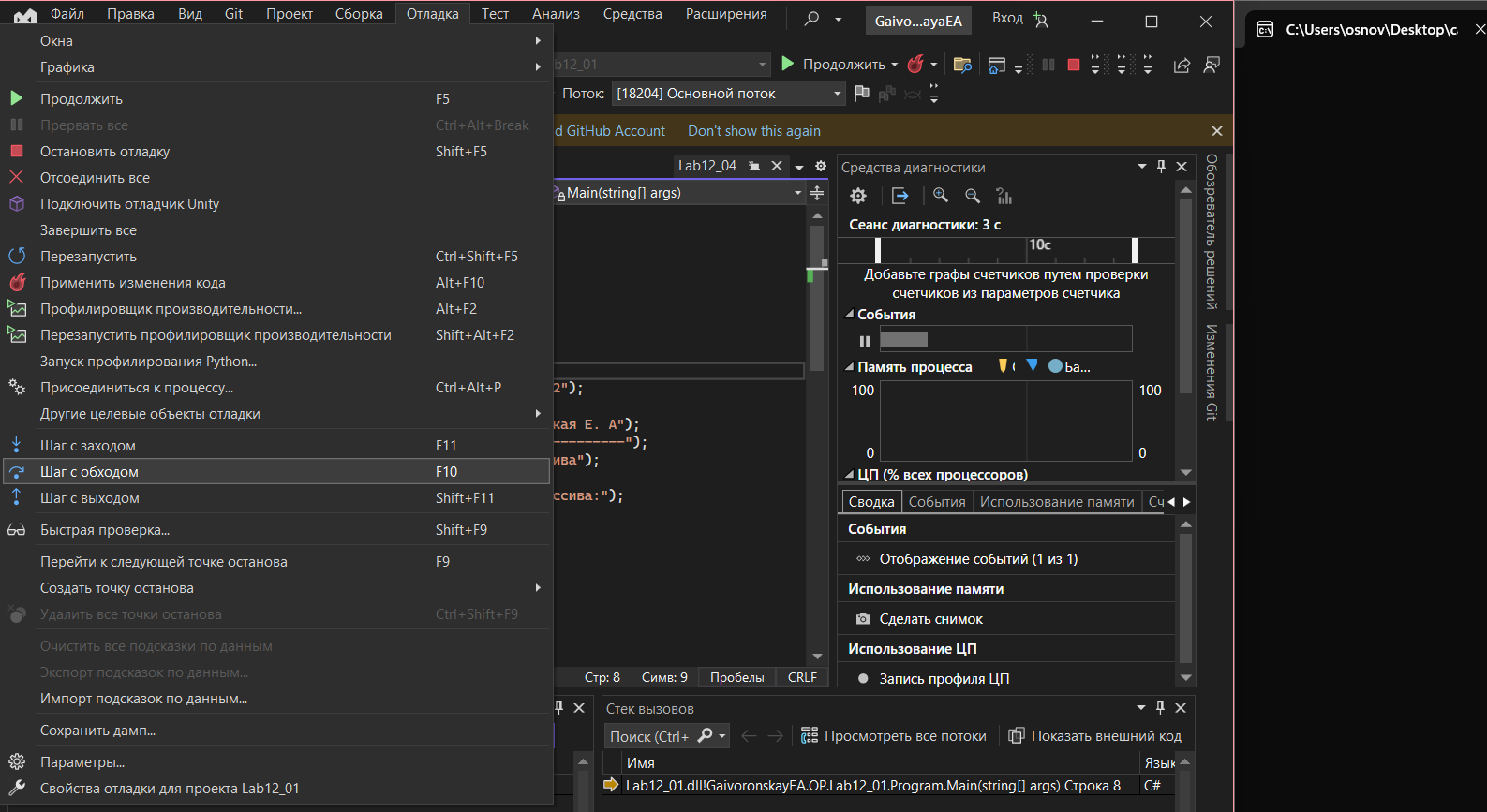


Рисунок 12.2 Запуск пошагового выполнения

При нажатии приложение запустится: промелькнет пустое консольное окно, а в редакторе кода будет выделена первая строка, следующая за

static void Main(string[] args)

как показано на рисунке 12.3.

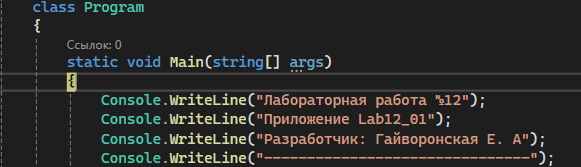
****

Рисунок 12.3 Начальное состояние при пошаговом выполнении

2. Нажмем «Шаг с обходом» еще два раза. Среда перейдет в состояние, изображенное на рисунке 12.4

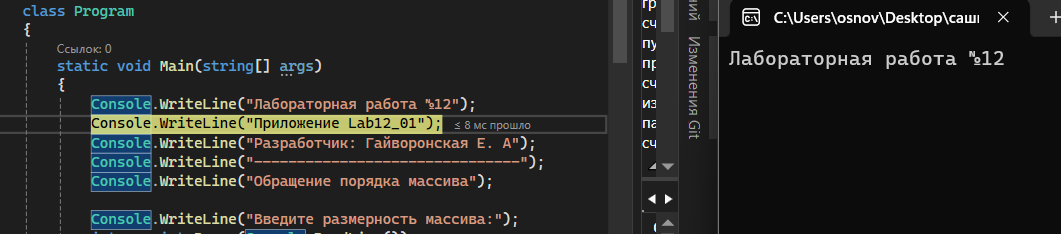
****

Рисунок 12.4 Состояние приложения до ввода исходных данных

3. Выполним еще пару шагов, используя «Шаг с обходом». Зафиксируем, как добавляются текст в окно консоли на рисунке 12.5

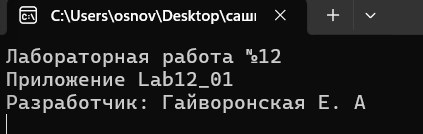
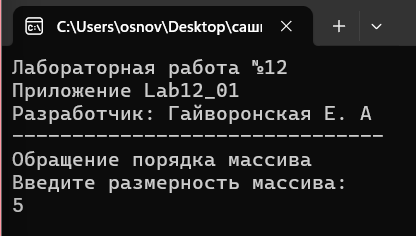
****

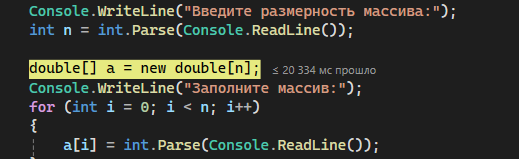
Рисунок 12.5 Состояние приложения в момент ввода информации о задаче

Выполним шаги до тех пор, пока не будет выделена строка

int n = int.Parse(Console.ReadLine());

Нажмем «Шаг с обходом». Появится окно консоли. Введем значение «5» и нажмем . В редакторе кода будет выделена следующая строка, а в окне «Локальные» значение переменной n изменится с «0» на «5» на рисунке 12.6.

****

****

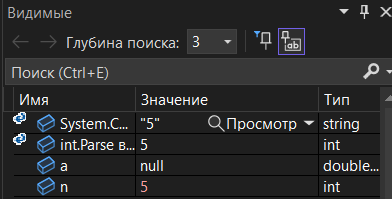
****

Рисунок 12.6 Состояние массива после определения размерности

4. Нажмем «Шаг с обходом» и зафиксируем, как изменится значение переменной a. Для этого развернем содержимое образовавшегося узла на рисунке 12.7. Массив по умолчанию заполняется нулями, элементы нумеруются с 0.

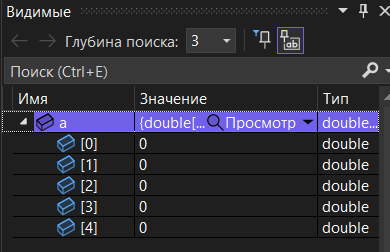


Рисунок 12.7 Состояние массива после выделения памяти

5. Выполним «Шаг с обходом» еще раз. Мы вошли в заголовок цикла. В список переменных в окне «Локальные» добавилась переменная-счетчик i. На рисунке12.8

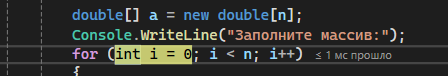


Рисунок 12.8 Состояние приложения при вводе элемента массива

Нажмем «Шаг с обходом». Выделено условие продолжения цикла. Значение условия в текущий момент узнаем, наведя курсор мыши на знак <. В данный момент условие истинно. Рисунок 12.9

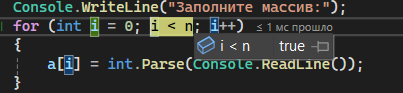
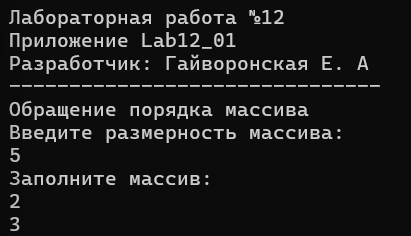
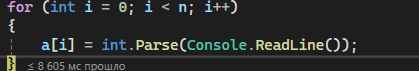


Рисунок 12.9 Значение условия продолжения цикла

6. Выполним «Шаг с обходом» несколько раз, выполняя в окне консоли ввод двух элементов массива по требованию.

Отметим в какие моменты производится модификация счетчика i, когда выполняется повторная проверка условия продолжения цикла. Зафиксируем, как меняются значения элементов массива в окне «Локальные» Рисунок 12.10





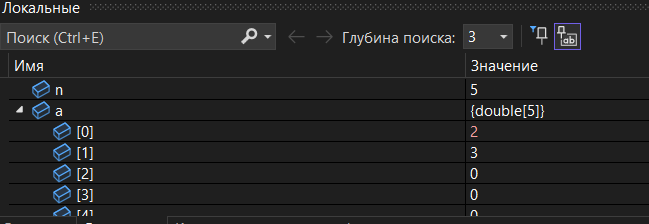


Рисунок 12.10 Состояние приложения при вводе элементов массива

7. В выполнение программы можно вмешиваться. Например, можно изменить значение любой переменной в окне «Локальные». Изменим значение нулевого элемента массива, кликнув по соответствующей ячейке таблицы и введя новое число. рисунок 12.11

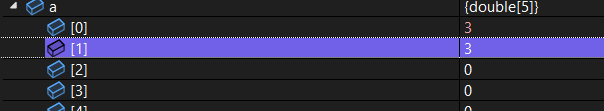


Рисунок 12.11 Изменение значения элемента массива

Второй способ вмешательства заключается в изменении порядка выполнения операторов программы.

Перетащим желтую стрелку на следующую строку: Console.WriteLine("Приложение Lab12\_01");

Отметим, что значения переменных при этом не изменятся

рисунок 12.12



Рисунок 12.12 Изменение выполнения порядка оператора

Выполним пару шагов, используя «Шаг с обходом». Зафиксируем, изменения в окне консоли. Происходит вывод строк, согласно выполняемым операторам. (рисунок 12.13).

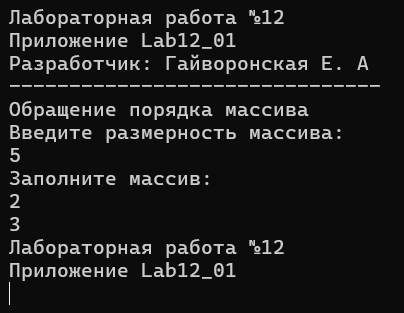


Рисунок 12.13 Вывод, после изменения порядка выполнения оператора

8. Выполним ввод массива еще раз. Для того, чтобы не выполнять его в пошаговом режиме, поставим точку останова на строке

for (int i = 0; i < n / 2; i++)

Для этого кликнем по серому полю слева от нее (рисунок 12.14).

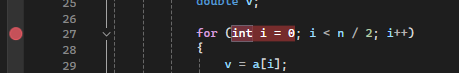
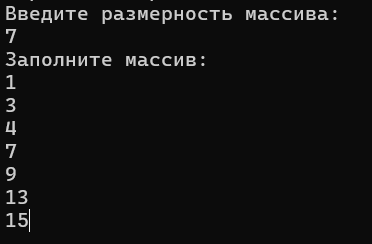


Рисунок 12.14 – Точка останова

Выполним ввод массива из 7 элементов:1, 3, 4, 7, 9, 13, 15.

Для этого нажмите кнопку F5 и введем в окне консоли заданную размерность и значения элементов. 10 По окончании ввода приложение окажется в состоянии, представленном на рисунке 12.15.





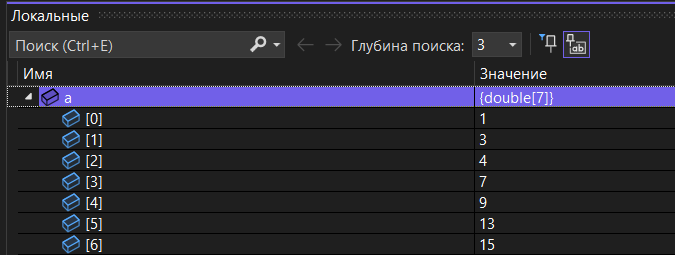
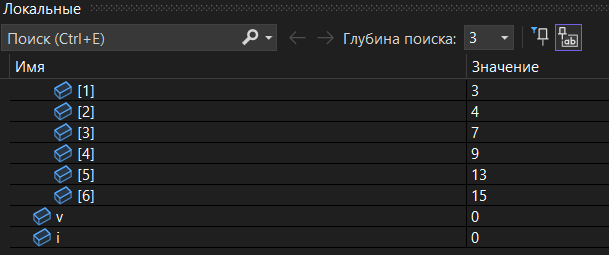
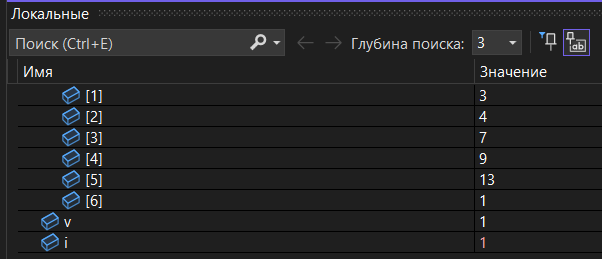


Рисунок 12.15 Состояние приложения после ввода массива

Для записи условия продолжения цикла for использована операция деления. Оба ее операнда являются целыми числами (int), поэтому она рассматривается как целочисленное деление. Убедимся в этом, наведя курсор мыши на символ операции «/»: результат деления 7 на 2 равен 

9. Выполним в пошаговом режиме цикл перестановки элементов. Проследим, как перемещаются значения между переменными, установим роль переменной v. Зафиксируем процесс перестановки для одной итерации цикла (рисунок 12.16)





12.16 Процесс перестановки для одной итерации цикла

Выполним программу в пошаговом режиме, пока не будет выделена строка

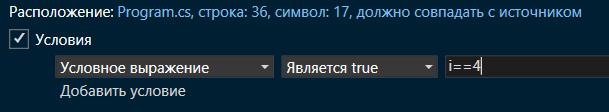
Console.WriteLine("Обращенный массив:");

10. Ранее мы использовали точку останова, чтобы «пропустить» ввод массива. Пропустим вывод первых 4-х элементов массива.

Для этого установим точку останова в теле цикла и добавим условие, согласно которому остановку нужно выполнить только в том случае, если значение счетчика i равно 4.

Поставим точку останова так, как показано на рисунке 12.17, и нажмем на кнопку «Параметры» рядом с ней. Зададим условие, как показано на рисунке и нажмем «Закрыть».





12.17 Точка останова с условием

Продолжим выполнение приложения. Для этого нажмем кнопку F5. Зафиксируем, какая информация была выведена в окно консоли (рисунок 12.18)

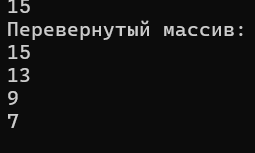


Рисунок 12.18 Результат выполнения до точки останова

11. Для того, чтобы посмотреть, выполнением какого именно оператора занята программа в данный момент, можно использовать кнопку ( ) «Прервать все» панели инструментов.

Нажмем ее. Среда с помощью выделения зеленым цветом покажет, что приложение остановилось на строке

Console.ReadLine();

как показано на рисунке 12.19.

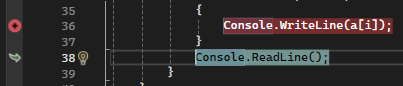


Рисунок 12.19 Определение участка кода, который выполняется в данный момент

Завершим отладку нажав на “Остановить отладку”

**Приложение Lab12\_03. Сортировка с выбором.**

***Задание:***

Разработать консольное приложение, реализующее алгоритм сортировки одномерного массива по возрастанию с выбором наибольшего. С помощью средств отладки проследить, как выполняется поиск наибольшего и перестановка элементов на некотором k-м шаге внешнего цикла.

***Константы:***

max = NegativeInfinity

imax = -1

***Переменные:***

max, n, a

***Исходные данные:***

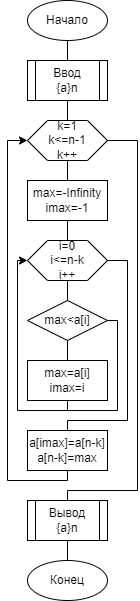
- a – массив данных

- n – размерность массива

***Результат:***

В окне консоли выводится отсортированный массив по возрастанию с выбором наибольшего.

**Блок-схема программы:**

****

***Код приложения:***

**Листинг 3 – Код консольного приложения Lab12\_03**

namespace GaivoronskayEA.OP.Lab12\_03

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Лабораторная работа №12");

Console.WriteLine("Приложение Lab12\_03");

Console.WriteLine("Разработчик: Гайворонская Е. А");

Console.WriteLine("-------------------------------");

Console.WriteLine("Сортировка с выбором");

Console.WriteLine("-------------------------------");

Console.WriteLine("Введите размер массива:");

int n = int.Parse(Console.ReadLine());

double[] a = new double[n];

Console.WriteLine("Заполните массив");

for (int i = 0; i < n; i++)

{

a[i] = int.Parse(Console.ReadLine());

}

for (int k = 1; k <= n - 1; k++)

{

double max = double.NegativeInfinity;

int imax = -1;

for (int i = 0; i <= n - k; i++)

{

if (max < a[i])

{

max = a[i];

imax = i;

}

}

a[imax] = a[n - k];

a[n - k] = max;

}

Console.WriteLine("Отсортированный массив:");

for (int i = 0; i < n; i++)

{

Console.WriteLine(a[i]);

}

}

}

}

**Тестирование:**

Пример 1)

**Исходные данные:**

Длинна массива 5

Массив [5, 4, 3, 2, 1]

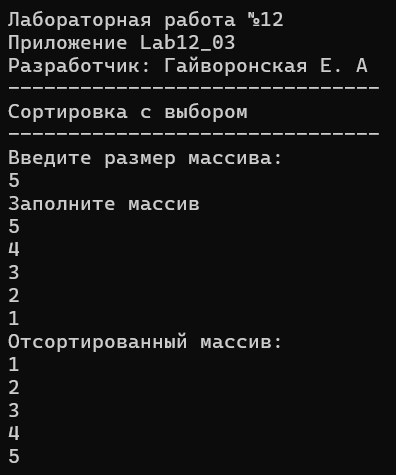


Рисунок 3 – Консольное приложение Lab12\_03

**Приложение Lab11\_04. Сортировка методом попарного сравнения**

***Задание:***

Разработать консольное приложение, реализующее алгоритм сортировки одномерного массива по возрастанию методом попарного сравнения

***Константы:***

Отсутствуют

***Переменные:***

n, a

***Исходные данные:***

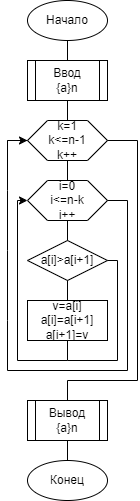
- a – массив данных

- n – размерность массива

***Результат:***

В окне консоли выводится отсортированный массив в порядке возрастания

**Блок-схема программы:**

****

***Код приложения:***

**Листинг 4 – Код консольного приложения Lab12\_04**

namespace GaivoronskayEA.OP.Lab11\_04

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Лабораторная работа №12");

Console.WriteLine("Приложение Lab12\_04");

Console.WriteLine("Разработчик: Гайворонская Е. А");

Console.WriteLine("-------------------------------");

Console.WriteLine("Метод попарного сравнения");

Console.WriteLine("Введите размерность массива:");

int n = int.Parse(Console.ReadLine());

double[] a = new double[n];

Console.WriteLine("Введите массив:");

for (int i = 0; i < n; i++)

{

a[i] = int.Parse(Console.ReadLine());

}

for (int k = 1; k < n; k++)

{

for (int i = 0; i < n - k; i++)

{

if (a[i] > a[i + 1])

{

double v = a[i];

a[i] = a[i + 1];

a[i + 1] = v;

}

}

}

Console.WriteLine("Отсортированный массив:");

for (int i = 0; i < n; i++)

{

Console.WriteLine(a[i]);

}

}

}

}

**Тестирование:**

Пример 1)

**Исходные данные:**

Длинна массива 5

Массив [6, 4, 3, 2, 1]

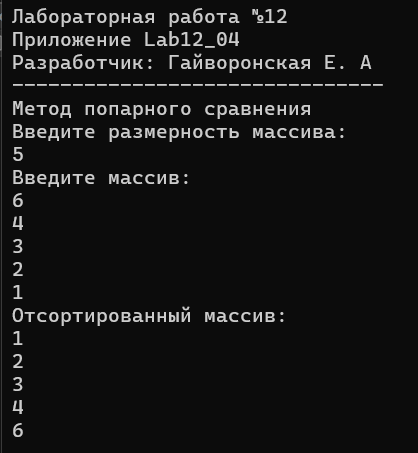


Рисунок 4 – Консольное приложение Lab12\_04