Министерство транспорта Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования

«Российский университет транспорта»

(ФГАОУ ВО РУТ(МИИТ), РУТ (МИИТ)

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

Лабораторная работа № 9

по дисциплине: «Программирование и основы алгоритмизации»

на тему: «Одномерные массивы»

Выполнил: ст. гр. ТУУ-111

Богомолов В.Н.

Вариант №5

25.12.2024

(дата выполнения)

Проверил: к.т.н., доц. Сафронов А.И.

\_\_.12.2024

(дата приёмки)

Москва – 2024 г.

1. **Цель**

Изучить и реализовать алгоритм обработки одномерных массивов с использованием интегрированной среды разработки (*IDE*) *Microsoft Visual Studio* и языка программирования *Visual C#*. Особое внимание уделяется перестановке элементов массива в соответствии с заданными условиями.

1. **Формулировка задачи**

Разработать программу на языке *Visual C#*, которая выполняет перестановку элементов одномерного массива целых чисел длины n, чтобы все положительные элементы предшествовали отрицательным, сохраняя их взаимный порядок.

1. **Блок-схема алгоритма**

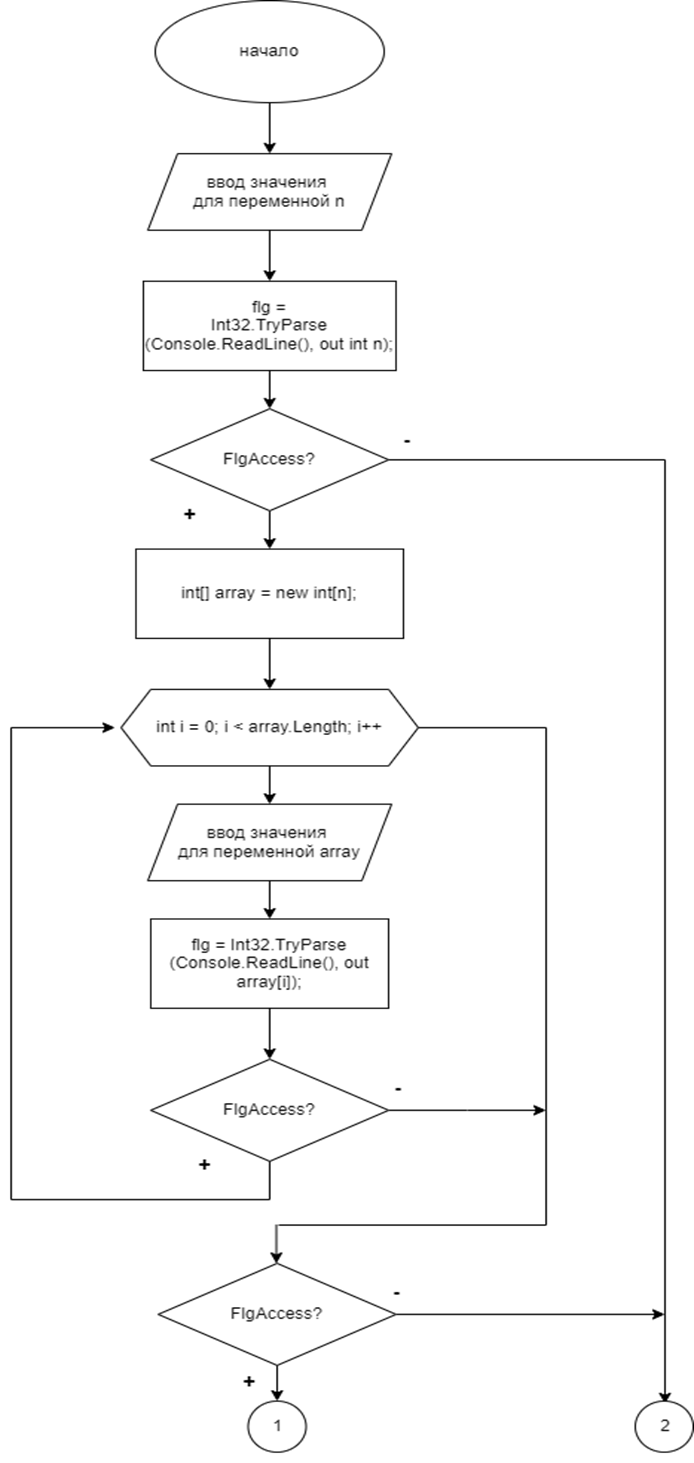
****

Рисунок 3.1.1 – Блок-схема алгоритма

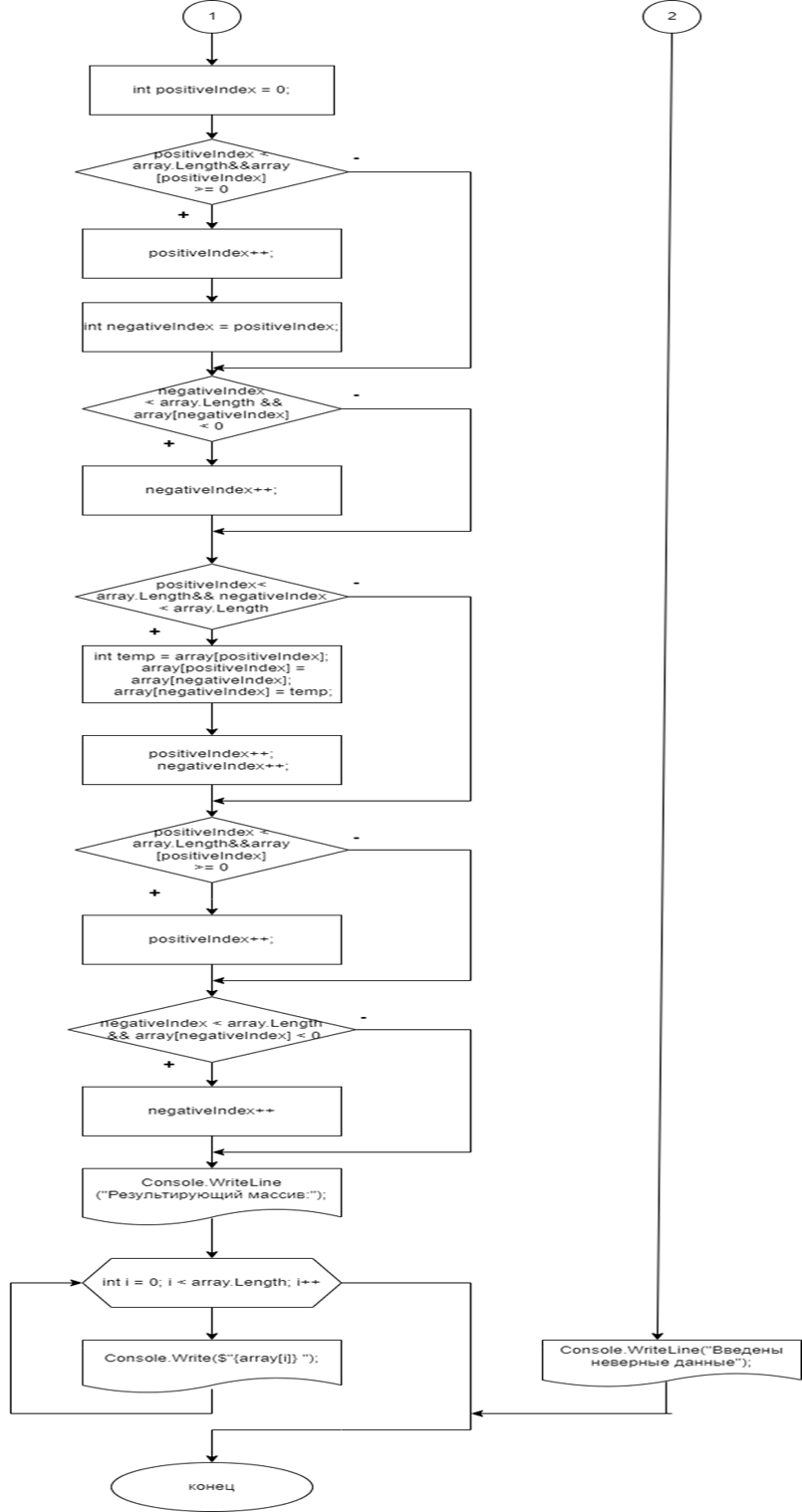
****

Рисунок 3.1.2 – Блок-схема алгоритма

1. **Подбор тестовых примеров**
   1. Тестовый пример №1

исходный массив: (1, -7, 3, 0, -9, -1, 5)

n=7

1-й эл. массива = 1

2-й эл. массива = -7

3-й эл. массива = 3

4-й эл. массива = 0

5-й эл. массива = -9

6-й эл. массива = -1

7-й эл. массива = 5

Следовательно, новый массив: (1, 3, 0, 5, -9, -1, -7)

* 1. Тестовый пример №2

исходный массив: (1, -7, 3, 0, -9, -1, t)

n=7

1-й эл. массива = 1

2-й эл. массива = -7

3-й эл. массива = 3

4-й эл. массива = 0

5-й эл. массива = -9

6-й эл. массива = -1

7-й эл. массива = t

Значение для седьмого элемента массива не является числом, следовательно, введены некорректные данные.

1. **Листинг (код) программы**

using System; // Подключение стандартного пространства имен для работы с основными функциями ввода-вывода

class Program

{

static void Main(string[] args)

{

// Запрашиваем у пользователя длину массива

Console.Write("Введите длину массива: ");

// Пытаемся преобразовать введенную строку в целое число

// Если преобразование успешно, результат сохраняется в переменной n, а flg принимает значение true

bool flg = Int32.TryParse(Console.ReadLine(), out int n);

// Проверяем, удалось ли корректно преобразовать введенные данные

if (flg == true)

{

// Создаем массив целых чисел длиной n

int[] array = new int[n];

// Цикл для ввода элементов массива пользователем

for (int i = 0; i < array.Length; i++)

{

// Просим пользователя ввести i-й элемент массива

Console.Write($"Введите {i + 1}-й элемент массива: ");

// Пытаемся преобразовать введенную строку в целое число и записать в массив

flg = Int32.TryParse(Console.ReadLine(), out array[i]);

// Если введенные данные некорректны, выходим из цикла

if (flg == false)

{

break; // Прерываем выполнение ввода, если обнаружена ошибка

}

}

// Проверяем, успешно ли были введены все элементы массива

if (flg == true)

{

// Инициализируем индекс для поиска положительных чисел

int positiveIndex = 0;

// Находим первый отрицательный элемент в массиве

while (positiveIndex < array.Length && array[positiveIndex] >= 0)

{

positiveIndex++; // Перемещаемся к следующему элементу, пока он не станет отрицательным

}

// Устанавливаем индекс для поиска отрицательных элементов

int negativeIndex = positiveIndex;

// Находим следующий положительный элемент после отрицательного

while (negativeIndex < array.Length && array[negativeIndex] < 0)

{

negativeIndex++; // Пропускаем все отрицательные элементы

}

// Основной цикл перестановки положительных и отрицательных чисел

while (positiveIndex < array.Length && negativeIndex < array.Length)

{

// Обмен значениями между положительным и отрицательным элементами

int temp = array[positiveIndex];

array[positiveIndex] = array[negativeIndex];

array[negativeIndex] = temp;

// Сдвигаем оба индекса к следующим элементам

positiveIndex++;

negativeIndex++;

// Находим следующий положительный элемент

while (positiveIndex < array.Length && array[positiveIndex] >= 0)

{

positiveIndex++;

}

// Находим следующий отрицательный элемент

while (negativeIndex < array.Length && array[negativeIndex] < 0)

{

negativeIndex++;

}

}

// Выводим результирующий массив

Console.WriteLine("Результирующий массив:");

for (int i = 0; i < array.Length; i++)

{

// Выводим каждый элемент массива через пробел

Console.Write($"{array[i]} ");

}

}

else

{

// Сообщение об ошибке, если хотя бы один из элементов массива введен некорректно

Console.WriteLine("Введены неверные данные");

}

}

else

{

// Сообщение об ошибке, если длина массива введена некорректно

Console.WriteLine("Введены неверные данные");

}

// Завершаем выполнение программы, ожидая нажатия клавиши, чтобы пользователь успел увидеть результат

Console.ReadKey(true);

}

}

1. **Расчет тестовых примеров на ПК**

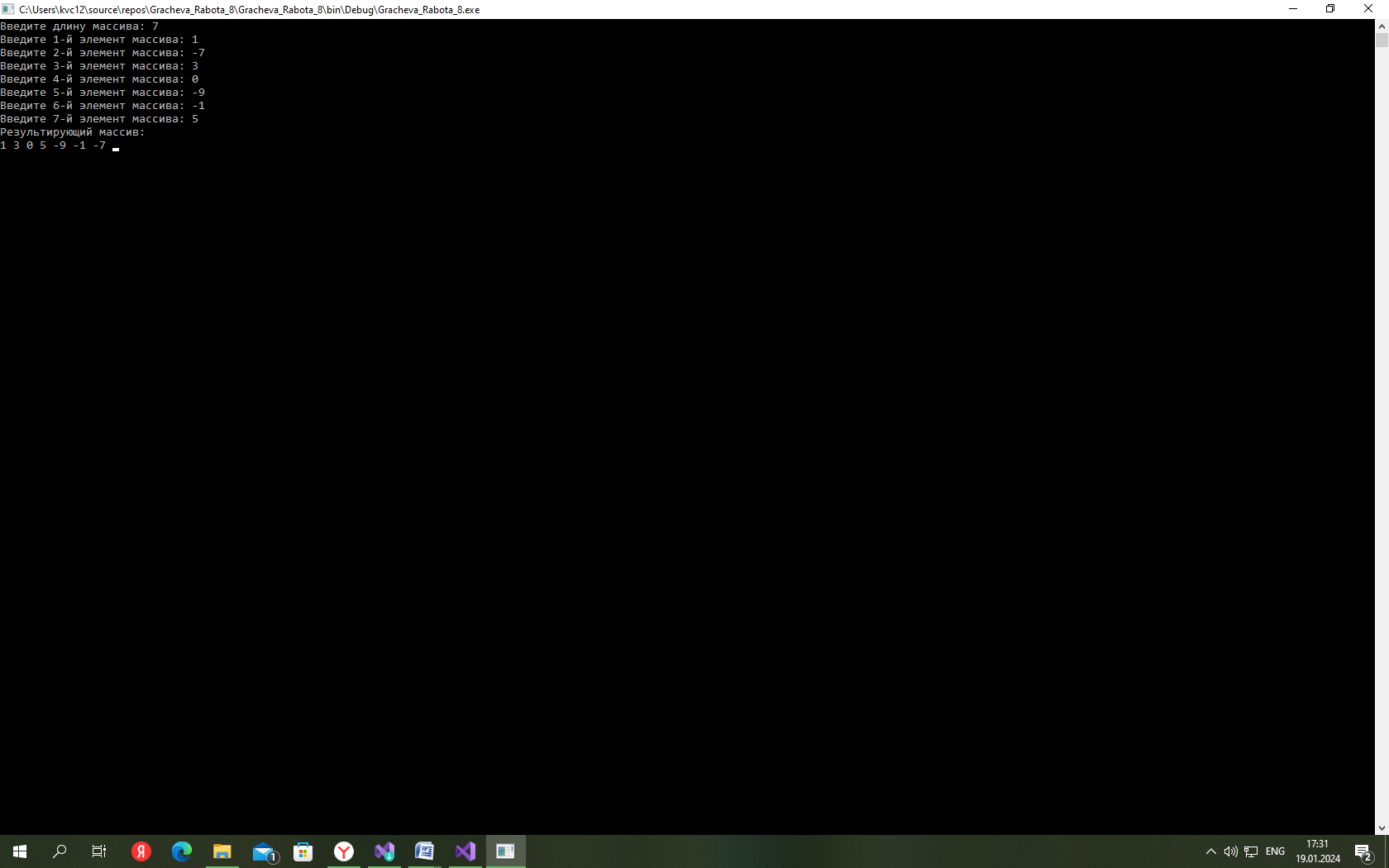


Рисунок 6.1.1 – Тестовый пример №1

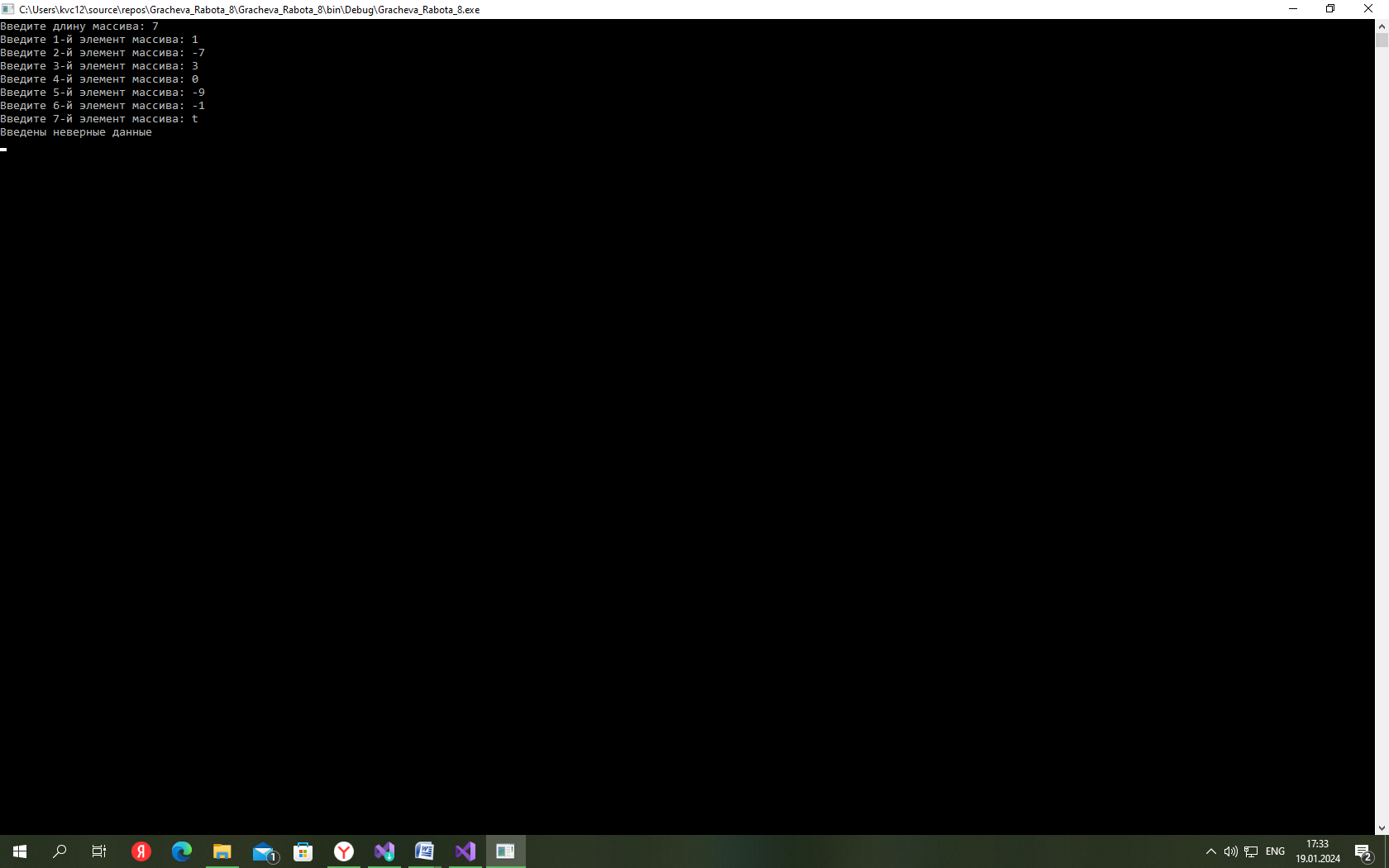


Рисунок 6.1.2 – Тестовый пример №2

1. **Вывод**

В ходе выполнения лабораторной работы было изучено использование одномерных массивов в языке программирования *Visual C#,* а также реализован алгоритм перестановки элементов массива в соответствии с заданными условиями. Были закреплены навыки работы с интегрированной средой разработки *Microsoft Visual Studio* и применения структурированных данных для решения практических задач.