

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ
ТЕХНИКИ

Утвержден на заседании кафедры

«Вычислительная техника» _____

" ____ " _____ 20 ____ г.

Заведующий кафедрой

_____ М.А. Митрохин

ОТЧЕТ ПО УЧЕБНОЙ (ОЗНАКОМИТЕЛЬНОЙ) ПРАКТИКЕ

(2022/2023 учебный год)

Гераскина Дарья Андреевна

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Наименование профиля подготовки «Программное обеспечение средств
вычислительной техники и автоматизированных систем»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 4 года

Год обучения _____ 1 _____ семестр _____ 2 _____

Период прохождения практики с 29.06.2023 по 12.07.2023

Кафедра «Вычислительная техника» _____

Заведующий кафедрой д.т.н., профессор, Митрохин М.А. _____

(должность, ученая степень, ученое звание, Ф.И.О.)

Руководитель практики д.т.н., профессор, Зинкин С.А.

(должность, ученая степень, ученое звание)

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ
ТЕХНИКИ

Утвержден на заседании кафедры

«Вычислительная техника» _____

" ____ " _____ 20 ____ г.

Заведующий кафедрой

_____ М.А. Митрохин

**ИНДИВИДУАЛЬНЫЙ ПЛАН ПРОХОЖДЕНИЯ УЧЕБНОЙ
(ОЗНАКОМИТЕЛЬНОЙ) ПРАКТИКИ**

(2022/2023 учебный год)

Гераськина Дарья Андреевна

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Наименование профиля подготовки «Программное обеспечение средств
вычислительной техники и автоматизированных систем»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 4

года Год обучения _____ 1 _____ семестр _____ 2 _____ Период

прохождения практики с 29.06.2023 по 12.07.2023

Кафедра «Вычислительная техника»

Заведующий кафедрой д.т.н., профессор, Митрохин М.А. _____

(должность, ученая степень, ученое звание, Ф.И.О.)

Руководитель практики д.т.н., профессор, Зинкин С.А. _____

(должность, ученая степень, ученое звание)

№ п / п	Планируемая форма работы во время практики	Количество часов	Календарные сроки проведения работы	Подпись руководите ля практики от вуза
1	Выбор темы и разработка индивидуально го плана проведения работ	2	29.06.2023 - 29.06.2023	
2	Подбор и изучение материала по теме работы	1 5	30.06.2023 – 02.07.23	
3	Разработка алгоритма	4 3	02.07.23 – 06.07.23	
4	Описание алгоритма и программы	1 8	6.07.23 – 08.07. 23	
5	Тестирование	5	08.07.23 – 08.07.23	
6	Получение и анализ результатов	1 0	08.07.23 – 10.07.23	
7	Оформление отчёта	1 5	10.07.23 – 12.07.2023	
	Общий объём часов	1 0 8		

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

ОТЧЁТ

О ПРОХОЖДЕНИИ УЧЕБНОЙ (ОЗНАКОМИТЕЛЬНОЙ) ПРАКТИКИ

(2022/2023 учебный год)

Гераськина Дарья Андреевна

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Наименование профиля подготовки «Программное обеспечение средств
вычислительной техники и автоматизированных систем»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 4 года

Год обучения 1 семестр 2

Период прохождения практики с 29.06.2023 по 12.07.2023

Кафедра «Вычислительная техника»

Гераськина Д.А. выполняла практическое задание «Шейкерная сортировка». На первоначальном этапе были изучен и проанализирован алгоритм шейкерной сортировки, был выбран метод решения и язык программирования С, на котором была написана программа сортировки массива. Также осуществила работу с файлами. Протестировала и отладила программу. Оформила отчёт.

Бакалавр Гераськина Д.А. " " 2023 г.

Руководитель Зинкин С.А. практики 2023 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

ОТЗЫВ

О ПРОХОЖДЕНИИ УЧЕБНОЙ (ОЗНАКОМИТЕЛЬНОЙ) ПРАКТИКИ

(2022/2023 учебный год)

Гераськина Дарья Андреевна

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Наименование профиля подготовки «Программное обеспечение средств
вычислительной техники и автоматизированных систем»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 4 года

Год обучения 1 семестр 2

Период прохождения практики с 29.06.2023 по 12.07.2023

Кафедра «Вычислительная техника»

В процессе выполнения практики Гераськина Д. А. решала следующие задачи: выбор решения для разработки алгоритма; добавление в код чтения данных из файла и записи отсортированного массива; тестирование готовой программы на разных входных данных, наборах содержащих от 100 до 1000 элементов.

За выполнение работы Гераськина Д. А. заслуживает оценки «_____».

Руководитель практики д.т.н., профессор, Зинкин С.А. « _____ » 2023 г.

Содержание

Введение.....	7
Выбор решения.....	8
Постановка задачи.....	10
1.1 Преимущества.....	10
1.2 Недостатки.....	10
1.3 Типичные сценарии применения.....	11
Описание программы.....	12
Схемы программы.....	15
4.2 Блок-схема алгоритма.....	15
Тестирование программы.....	16
5.1 Тестирование на разных наборах данных готовой программы.....	16
5.2 Анализ полученных результатов тестирования.....	17
Отладка.....	19
Совместная работа.....	20
Заключение.....	22
Список используемой литературы.....	23
Приложение А.....	24
Приложение Б.....	29

Введение

Microsoft Visual Studio — это интегрированная среда разработки (IDE), созданная компанией Microsoft. Она используется для разработки различных программных приложений, включая настольные, веб-, мобильные и облачные приложения. Visual Studio предоставляет обширный набор инструментов и функций, которые помогают разработчикам на протяжении всего жизненного цикла разработки программного обеспечения.

Некоторые ключевые особенности Visual Studio включают:

1. Редактор кода: Visual Studio предлагает мощный редактор кода с поддержкой IntelliSense, поддерживает несколько языков программирования, включая C#, VB.NET, F#, C++, JavaScript и другие.
2. Инструменты отладки.
3. Система проектов.
4. Интегрированное тестирование.
5. Сотрудничество и контроль версий.
6. Расширяемость.
7. Разработка облачных приложений.

Язык программирования Си (C) является одним из наиболее популярных и влиятельных языков программирования. Он был разработан в начале 1970-х годов Деннисом Ритчи в Bell Labs и с тех пор стал широко используемым в индустрии разработки программного обеспечения.

Язык программирования Си остается популярным среди разработчиков, особенно в области системного программирования, встроенных систем и разработки низкоуровневого ПО. Он также является хорошим выбором для изучения основ программирования, поскольку множество концепций и подходов, применяемых в Си, переносятся на другие языки программирования.

Выбор решения

Алгоритм шейкерной сортировки (CocktailSort), также известный как сортировка перемешиванием или коктейльная сортировка, является улучшением алгоритма пузырьковой сортировки. Он выполняет сортировку элементов списка путем многократного прохода через список, меняя местами соседние элементы, если они находятся в неправильном порядке.

Краткое описание шейкерной сортировки:

1. Начинаем с исходного списка элементов, который нужно отсортировать.
2. Устанавливаем два указателя: один на начало списка (левый указатель), а другой на конец списка (правый указатель).
3. Пока левый указатель меньше правого указателя:
 - 1) Проходим по списку слева направо, сравнивая пары соседних элементов.
 - 2) Если текущий элемент больше следующего элемента, меняем их местами.
 - 3) После каждого прохода справа налево самый большой элемент перемещается в конец списка.
4. Уменьшаем правый указатель на 1 (так как самый большой элемент уже находится в конце списка).
5. Пока левый указатель меньше правого указателя:
 - 1) Проходим по списку справа налево, сравнивая пары соседних элементов.
 - 2) Если текущий элемент меньше предыдущего элемента, меняем их местами.
 - 3) После каждого прохода слева направо самый маленький элемент перемещаем в начало списка.

6. Увеличиваем левый указатель на 1 (так как самый маленький элемент уже находится в начале списка).

7. Повторяем шаги 3-6 до тех пор, пока левый указатель не станет больше или равным правому указателю.

После завершения алгоритма шейкерной сортировки, элементы списка будут упорядочены по возрастанию (или по другому заданному порядку), начиная с первого элемента и заканчивая последним элементом.

Постановка задачи

1.1 Преимущества

1) Простота реализации: Шейкерная сортировка легко реализуется и понимается. Она не требует сложных структур данных или специальных операций.

2) Устойчивость: Алгоритм шейкерной сортировки является устойчивым, что означает, что элементы с одинаковыми значениями сохраняют свой относительный порядок после сортировки.

3) Хорошая производительность в некоторых случаях: Шейкерная сортировка может быть эффективна на частично отсортированных списках или списках с небольшим количеством элементов, так как она может обнаружить предварительно отсортированную часть и прекратить дальнейшую обработку.

1.2 Недостатки

1) Низкая эффективность в худшем случае: Шейкерная сортировка имеет квадратичную сложность времени в худшем случае, то есть время выполнения зависит от квадрата количества элементов. Это делает ее неэффективной для больших массивов данных.

2) Ограниченная эффективность на случайных данных: По сравнению с некоторыми более сложными алгоритмами сортировки, шейкерная сортировка может быть менее эффективной при работе с случайно упорядоченными данными или данными, содержащими большое количество повторяющихся значений.

3) Дополнительные затраты на перемещение элементов: Шейкерная сортировка требует обмена элементов путем их последовательного сравнения и обмена. Это может привести к дополнительным затратам на перемещение элементов в списке.

4) Ограниченный выбор алгоритма: В сравнении с другими сортировками, такими как быстрая сортировка или сортировка слиянием, шейкерная сортировка имеет ограниченные возможности и выбор алгоритмов для различных сценариев сортировки.

В целом, шейкерная сортировка имеет свои преимущества и недостатки, и ее эффективность зависит от размера списка, его упорядоченности и требований к производительности в конкретной задаче сортировки.

1.3 Типичные сценарии применения

Шейкерная сортировка может быть полезной в следующих типичных сценариях:

Частично отсортированные списки: Если вы имеете дело с частично отсортированными списками, где большая часть элементов уже находится близко к своим конечным позициям, шейкерная сортировка может быть эффективной. Она может быстро обнаружить уже отсортированную часть списка и сосредоточиться только на оставшихся элементах.

Малые массивы данных: Шейкерная сортировка может быть применена к небольшим массивам данных, где требуется простота реализации, а производительность не является основным фактором. В таких случаях преимущество простоты и понятности алгоритма может перевешивать его неоптимальную производительность.

Образцовые данные: Если у вас есть образцовые данные, в которых известен порядок элементов, шейкерная сортировка может быть применена для проверки правильности сортировки других алгоритмов. Вы можете использовать шейкерную сортировку для сравнения результатов с другими алгоритмами и убедиться, что они работают правильно.

Описание программы

В программе для шейкерной сортировки подключены следующие заголовочные файлы: <stdio.h> – стандартный заголовочный файл ввода – вывода; <stdlib.h> - заголовочный файл, который содержит в себе функции, занимающиеся выделением памяти, контролем процесса выполнения программы, преобразованием типов в другие; <string.h> - заголовочный файл, содержащий функции для работы со строками, оканчивающимися на 0, и различными функциями работы с памятью; <locale.h> - заголовочный файл, который используется для задач, связанных с локализацией; <time.h> - заголовочный файл, содержащий типы функции для работы с датой и временем.

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <locale.h>
#include <time.h>
```

Два макроса отвечающих за количество сортируемых элементов и длину строки.

```
#define a 10000
#define b 100000
```

Далее подключаем русский язык и функции расчета времени, после выполнения работы алгоритма выводим размер массива и затраченное время на выполнение программы.

```
setlocale(LC_ALL, "RUS");
clock_t start_time, end_time;
double cpu_time_used;
start_time = clock();

end_time = clock();

cpu_time_used = ((double)(end_time - start_time)) /
CLOCKS_PER_SEC;
printf("Размер массива = %d\n", a);
printf("Время выполнения программы: %f секунд\n",
cpu_time_used);
```

После мы создаём метку, на которую переходим при наличии ошибки в записи на указанный файл, далее считываем отсортированный массив из файла.

label:

```
printf("Введите путь или имя файла с расширением: ");
scanf("%s", filename);

start_time = clock();

if ((file = fopen(filename, "r")) == NULL) {
    printf("ошибка при открытии файла %s\n", filename);
    goto label;
}

fgets(string, b, file);

printf("\t# Исходный массив #\n%s", string);

    fclose(file);

    file = fopen("result.txt", "w");
    n++;
    fprintf(file, "%d", array[0]);
    for (int j = 1; j < n; j++) {
        fprintf(file, ", %d", array[j]);
    }
    fclose(file);

printf("\t# Отсортированный массив #\n");
file = fopen("result.txt", "r");

fgets(string, b, file);
printf("%s", string);

fclose(file);
```

Алгоритм шейкерной сортировки описан следующим образом:

```
while (1) {
    while (string[i] != ',') {
        item[k] = string[i];
        k++;
        i++;
        if (string[i] == '\n') {break;}
    }
    array[n] = atoi(item);
    if (string[i] == '\n') break;
    n++;
    i++;
}
```

```

        k = 0;
        memset(item, 0, 8);
    }
    int right = n;
    while(left < right) {
        for(int j = left; j < right; j++) {
            if(array[j] > array[j + 1]) {
                tmp = array[j];
                array[j] = array[j + 1];
                array[j + 1] = tmp;
                last = j;
            }
        }
        right = last;

        for(int j = right - 1; j >= left; j--) {
            if(array[j] > array[j + 1]) {
                tmp = array[j];
                array[j] = array[j + 1];
                array[j + 1] = tmp;
                last = j;
            }
        }
        left = last++;
    }
}

```

Схемы программы

4.2 Блок-схема алгоритма

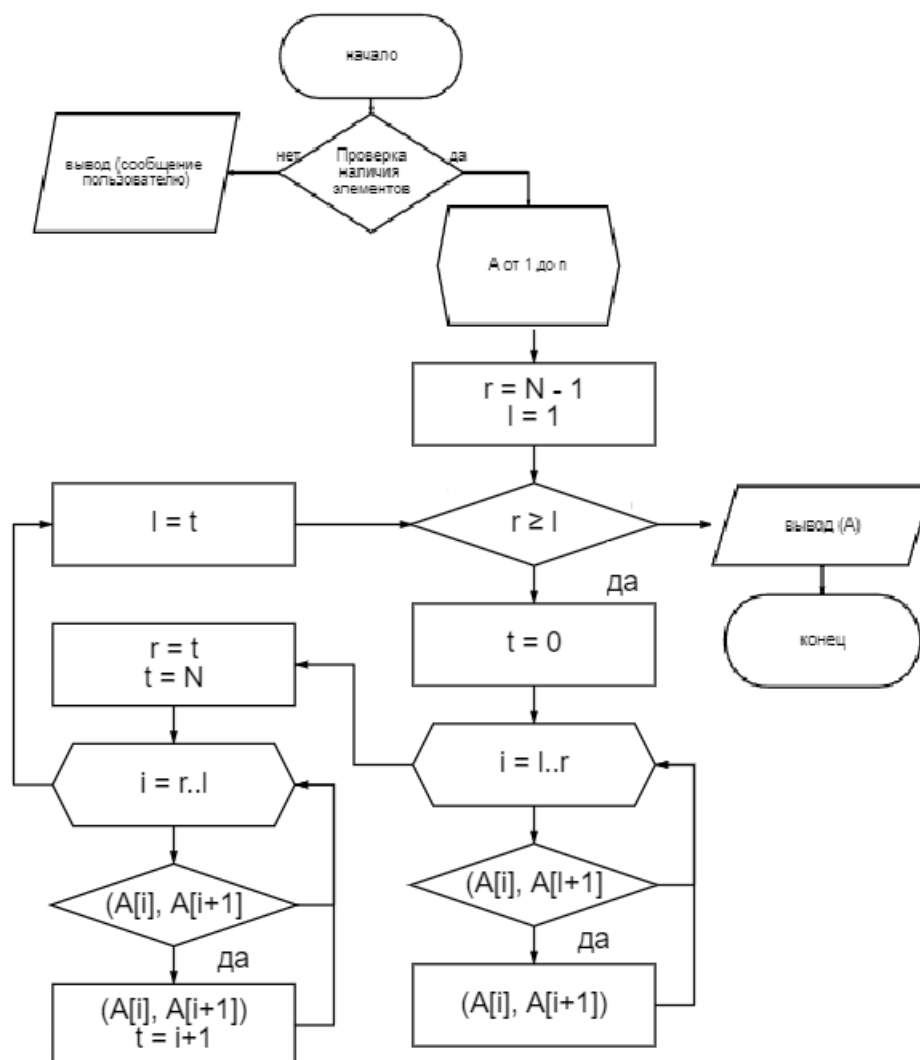


Рисунок 1 – Блок-схема алгоритма

Тестирование программы

5.1 Тестирование на разных наборах данных готовой программы

Тестовый набор данных представлен в таблице 1. Результаты тестирования приведены в Приложении А на рисунках А.1 - А.10 и проанализированы в диаграмме Рисунок 2.

Таблица 1 – Тестовый набор данных

№ теста	Размер массива size	Время выполнения сортировки в секундах
1	100	0.025
2	200	0.043
3	300	0.065
4	400	0.095
5	500	0.130
6	600	0.139
7	700	0.181
8	800	0.241
9	900	0.251
10	1000	0.254

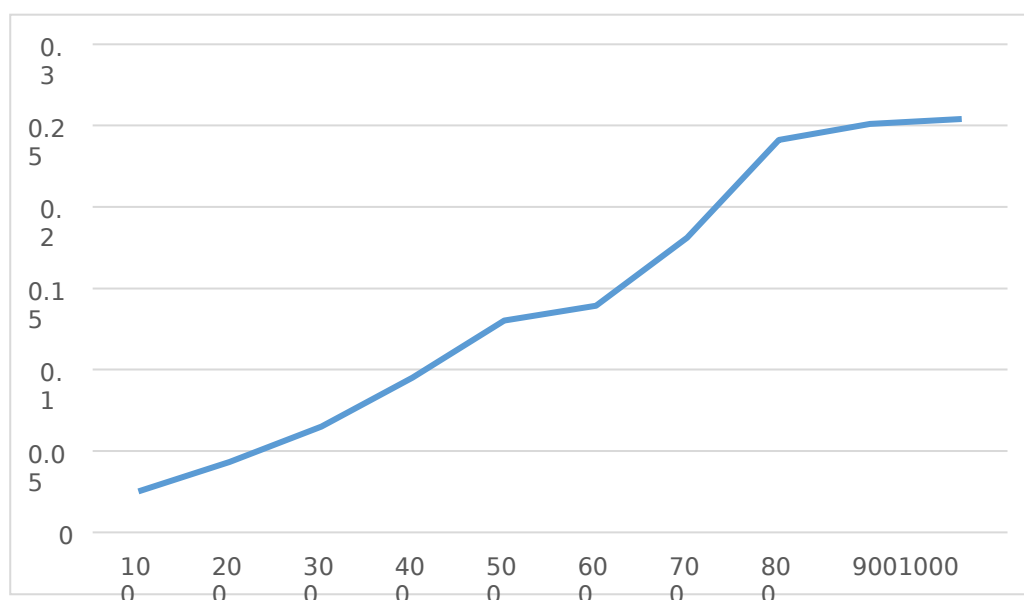


Рисунок 2 – Результаты тестирования

5.2 Анализ полученных результатов тестирования

Алгоритм шейкерной сортировки имеет следующую производительность и число базовых операций в различных сценариях:

Количество базовых операций (перестановок):

1. В наилучшем случае, когда список уже отсортирован, шейкерная сортировка выполнит только один проход по списку без каких-либо перестановок. Количество базовых операций будет равно числу элементов в списке.
2. В наихудшем случае, когда список отсортирован в обратном порядке, требуется полный проход в обе стороны и несколько итераций для перемещения каждого элемента на свое место. Количество базовых операций будет равно двукратному квадрату числа элементов в списке.
3. В случае случайного набора данных, количество базовых операций будет зависеть от конкретного расположения элементов в списке. В среднем, оно также будет близко к двукратному квадрату числа элементов.

Производительность в секундах:

1. Время выполнения шейкерной сортировки зависит от реализации алгоритма и характеристик компьютера, на котором он выполняется. В лучшем случае, когда список уже отсортирован, шейкерная сортировка может быть очень быстрой и зависит преимущественно от количества элементов в списке. Время выполнения для уже отсортированного списка будет разумно малым.
2. В наихудшем случае, когда список отсортирован в обратном порядке, шейкерная сортировка будет иметь более длительное

время выполнения. Оно также зависит от числа элементов в списке, но будет значительно больше, чем в наилучшем случае.

3. Для случайных наборов данных время выполнения шейкерной сортировки будет находиться где-то между наилучшим и наихудшим случаями, в зависимости от степени неупорядоченности данных.

Отладка

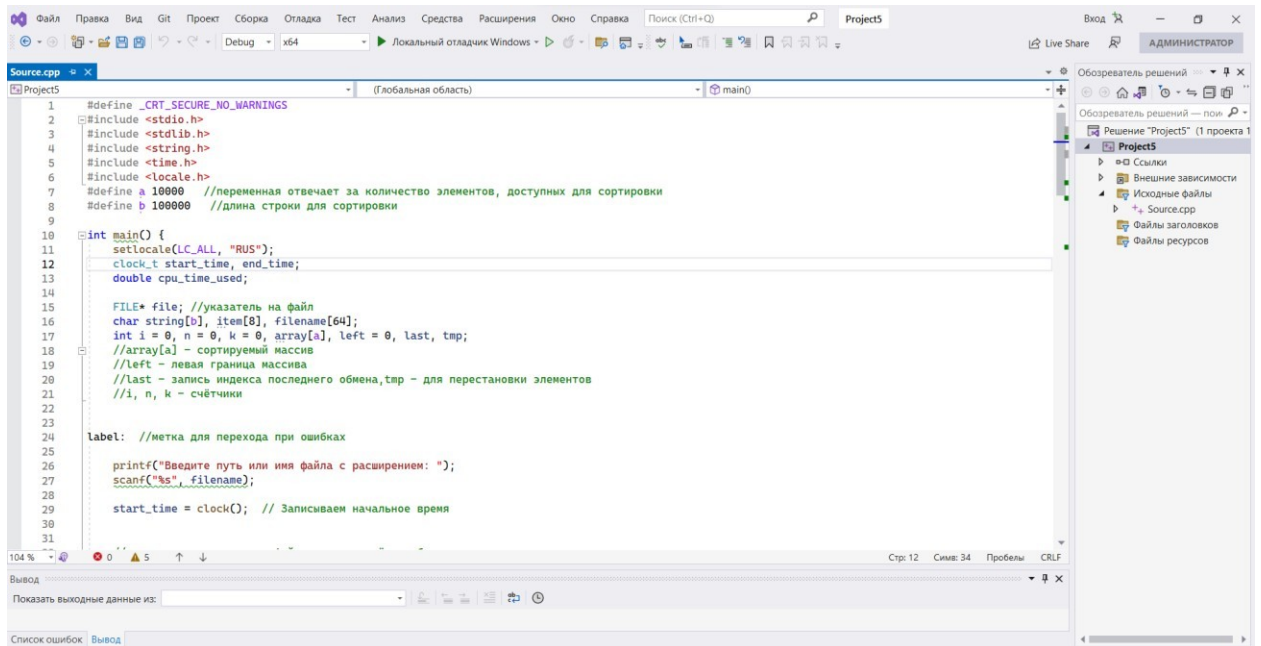


Рисунок 3 – Окно кода

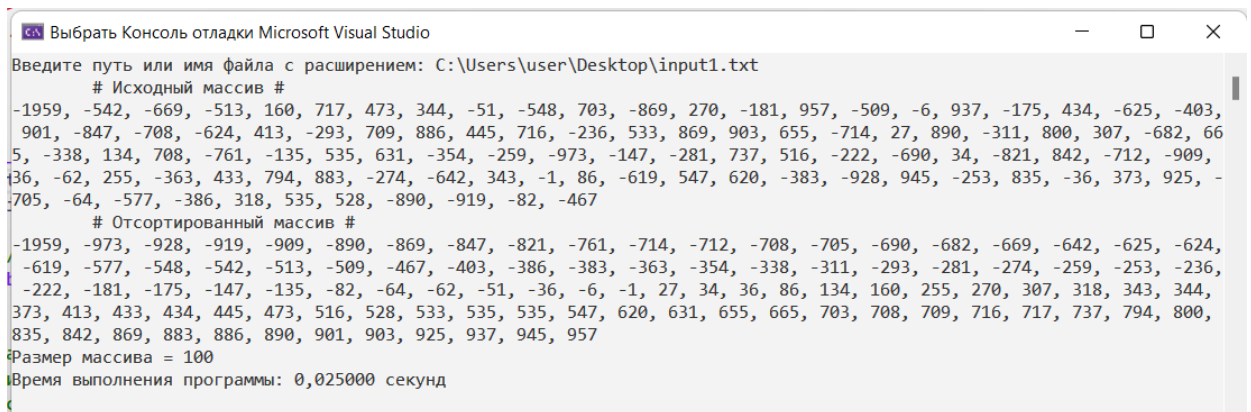


Рисунок 4 – Консоль отладки

Совместная работа

Commits

main

Commits on Jul 5, 2023

- Final release 1.2**
KulakhmetovS committed now
Verified f9682c3 <>
- otshot**
KulakhmetovS committed 1 hour ago
Verified 61a3e60 <>

Commits on Jul 4, 2023

- Final release 1.1**
KulakhmetovS committed yesterday
Verified 154051a <>

Commits on Jul 2, 2023

- Final release 1.0**
KulakhmetovS committed 3 days ago
Verified cbc2e9 <>

Commits on Jul 1, 2023

- File fo MS Visual Studio**
KulakhmetovS committed 4 days ago
Verified d0c918e <>
- Final**
KulakhmetovS committed 4 days ago
Verified 5f7e815 <>
- sort algorithn**
KulakhmetovS committed 4 days ago
Verified 0425ff6 <>

Рисунок 5 – Совместная работа Github

Commits

rargerda-patch...

Commits on Jul 5, 2023

- Add files via upload**
rargerda committed 10 hours ago
Verified 9e4dcf2 <>

Commits on Jul 4, 2023

- Final release 1.1**
KulakhmetovS committed yesterday
Verified 154051a <>

Commits on Jul 2, 2023

- Final release 1.0**
KulakhmetovS committed 3 days ago
Verified cbc2e9 <>


Commits on Jul 1, 2023


- File fo MS Visual Studio**
KulakhmetovS committed 4 days ago
Verified d0c918e <>
- Final**
KulakhmetovS committed 4 days ago
Verified 5f7e815 <>
- sort algorithn**
KulakhmetovS committed 4 days ago
Verified 0425ff6 <>

Newer Older

Рисунок 6 – Совместная работа Github

Commits

 rargerda ▾

 Commits on Jul 1, 2023



Add files via upload  rargerda committed 4 days ago	Verified 67f3bd5 <>
sort algorithm  KulakhmetovS committed 4 days ago	Verified 0425ff6 <>

Рисунок 7 – Совместная работа Github

Заключение

В результате анализа данных, полученных в результате тестирования алгоритма шейкерной сортировки, можно сделать вывод, что время, затраченное на работу программы относительно количества элементов увеличивается линейно, то есть с увеличением количества элементов пропорционально увеличивается время работы программы.

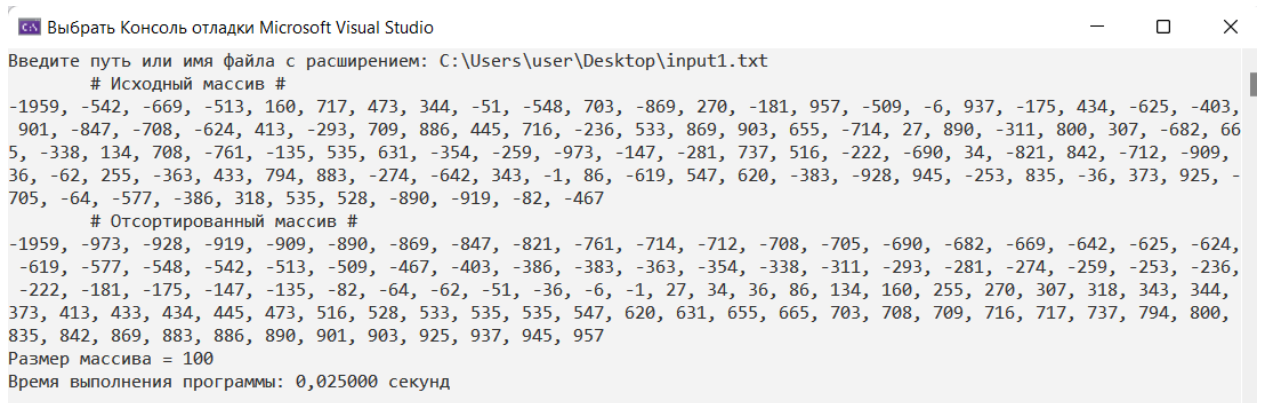
Таким образом, в процессе создания данного проекта разработана программа, реализующая алгоритм шейкерной сортировки.

При выполнении данной работы были получены навыки разработки программ в среде разработки Microsoft Visual Studio и углублены знания языка программирования Си.

Список используемой литературы

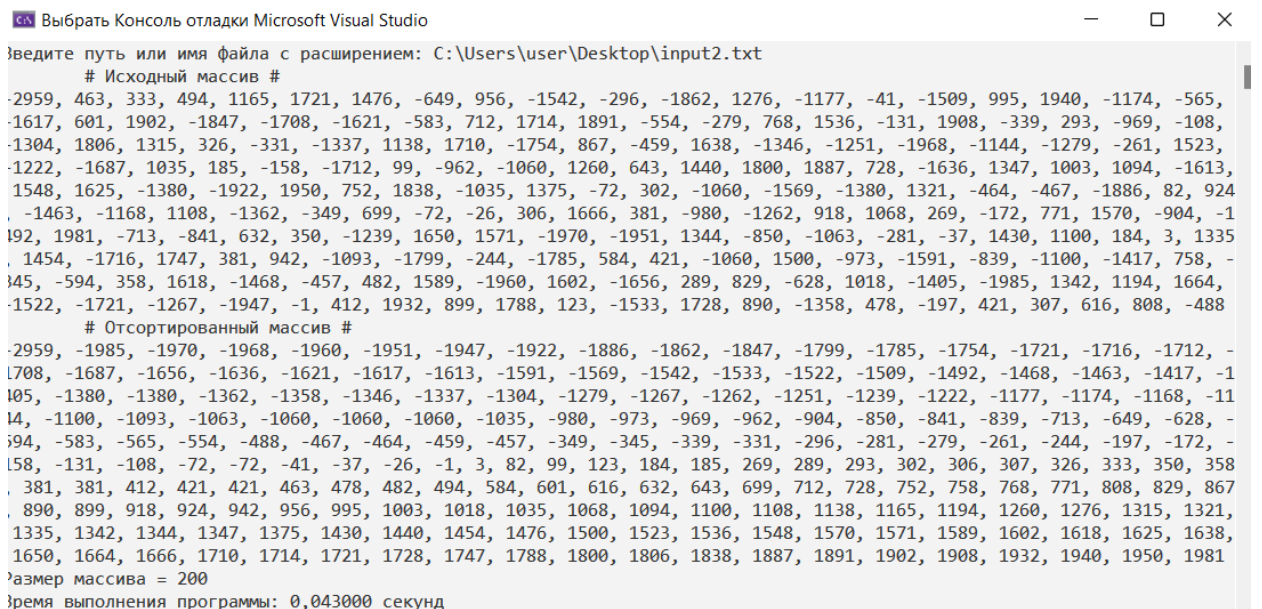
1. Мельников Б. Ф. Алгоритмы – М.: БХВ, 2003. -192с.
2. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: Построение и анализ - М.: МЦНМО, 2001. - 960 с.
3. Никлаус Вирт. Алгоритмы и структуры данных – М.:ДМК-Пресс,2016. -272с.
4. Харви Дейтел, Пол Дейтел. Как программировать на C/C++. М.: Бином, 2022г. – 1002с.

Приложение А



```
Выбрать Консоль отладки Microsoft Visual Studio
Введите путь или имя файла с расширением: C:\Users\user\Desktop\input1.txt
# Исходный массив #
-1959, -542, -669, -513, 160, 717, 473, 344, -51, -548, 703, -869, 270, -181, 957, -509, -6, 937, -175, 434, -625, -403,
901, -847, -708, -624, 413, -293, 709, 886, 445, 716, -236, 533, 869, 903, 655, -714, 27, 890, -311, 800, 307, -682, 66
5, -338, 134, 708, -761, -135, 535, 631, -354, -259, -973, -147, -281, 737, 516, -222, -690, 34, -821, 842, -712, -909,
36, -62, 255, -363, 433, 794, 883, -274, -642, 343, -1, 86, -619, 547, 620, -383, -928, 945, -253, 835, -36, 373, 925, -
705, -64, -577, -386, 318, 535, 528, -890, -919, -82, -467
# Отсортированный массив #
-1959, -973, -928, -919, -909, -890, -869, -847, -821, -761, -714, -712, -708, -705, -690, -682, -669, -642, -625, -624,
-619, -577, -548, -542, -513, -509, -467, -403, -386, -383, -363, -354, -338, -311, -293, -281, -274, -259, -253, -236,
-222, -181, -175, -147, -135, -82, -64, -62, -51, -36, -6, -1, 27, 34, 36, 86, 134, 160, 255, 270, 307, 318, 343, 344,
373, 413, 433, 434, 445, 473, 516, 528, 533, 535, 535, 547, 620, 631, 655, 665, 703, 708, 709, 716, 717, 737, 794, 800,
835, 842, 869, 883, 886, 890, 901, 903, 925, 937, 945, 957
Размер массива = 100
Время выполнения программы: 0,025000 секунд
```

Рисунок А.1 – Результаты тестирования



```
Выбрать Консоль отладки Microsoft Visual Studio
Введите путь или имя файла с расширением: C:\Users\user\Desktop\input2.txt
# Исходный массив #
-2959, 463, 333, 494, 1165, 1721, 1476, -649, 956, -1542, -296, -1862, 1276, -1177, -41, -1509, 995, 1940, -1174, -565,
-1617, 601, 1902, -1847, -1708, -1621, -583, 712, 1714, 1891, -554, -279, 768, 1536, -131, 1908, -339, 293, -969, -108,
-1304, 1806, 1315, 326, -331, -1337, 1138, 1710, -1754, 867, -459, 1638, -1346, -1251, -1968, -1144, -1279, -261, 1523,
-1222, -1687, 1035, 185, -158, -1712, 99, -962, -1060, 1260, 643, 1440, 1800, 1887, 728, -1636, 1347, 1003, 1094, -1613,
1548, 1625, -1380, -1922, 1950, 752, 1838, -1035, 1375, -72, 302, -1060, -1569, -1380, 1321, -464, -467, -1886, 82, 924
-1463, -1168, 1108, -1362, -349, 699, -72, -26, 306, 1666, 381, -980, -1262, 918, 1068, 269, -172, 771, 1570, -904, -1
92, 1981, -713, -841, 632, 350, -1239, 1650, 1571, -1970, -1951, 1344, -850, -1063, -281, -37, 1430, 1100, 184, 3, 1335
1454, -1716, 1747, 381, 942, -1093, -1799, -244, -1785, 584, 421, -1060, 1500, -973, -1591, -839, -1100, -1417, 758, -
45, -594, 358, 1618, -1468, -457, 482, 1589, -1960, 1602, -1656, 289, 829, -628, 1018, -1405, -1985, 1342, 1194, 1664,
-1522, -1721, -1267, -1947, -1, 412, 1932, 899, 1788, 123, -1533, 1728, 890, -1358, 478, -197, 421, 307, 616, 808, -488
# Отсортированный массив #
-2959, -1985, -1970, -1968, -1960, -1951, -1947, -1922, -1886, -1862, -1847, -1799, -1785, -1754, -1721, -1716, -1712, -
1708, -1687, -1656, -1636, -1621, -1617, -1613, -1591, -1569, -1542, -1533, -1522, -1509, -1492, -1468, -1463, -1417, -1
05, -1380, -1380, -1362, -1358, -1346, -1337, -1304, -1279, -1267, -1262, -1251, -1239, -1222, -1177, -1174, -1168, -11
14, -1100, -1093, -1063, -1060, -1060, -1060, -1035, -980, -973, -969, -962, -904, -850, -841, -839, -713, -649, -628, -
94, -583, -565, -554, -488, -467, -464, -459, -457, -349, -345, -339, -331, -296, -281, -279, -261, -244, -197, -172, -
158, -131, -108, -72, -72, -41, -37, -26, -1, 3, 82, 99, 123, 184, 185, 269, 289, 293, 302, 306, 307, 326, 333, 350, 358
381, 381, 412, 421, 421, 463, 478, 482, 494, 584, 601, 616, 632, 643, 699, 712, 728, 752, 758, 768, 771, 808, 829, 867
890, 899, 918, 924, 942, 956, 995, 1003, 1018, 1035, 1068, 1094, 1100, 1108, 1138, 1165, 1194, 1260, 1276, 1315, 1321,
1335, 1342, 1344, 1347, 1375, 1430, 1440, 1454, 1476, 1500, 1523, 1536, 1548, 1570, 1571, 1589, 1602, 1618, 1625, 1638,
1650, 1664, 1666, 1710, 1714, 1721, 1728, 1747, 1788, 1800, 1806, 1838, 1887, 1891, 1902, 1908, 1932, 1940, 1950, 1981
Размер массива = 200
Время выполнения программы: 0,043000 секунд
```

Рисунок А.2 – Результаты тестирования


```
Выбрать Консоль отладки Microsoft Visual Studio
Введите путь или имя файла с расширением: C:\Users\user\Desktop\input3.txt

# Исходный массив #
-3959, -2536, -2667, -504, -1834, 722, 2477, 2354, -42, -2540, 2705, 1141, 2278, 1825, 960, -2509, -5, 2941, 1827, 2436, -614, -398
, 902, -2847, -2708, -2620, 2419, -2287, -1285, -1108, 2447, 723, -231, 2537, -1131, -1091, -1337, -705, 2033, 893, 1699, 2808, -16
83, -2672, 2671, 1664, 139, -1290, 1249, -2133, -1457, 640, -343, -248, -966, -2143, -278, 740, 525, -2222, -2686, 35, 1187, -1158,
-2712, -2899, 39, -59, -1739, 1645, 442, 2802, 888, -2272, -2634, 348, 4, -1904, -2611, 548, -1374, -2379, -2920, -1049, -2247, 28
39, 1966, -1625, -1071, -696, 1942, -566, -2378, 2322, 2537, 535, 1116, -918, 1926, 1539, 1833, -1890, 1639, 2654, 1701, 929, -1025
, -694, -1332, 1383, 2021, 1741, -80, -1931, -2731, 2829, -227, 571, 2097, 1510, 2983, -1712, 160, -2367, 1352, -2237, 2652, 572, 1
031, -2950, 346, -1850, 1939, 721, -1036, 430, -1898, -2814, -2996, 2336, 455, -2715, 749, 1382, -57, -92, -796, 757, -2783, -2415,
-2579, -2058, 502, -1972, 1411, 2164, -2100, -414, -2241, -1345, 2408, -2642, 620, -466, 545, -2518, 591, 1041, 602, -2654, 1290,
-2169, 373, 2019, 1596, -2983, 344, 2196, -1335, -2520, -720, 1734, -2947, -1001, -586, 934, -2101, 788, -2876, -2533, 728, -109, -
2356, 1480, 2805, -579, -692, -2384, 1810, 513, -693, -1385, -2068, 2449, -403, 2249, 1517, -1449, 1795, -2702, -2777, 2007, 2844,
-396, -13, -303, 195, -518, 93, -659, -2482, -1413, 2310, 502, -1553, -1804, -1544, -2383, -423, -1207, -204, 279, -1414, -205, 100
5, 153, -531, 2619, -2465, -2710, -2963, -2825, -2813, 2653, -1043, -2810, -1188, 1885, -1847, 2510, 1200, -366, -2732, -948, -675,
1643, -642, 1886, -2128, 1429, 2865, -861, 2841, -1584, 878, -1007, 1321, -2352, 1020, 2699, 557, 1472, 888, -2615, 2075, 1711, -4
00, -490, 90, -135, 2859, -314, -1601, 788, 253, 1421, 2002, 1584, -2822, 1284, 84, -1579, 1613, 2754, 831, -2073, 1169, -846, -128
3, 2187

# Отсортированный массив #
-3959, -2996, -2983, -2963, -2950, -2947, -2920, -2899, -2876, -2847, -2825, -2822, -2814, -2813, -2810, -2783, -2777, -2732, -2731
, -2715, -2712, -2710, -2708, -2702, -2686, -2672, -2667, -2654, -2642, -2634, -2620, -2615, -2611, -2579, -2540, -2536, -2533, -25
20, -2518, -2509, -2482, -2465, -2415, -2384, -2383, -2379, -2378, -2367, -2356, -2352, -2287, -2272, -2247, -2241, -2237, -2222, -
2169, -2143, -2133, -2128, -2101, -2100, -2073, -2068, -2058, -1972, -1931, -1904, -1898, -1890, -1850, -1847, -1834, -1804, -1739,
-1712, -1683, -1625, -1601, -1584, -1579, -1553, -1544, -1457, -1449, -1414, -1413, -1385, -1374, -1345, -1337, -1335, -1332, -129
0, -1285, -1283, -1207, -1188, -1158, -1131, -1108, -1091, -1071, -1049, -1043, -1036, -1025, -1007, -1001, -966, -948, -918, -861,
-846, -796, -720, -705, -696, -694, -693, -692, -675, -659, -642, -614, -586, -579, -566, -531, -518, -504, -490, -466, -423, -414
, -403, -400, -398, -396, -366, -343, -314, -303, -278, -248, -231, -227, -205, -204, -135, -109, -92, -80, -59, -57, -42, -13, -5,
4, 35, 39, 84, 90, 93, 139, 153, 160, 195, 253, 279, 344, 346, 348, 373, 430, 442, 455, 502, 502, 513, 525, 535, 545, 548, 557, 57
1, 572, 591, 602, 620, 640, 721, 722, 723, 728, 740, 749, 757, 788, 788, 831, 878, 888, 888, 893, 902, 929, 934, 960, 1005, 1020, 1
031, 1041, 1116, 1141, 1169, 1187, 1200, 1249, 1284, 1290, 1321, 1352, 1382, 1383, 1411, 1421, 1429, 1472, 1480, 1510, 1517, 1539,
1584, 1596, 1613, 1639, 1643, 1645, 1664, 1699, 1701, 1711, 1734, 1741, 1795, 1810, 1825, 1827, 1833, 1885, 1886, 1926, 1939, 1942,
1966, 2002, 2007, 2019, 2021, 2033, 2075, 2097, 2164, 2187, 2196, 2249, 2278, 2310, 2322, 2336, 2354, 2408, 2419, 2436, 2447, 2449
, 2477, 2510, 2537, 2537, 2619, 2652, 2653, 2654, 2671, 2699, 2705, 2754, 2802, 2805, 2808, 2829, 2839, 2841, 2844, 2859, 2865, 294
1, 2983
Размер массива = 300
Время выполнения программы: 0,065000 секунд
```

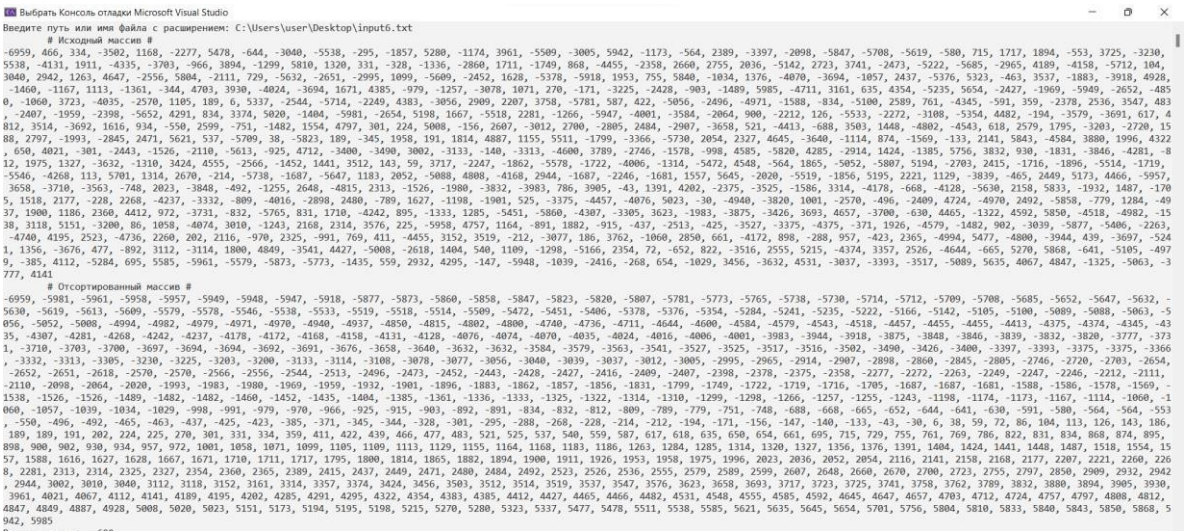
Рисунок А.3 – Результаты тестирования

```
Выбрать Консоль отладки Microsoft Visual Studio
Введите путь или имя файла с расширением: C:\Users\user\Desktop\input4.txt

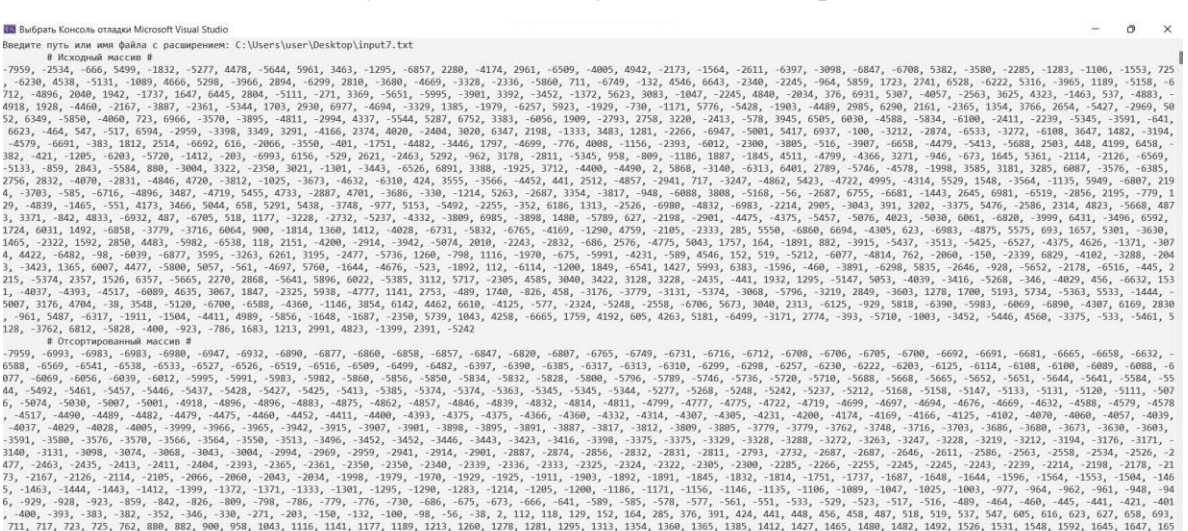
# Исходный массив #
-4959, -1535, 2334, -1503, -833, 3723, -523, 1355, -1041, -3539, 1705, 142, 3279, -3175, -2040, -3509, -1005, -59, 827, 1436, -3613, 2603, -98, -3847, -3708, 381, -2581, -12
86, -284, -107, 1447, 1724, 2770, -463, -2131, -90, -2336, -1704, -2967, -2107, 700, 3809, 3319, 2330, -2329, 664, 3140, 3711, 250, 2868, -2456, -359, -3342, -3247, 35, 858,
-3278, -2260, -474, -3222, 315, -965, 2188, -2158, -3712, 2103, -2961, -3059, -738, 2646, -557, 3803, 3889, 2729, -3633, 3349, 3005, 3098, -3610, -452, -373, 622, -3919, -4
8, -1246, -161, 966, 3376, 1930, -1695, -3058, -3565, -3377, -678, 1537, 1536, -3884, -1918, 2927, -3461, 833, 3112, 639, 1655, 2702, -2071, 1976, -1694, 3670, 2384, 1021, 7
42, -1079, -930, 2270, 1829, -1226, 3572, 1097, -3490, 3984, 1289, -2840, -1366, 2353, -3236, 3653, 3573, 31, 51, -653, -2850, -3061, 1722, 1965, -570, 3104, 2188, -1995, -6
64, 3456, 286, -250, -1618, 2944, -3092, -3795, -2243, -3782, -1414, 2422, -3057, -497, 1029, -3589, 1165, -3100, -3413, -1240, -2345, -2592, 2359, -379, 535, 1546, 2483, -4
08, 41, -398, -3653, -1710, 2833, -2627, -981, 596, -3982, -655, 3197, -334, -3519, -3720, 734, -3947, -2001, -1585, -65, 2900, -212, -1875, -3533, -272, 2892, -3355, 2481,
-2195, -1579, 2309, 2617, 2811, -2487, 2308, 3616, -1067, -2551, 598, 1249, -3483, 3553, 2796, 2300, 2224, -993, 1844, -3395, 2988, -3302, -805, 483, -907, 2342, 2520, -2413
, 1311, -2498, 3448, -2803, 1457, 2618, 578, -206, 2797, 3280, -413, 796, 6, -846, 470, 3620, -1464, 291, 2038, -3824, -1812, 1654, 3958, 2191, -187, 2886, -846, -490, -3800
, -1366, -3731, 53, 326, 2644, -1641, 886, -1127, 430, 1866, 140, 3842, -2584, 1879, 3995, -1679, -1351, -1980, 1699, -443, 473, -111, -3614, 1075, -1289, -1400, -1490, 1001
, -1134, -2141, 2687, 1400, -2212, 3254, -3579, 1002, -1416, -3821, -1716, -915, 3423, 614, 3755, -2169, 2929, 169, -1846, -2282, -2813, -26, 3326, -1632, 689, 1423, -1446,
-566, -3453, 3441, -2489, 2142, -1942, 1716, -247, -3863, 422, -3723, -2007, -3315, 528, 2547, -2565, -136, 948, -3807, 3193, -703, 414, 283, -3897, -3515, -3720, 454, -2269
, -1888, -300, 3313, 669, 1786, 262, 313, -3648, 3182, 51, -3088, -1193, -2168, 943, 313, -247, 318, -444, 3644, -21, -3519, 144, 3194, 220, 3129, -1839, 1535, 448, -828, -1
535, 43, 1657, -1711, -1564, -2749, 22, -1849, 1507, 745, 647, 1185, -3688, 474, 19, -1832, 2017, -1215, -2096, -2044, 3391, -1799, -375, -1526, 414, -2687, -2179, 1331, -21
29, -3631, 157, -168, 67, 3487, 294, 3518, -3824

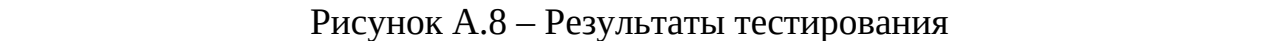
# Отсортированный массив #
-4959, -3982, -3947, -3919, -3897, -3884, -3863, -3847, -3824, -3824, -3821, -3807, -3800, -3795, -3782, -3731, -3723, -3720, -3720, -3712, -3708, -3688, -3653, -3648, -3633
, -3631, -3614, -3613, -3610, -3589, -3579, -3565, -3539, -3533, -3519, -3519, -3515, -3509, -3490, -3483, -3461, -3453, -3413, -3395, -3377, -3355, -3342, -3315, -3302, -32
78, -3247, -3236, -3222, -3175, -3100, -3092, -3088, -3061, -3059, -3058, -3057, -2967, -2961, -2850, -2840, -2813, -2803, -2749, -2687, -2627, -2592, -2584, -2581, -2565, -
2551, -2498, -2489, -2487, -2456, -2413, -2345, -2336, -2329, -2282, -2269, -2260, -2243, -2212, -2195, -2179, -2169, -2168, -2158, -2141, -2131, -2129, -2107, -2096, -2071,
-2044, -2040, -2007, -2001, -1995, -1980, -1942, -1918, -1888, -1875, -1849, -1846, -1839, -1832, -1812, -1799, -1716, -1711, -1710, -1704, -1695, -1694, -1679, -1641, -163
2, -1618, -1585, -1579, -1564, -1535, -1535, -1526, -1503, -1490, -1464, -1446, -1416, -1414, -1400, -1366, -1366, -1351, -1289, -1286, -1246, -1240, -1226, -1215, -1193, -1
134, -1127, -1079, -1067, -1041, -1005, -993, -981, -965, -930, -915, -907, -846, -846, -833, -828, -805, -738, -703, -678, -664, -655, -653, -570, -566, -557, -523, -497, -
490, -474, -463, -452, -444, -443, -413, -408, -398, -379, -375, -373, -359, -334, -300, -284, -272, -250, -247, -247, -212, -206, -187, -168, -161, -136, -111, -107, -98, -
90, -65, -59, -48, -26, -21, 6, 19, 22, 31, 35, 41, 43, 51, 51, 53, 67, 140, 142, 144, 157, 169, 220, 250, 262, 283, 286, 291, 294, 313, 313, 315, 318, 326, 381, 414, 414, 4
22, 430, 448, 454, 470, 473, 474, 483, 528, 535, 578, 596, 598, 614, 622, 639, 647, 664, 669, 689, 700, 734, 742, 745, 796, 827, 833, 858, 886, 943, 948, 966, 1001, 1002, 10
21, 1029, 1075, 1097, 1165, 1185, 1249, 1289, 1311, 1331, 1355, 1400, 1423, 1436, 1447, 1457, 1507, 1535, 1536, 1537, 1546, 1654, 1655, 1657, 1699, 1705, 1716, 1722, 1724, 1
786, 1829, 1844, 1866, 1879, 1930, 1965, 1976, 2017, 2038, 2103, 2142, 2188, 2188, 2191, 2224, 2270, 2300, 2308, 2309, 2330, 2334, 2342, 2353, 2359, 2384, 2422, 2481, 2483,
2520, 2547, 2603, 2617, 2618, 2644, 2646, 2687, 2702, 2729, 2770, 2796, 2797, 2811, 2833, 2868, 2886, 2892, 2900, 2927, 2929, 2944, 2988, 3005, 3098, 3104, 3112, 3129, 3140,
3182, 3193, 3194, 3197, 3254, 3279, 3280, 3313, 3319, 3326, 3349, 3376, 3391, 3423, 3441, 3448, 3456, 3487, 3518, 3553, 3572, 3573, 3616, 3620, 3644, 3653, 3670, 3711, 3723
, 3755, 3803, 3809, 3842, 3889, 3958, 3984, 3995
Размер массива = 400
Время выполнения программы: 0,095000 секунд
```

Рисунок А.4 – Результаты тестирования



```
er\Desktop\input6.txt
```





Выбрать Консоль отладки Microsoft Visual Studio

Рисунок А.9 – Результаты тестирования

Введите путь или имя файла с расширением: C:\Users\user\Desktop\input10.txt

Исходный массив

```
-19959, 8467, -3666, -3581, 9169, 5724, 1478, -643, -3039, -5537, -4295, -1856, -6720, 6827, -39, -9509, -7805, 1942, -5173, -4564, 2390, 4604, -6098, -9847, -9708, 2382, 7421, 8716, 9718, 9895, -4553, -8275, 47
71, 1538, -8131, 9912, -4334, -3702, 7035, -106, -1298, -6190, 1321, 332, 7673, -5336, 5141, -2289, -1748, -3132, -4454, -2357, 2661, 2756, -9964, 2859, -1277, -250, -2472, -9222, 2316, 6965, -7811, -8158, -971
2, 185, -960, -1058, 9264, -7353, -2555, -6196, 5890, -3271, -5631, 5350, 5006, 1108, -5688, -6452, 9629, 2623, -5917, 9954, 8756, 1840, -5034, -2624, 3931, -3693, 6944, 2438, -5375, 1323, -4463, -8463, 6118, -7
918, -7072, 6541, -5167, 1114, -5361, -343, -7297, 70, 3977, -7694, 1673, -7615, -4979, -1356, -3077, 9073, -3730, -4171, -3224, 9573, -4903, 6512, -6815, 3290, -839, 8036, -7646, -5234, -6346, -5574, -5909, 205
2, -2651, -8850, 6941, -8277, 3966, -6570, 1106, 190, 8007, 1337, 5457, 2287, -2248, 383, 4945, -1091, 2208, -242, -5780, 8588, -3578, -5055, -2495, 3030, 6413, -831, -9100, 2590, 8762, -8345, 7410, -3641, -2377
, -9464, -8453, -3517, -2406, -5959, -6398, -5651, 291, 835, -626, 1020, -5404, -5980, -2653, -6802, 9668, -5517, -1719, -5266, -9947, -8001, -3583, -2063, -3100, -6212, 8127, -9533, -6272, 4893, -5353, -7518, 7
807, -7579, 4310, -3383, -7188, -486, 4309, -2384, 8935, 7451, -9401, -4751, 6519, 1555, -7203, 302, -3776, 1008, -4156, 2608, 4989, 2701, -6805, -9516, -6907, 4343, 522, -8413, -687, -497, -2552, -4801, 3458, -7
3382, -9421, 9796, 4798, 5281, 9589, -9283, -1992, -2844, -9529, -6379, 8538, 2292, -3962, -5822, 8190, -344, -2042, -3809, 9815, -7113, 9156, 1511, 6202, -7366, -5729, -9946, -9673, -7355, -3639, -5114, 8875, -7
1568, -132, -9859, -6157, 8584, -8120, 1997, 322, 8651, 21, -4301, -6643, -1525, -2109, -5612, -4925, 712, -7400, -7490, -8998, -3132, 7861, 4688, 3401, -211, 5255, 6423, -4998, 585, -5819, 285, -2913, 1425, -1
384, -6244, -168, 9315, -5831, -7846, -4280, 7189, 9976, 1328, -7632, -1309, -8576, 555, -6566, 6549, -2559, -488, 144, 8060, -8283, -6247, 6139, 2423, 6279, -4805, 6687, 2529, -7452, 7437, 9866, 2949, -9807, -68
06, -6783, -9585, -1715, 6105, -5513, 6282, 2455, -4267, 8114, 1701, 1315, -9330, -4214, 2263, -5687, -5646, 1184, -9948, -9088, 808, -8168, -9056, -5687, -2245, -1680, 9558, -6355, -2019, -9519, -5856, -6805, -7
9779, -2871, -7839, -4465, -9551, 1173, 466, 2044, -8342, -3709, -3562, 7253, -3767, -847, -491, -5255, -9352, 3186, -1687, -5526, -1979, -7832, 4018, 8787, -95, 7958, -2609, 202, -6375, -3524, -5586, -686, -41
77, -667, -4127, -5629, -9842, 1833, -1931, -2513, -1704, -2482, -1823, 7773, 2269, -8237, -7332, 7192, 3985, -6898, -1520, -788, -2373, -5198, -5901, 526, -7375, -8457, -8076, 1023, -29, 3061, 4181, 1002, -2569
, 7505, -2408, -7276, 3031, -1508, -9858, 7222, 1285, 3064, -2100, 9187, -1640, -7580, 973, 4270, -531, -9765, 832, 9711, -4241, 8096, -5333, -2715, 2550, -9066, 3094, -7305, -8377, -1182, -7875, -3425, -8307, -7
7343, -3699, 7371, -7535, -5322, -7408, -6150, -4517, -8982, -1537, -8882, -6849, -7200, 8087, 1859, -8074, -990, -5243, 2169, -9686, -424, 226, 2043, -7243, -2836, 4891, -2118, 7086, -436, -6513, -424, 4474, -7
7375, -4374, -4371, 1927, -4578, -1481, -3898, 4962, -9877, -5405, -6263, 3261, 195, 2524, -8736, -1740, -3798, -1884, -4970, -9675, -990, 770, -3589, -4454, -8848, -8481, -211, 4924, 187, -8238, -5060, -9150, 8
662, 3829, 899, 7713, 8958, 7578, -1635, 3007, 1477, -8800, -3943, -3561, -7697, 2760, 9357, -7676, -3523, -4892, -8888, 4887, 9801, -7151, 4460, -7573, 2993, -2617, 9405, -3460, 1110, -1297, 2835, 2355, -3928,
-651, 8823, 4485, -9445, -6785, -8374, -643, -1474, 3357, -664, -6730, -6132, -640, 2896, 3022, -384, 112, 2717, 8696, 1585, -5960, -5578, -5872, -5772, -5435, -3441, -1068, -7705, -146, 2053, 6962, -6416, -267,
-3346, 6972, -8544, 4369, -7469, -7037, -7393, -7517, -9089, 1635, 67, -7153, -5325, 2938, -7777, 7859, -6247, -3489, -7690, -9826, -8542, 5285, 1782, -6151, 7398, -7722, 7390,
2193, 2734, -8363, -3467, -4444, -8007, 176, -4296, -3038, 548, 5881, -9700, 4413, 6641, 9855, -5146, 3142, 1462, -2390, 876, -9577, 2677, -8248, 8443, 1705, 2673, 40, -687, -9125, -9929, 2818, -9390, -8083, 4
932, -1889, 694, 3169, -6170, -9961, -3513, -1316, 9090, 9497, -7411, -4011, 5145, 9353, 9314, 8651, -3261, -7957, 1258, -9665, -1241, 1192, -2395, -4737, 2181, -1498, -6171, -6226, -9393, -709, -4003, 7540, -44
5, -4440, 1626, -3533, -460, -3872, 1239, -2188, -827, -9400, -3923, -9786, -1317, -1787, -6009, -4177, -4399, -6609, 5759, -7330, -3573, -1974, -5916, 75, 8786, 5498, -5031, -3713, -6154, 2603, -9497, -8780, -7
338, -4259, -7637, -990, -7830, -2512, 8240, 2164, -4459, -2381, -9088, -2489, -3296, 1817, -768, -9250, -4796, -5025, -8461, -9697, 1422, -8903, 1247, 3584, 3648, -7029, 7864, -7088, 1075, -8456, -1289, 7546, 8
678, -8231, 5262, -1481, 3985, -1712, 5944, -7135, 8540, -6756, -4493, -1683, -2131, -399, -1678, -8869, -5529, -2849, -4914, -1431, -238, -100, 7103, 4423, -6473, 1600, -3032, 4015, -4445, -9972, -8458, -4654, 8
-7912, 7057, 2637, -7592, -3338, -4931, -5319, -8412, 1142, -9392, 2099, -8780, -8142, -47, -8113, 4146, -9310, -2051, 2843, -8571, -4381, -9252, -2934, -5464, -9218, 8035, 2225, 5189, -2962, -147, -4372, 1124,
5748, 9923, -6641, 2256, -5235, -5056, 4955, -6683, 2725, -4590, -8976, -9646, 1000, -7452, -504, 8584, -485, 7964, -6659, -1925, 7913, 6142, 1195, -8053, -4929, -9575, 4606, -3828, -5572, 2403, -3295, -9375, -7
189, 9375, 92, 6565, 6036, 4736, -860, 813, -4006, -1744, -3348, -6065, 837, -9519, -8645, -8898, 8230, 7841, 4625, -7889, 2636, -5814, 9690, -8350, -4338, -8367, 893, 353, -8585, 3452, 4008, -7298, -7768
, -4546, 6303, 6634, -3698, 4256, -9852, 1124, 2317, -5787, -2892, -5973, -801, -8921, -8633, 6858, -5951, -5846, 1360, 5264, 1903, -6324, -358, -3092, 4902, -6439, -1512, -5053, -8718, 3653, 673, -7780, -4598,
-3077, -6109, 9369, -6122, -9742, 9008, -7382, -6030, 2, -8056, -219, -3497, 2392, 2684, -4688, -3302, -4411, 2722, -4062, 9037, -3590, 1460, -3766, 2508, -39, -6041, -3507, -8485, -4732, -5064, -1132, -9942, 47
00, 3971, -3737, 5117, 6215, -5466, -2185, 8330, -6961, 211, 713, 1019, -8046, 6085, -9291, -5317, -5227, -1620, 180, -4050, -3459, 8115, -8321, 7110, -4103, -6928, -9212, -6024, 8132, -45, -1312, -3888, 8, 2
941, 5790, -8277, -8638, -9972, -4817, -5223, -2800, -4029, -8115, -8027, -8929, 1333, -7134, -3848, 4295, 2167, -9176, -324, 5629, -1351, -7402, -6691, -5307, -5314, 79, 116, 2249
```

Отсортированный массив

```
-19959, -9677, -9972, -9872, -9964, -9961, -9948, -9947, -9946, -9942, -9929, -9877, -9860, -9859, -9858, -9852, -9847, -9842, -9826, -9807, -9786, -9779, -9765, -9742, -9712, -9708, -9700, -9697, -9686, -9675,
-9673, -9665, -9646, -9585, -9577, -9575, -9551, -9533, -9529, -9519, -9519, -9516, -9509, -9407, -9464, -9445, -9421, -9401, -9400, -9393, -9392, -9390, -9375, -9352, -9330, -9310, -9291, -9252, -9250, -9225, -9218, -9212, -9203, -9176, -9150, -9125, -9113, -9100, -9089, -9088, -9088, -9056, -8998, -8986, -8986, -8986, -8869, -8869, -8850, -8848, -8800, -8780, -8780, -8736, -8
718, -8683, -8645, -8638, -8585, -8584, -8576, -8571, -8544, -8542, -8485, -8481, -8463, -8461, -8458, -8457, -8456, -8453, -8413, -8412, -8377, -8374, -8374, -8367, -8363, -8350, -8345, -8342, -8321, -8307, -82
83, -8277, -8277, -8275, -8248, -8242, -8238, -8237, -8231, -8169, -8158, -8131, -8120, -8115, -8076, -8074, -8056, -8053, -8046, -8027, -8007, -8001, -7989, -7957, -7918, -7912, -7875, -7859, -7846, -7839, -783
2, -7830, -7811, -7780, -7777, -7772, -7705, -7697, -7694, -7676, -7646, -7637, -7632, -7615, -7592, -7588, -7579, -7573, -7518, -7517, -7490, -7469, -7452, -7452, -7411, -7408, -7400, -7393
, -7382, -7375, -7375, -7366, -7355, -7353, -7343, -7338, -7332, -7330, -7305, -7300, -7297, -7276, -7268, -7243, -7203, -7200, -7188, -7153, -7151, -7135, -7134, -7113, -7088, -7072, -7057, -7037, -7029, -7005,
-6965, -6961, -6928, -6907, -6898, -6849, -6806, -6805, -6805, -6802, -6785, -6779, -6756, -6730, -6720, -6703, -6691, -6683, -6659, -6641, -6609, -6570, -6566, -6513, -6473, -6452, -6443, -6439, -6416, -6398,
-6379, -6375, -6355, -6346, -6324, -6272, -6263, -6247, -6247, -6244, -6226, -6212, -6196, -6190, -6171, -6170, -6169, -6157, -6154, -6151, -6150, -6132, -6122, -6098, -6065, -6041, -6030, -6024, -6015, -6009, -6
5980, -5973, -5969, -5900, -5959, -5951, -5917, -5916, -5901, -5872, -5856, -5840, -5831, -5822, -5819, -5814, -5787, -5780, -5772, -5729, -5687, -5687, -5651, -5648, -5631, -5629, -5612, -5608, -5586, -5578, -5
572, -5537, -5529, -5526, -5517, -5517, -5513, -5464, -5446, -5435, -5405, -5404, -5375, -5361, -5353, -5336, -5333, -5325, -5322, -5319, -5314, -5307, -5266, -5255, -5243, -5235, -5234, -5227, -5223, -5198, -51
73, -5167, -5146, -5114, -5064, -5060, -5056, -5055, -5053, -5034, -5031, -5025, -4998, -4979, -4970, -4951, -4929, -4929, -4925, -4914, -4903, -4892, -4891, -4817, -4801, -4796, -4751, -4737, -4732, -4688, -465
4, -4598, -4590, -4578, -4564, -4553, -4546, -4517, -4493, -4465, -4463, -4459, -4454, -4454, -4444, -4440, -4435, -4411, -4399, -4381, -4374, -4372, -4371, -4338, -4334, -4301, -4296, -4295, -4294, -4280, -4267
, -4241, -4214, -4177, -4177, -4171, -4171, -4156, -4127, -4103, -4062, -4050, -4011, -4006, -4005, -4003, -3962, -3943, -3928, -3923, -3888, -3872, -3848, -3847, -3828, -3809, -3798, -3776, -3766, -3737, -3730, -3713,
-3709, -3702, -3699, -3698, -3693, -3666, -3641, -3639, -3590, -3589, -3583, -3578, -3573, -3562, -3561, -3538, -3533, -3524, -3523, -3517, -3513, -3507, -3497, -3489, -3467, -3460, -3459, -3441, -3425,
-3383, -3382, -3348, -3346, -3302, -3296, -3295, -3271, -3261, -3244, -3132, -3132, -3100, -3098, -3092, -3077, -3077, -3039, -3038, -3032, -2962, -2934, -2913, -2892, -2871, -2849, -2844, -2836, -2800, -2738, -2
2715, -2053, -2051, -2024, -2017, -2009, -2569, -2559, -2555, -2552, -2513, -2512, -2495, -2482, -2472, -2409, -2408, -2406, -2395, -2390, -2384, -2381, -2377, -2373, -2357, -2289, -2248, -2245, -2188, -2185, -2
131, -2118, -2109, -2100, -2063, -2051, -2042, -2019, -1992, -1982, -1979, -1974, -1931, -1925, -1919, -1889, -1884, -1856, -1823, -1787, -1748, -1744, -1740, -1719, -1715, -1712, -1705, -1704, -1687, -1683, -16
80, -1678, -1640, -1635, -1620, -1568, -1537, -1525, -1520, -1512, -1508, -1498, -1481, -1481, -1474, -1431, -1384, -1357, -1317, -1316, -1312, -1309, -1297, -1289, -1272, -1256, -1241, -1132, -1091, -106
8, -1058, -990, -990, -990, -960, -860, -839, -833, -831, -827, -801, -788, -768, -713, -709, -687, -687, -686, -667, -664, -651, -643, -640, -626, -504, -497, -491, -488, -486, -485, -460, -445, -436, -42
4, -424, -399, -384, -358, -344, -343, -324, -267, -259, -242, -238, -219, -211, -211, -189, -186, -168, -147, -146, -132, -106, -100, -95, -70, -47, -45, -39, -39, -29, 2, 8, 21, 40, 67, 75, 79, 92, 105, 112, 1
16, 144, 176, 187, 190, 195, 202, 211, 226, 285, 291, 302, 322, 321, 353, 383, 466, 522, 526, 540, 555, 585, 673, 694, 712, 770, 808, 813, 832, 835, 837, 876, 893, 899, 931, 973, 1000, 1002, 1000, 1020, 1023, 10
59, 1075, 1100, 1106, 1110, 1114, 1124, 1124, 1114, 1192, 1195, 1224, 1239, 1247, 1258, 1285, 1315, 1321, 1323, 1328, 1333, 1337, 1342, 1360, 1422, 1425, 1460, 1462, 1477, 1478, 1511, 1538, 1555, 1585, 1600, 162
6, 1635, 1672, 1701, 1782, 1817, 1833, 1840, 1903, 1927, 1933, 1942, 1997, 2043, 2044, 2052, 2053, 2059, 2164, 2167, 2169, 2181, 2193, 2208, 2225, 2249, 2250, 2263, 2269, 2287, 2292, 2316, 2317, 2355, 2382, 2390
, 2392, 2403, 2423, 2438, 2455, 2508, 2524, 2529, 2550, 2590, 2603, 2608, 2623, 2636, 2637, 2661, 2673, 2677, 2684, 2701, 2717, 2722, 2725, 2734, 2756, 2760, 2818, 2835, 2843, 2859, 2896, 2938, 2941, 2949, 2993,
3007, 3022, 3030, 3031, 3061, 3064, 3142, 3149, 3180, 3186, 3261, 3290, 3357, 3401, 3452, 3458, 3584, 3648, 3653, 3649, 3829, 3931, 3966, 3971, 3977, 3985, 3985, 4008, 4015, 4018, 4146, 4181, 4256, 4270, 4295, 4309,
4310, 4343, 4369, 4413, 4423, 4460, 4474, 4485, 4604, 4606, 4625, 4608, 4700, 4716, 4771, 4790, 4807, 4893, 4902, 4924, 4932, 4945, 4955, 4962, 4989, 5006, 5117, 5141, 5145, 5185, 5205, 5255, 5262, 5264, 5281, 5
350, 5457, 5498, 5573, 5574, 5629, 5724, 5748, 5759, 5790, 5881, 5890, 5944, 6036, 6085, 6145, 6148, 6193, 6192, 6202, 6215, 6279, 6282, 6303, 6413, 6423, 6512, 6519, 6549, 6565, 6634, 6641, 6687, 6827, 68
58, 6941, 6944, 6962, 6972, 7035, 7086, 7103, 7110, 7189, 7192, 7222, 7253, 7371, 7398, 7410, 7421, 7437, 7451, 7505, 7546, 7549, 7578, 7673, 7713, 7773, 7807, 7825, 7841, 7861, 7864, 7870, 7913, 7958, 7964, 800
7, 8035, 8060, 8087, 8114, 8115, 8127, 8132, 8190, 8230, 8240, 8330, 8443, 8467, 8538, 8540, 8584, 8588, 8636, 8651, 8651, 8662, 8678, 8696, 8716, 8756, 8762, 8786, 8787, 8823, 8875, 8896, 8935, 8958, 9008, 9037
, 9072, 9090, 9156, 9169, 9187, 9264, 9314, 9353, 9357, 9369, 9375, 9405, 9497, 9558, 9589, 9629, 9668, 9690, 9711, 9718, 9796, 9801, 9815, 9855, 9866, 9895, 9912, 9923, 9954, 9976
```

Размер массива = 1800

Время выполнения программы: 0,254000 секунд

Рисунок А.10 – Результаты тестирования

Приложение Б

Листинг

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <locale.h>
#define a 10000
#define b 70000

int main() {
    setlocale(LC_ALL, "RUS");
    clock_t start_time, end_time;
    double cpu_time_used;

    FILE* file;
    char string[b], item[8], filename[64];
    int i = 0, n = 0, k = 0, array[a], left = 0, last, tmp;

label: memset(string, 0, b);
    memset(item, 0, 8);
    i = 0;
    n = 0;
    k = 0;

    printf("Введите путь или имя файла с расширением: ");
    scanf("%s", filename);

    start_time = clock();
    if ((file = fopen(filename, "r")) == NULL) {
        printf("ошибка при открытии файла %s\n", filename);
        goto label;
    }

    fgets(string, b, file);
```

```

fclose(file);

while (1) { while (string[i] !=
    ',') {
        item[k] = string[i];
        k++;
        i++;
        if (string[i] == '\n') { break; }
    }

    array[n] = atoi(item);

    if (array[n] == NULL) {
        printf("Данные в файле не корректны!\nИсправьте их
или выберите другой файл!\n");
        goto label;
    }

    if (string[i] == '\n') break;
    n++;
    i++;
    k = 0;
    memset(item, 0, 8);
}

int right = n;

while (left < right) {
    for (int j = left; j < right; j++) {
        if (array[j] > array[j + 1]) {
            tmp = array[j];
            array[j] = array[j + 1];
            array[j + 1] = tmp;
            last = j;
        }
    }
    right = last;
    for (int j = right - 1; j >= left; j--) {
        if (array[j] > array[j + 1]) {
            tmp = array[j];
            array[j] = array[j + 1];

```

```

        array[j + 1] = tmp;
        last = j;
    }
}
left = last++;
}

file = fopen("result.txt", "w");
n++;
fprintf(file, "%d", array[0]);

for (int j = 1; j < n; j++) {
    fprintf(file, ", %d", array[j]);
}

fclose(file);

printf("\t# Исходный массив #\n%s", string);
-----
printf("\t# Оптимизированный массив #\n");
file = fopen("result.txt", "r");

fgets(string, b, file);
printf("%s", string);

fclose(file);

end_time = clock();

cpu_time_used = ((double)(end_time - start_time)) /
CLOCKS_PER_SEC;
printf("\n");
printf("Размер массива = %d\n", n);
printf("Время выполнения программы: %f секунд\n",
cpu_time_used);

return 0;
}

```