

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 ПРАКТИКЕ: ОЗНАКОМИТЕЛЬНОЙ ПРАКТИКЕ
(2024/2025 учебный год)

_____Сияляко Матвей Алексеевич

Направление подготовки (специальность) 09.05.01 «Применение и эксплуатация
автоматизированных систем специального назначения»

Наименование профиля подготовки (специализация) «Эксплуатация вычислительных
машин, комплексов, систем и сетей»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 5 лет

Год обучения 1 семестр 2

Период прохождения практики с 20.06.2025 по 17.07.2025

Кафедра «Вычислительная техника»

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

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

Руководитель практики к.т.н., доцент, Карамышева Н.С.

(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

ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ: ОЗНАКОМИТЕЛЬНОЙ ПРАКТИКЕ

(2024/2025 учебный год)

Юсупов Владислав Ренатович

Направление подготовки (специальность) 09.05.01 «Применение и эксплуатация
автоматизированных систем специального назначения»

Наименование профиля подготовки (специализация) «Эксплуатация вычислительных
машин, комплексов, систем и сетей»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 5 лет

Год обучения 1 семестр 2

Период прохождения практики с 20.06.2025 по 17.07.2025

Кафедра «Вычислительная техника»

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

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

Руководитель практики к.т.н., доцент, Карамышева Н.С.

(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

ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ: ОЗНАКОМИТЕЛЬНОЙ ПРАКТИКЕ

(2024/2025 учебный год)

Семёнов Никита Дмитриевич

Направление подготовки (специальность) 09.05.01 «Применение и эксплуатация
автоматизированных систем специального назначения»

Наименование профиля подготовки (специализация) «Эксплуатация вычислительных
машин, комплексов, систем и сетей»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 5 лет

Год обучения 1 семестр 2

Период прохождения практики с 20.06.2025 по 17.07.2025

Кафедра «Вычислительная техника»

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

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

Руководитель практики к.т.н., доцент, Карамышева Н.С.

(должность, ученая степень, ученое звание)

№ п/п	Планируемая форма работы во время практики	Количество часов	Календарные сроки проведения работы	Подпись руководителя практики от вуза
1	Выбор темы и разработка индивидуального плана проведения работ (распределенная практика)	26	20.06.25 – 24.06.25	
2	Подбор и изучение материала по теме работы (распределенная практика)	26	24.06.25 – 26.06.25	
3	Разработка алгоритма (распределенная практика)	26	26.06.25 – 28.06.25	
4	Описание алгоритма и программы (сосредоточенная практика)	30	04.07.25 – 09.07.25	
5	Тестирование (сосредоточенная практика)	30	09.07.25 – 12.07.25	
6	Получение и анализ результатов (сосредоточенная	38	12.07.25 – 14.07.25	

	практика)			
7	Оформление отчёта (сосредоточенная практика)	40	14.07.25 – 17.07.25	
	Общий объём часов	216		

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

ОТЧЁТ

О ПРОХОЖДЕНИИ УЧЕБНОЙ (ОЗНАКОМИТЕЛЬНОЙ) ПРАКТИКИ

(2024/2025 учебный год)

Силявко Матвей Алексеевич

Направление подготовки (специальность) 09.05.01 «Применение и эксплуатация автоматизированных систем специального назначения»

Наименование профиля подготовки (специализация) «Эксплуатация вычислительных машин, комплексов, систем и сетей»

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

Год обучения 1 семестр 2

Период прохождения практики с 20.06.2025 по 17.07.2025

Кафедра «Вычислительная техника»

Силявко М.А. выполнял практическое задание «Сортировка Шелла». На первоначальном этапе были изучен и проанализирован алгоритм сортировки Шелла, был выбран метод решения и язык программирования C++, на котором была написана программа сортировки массива методом Шелла. Также, осуществил работу по ручному вводу первоначальных значений. Протестировал и отладил программу. Оформил отчёт.

Специалист Силявко М.А. _____ " ____ " _____ 2025 г.

Руководитель Карамышева Н.С. _____ " ____ " _____ 2025 г.
практики

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

ОТЧЁТ

О ПРОХОЖДЕНИИ УЧЕБНОЙ (ОЗНАКОМИТЕЛЬНОЙ) ПРАКТИКИ

(2024/2025 учебный год)

Юсупов Владислав Ренатович

Направление подготовки (специальность) 09.05.01 «Применение и эксплуатация
автоматизированных систем специального назначения»

Наименование профиля подготовки (специализация) «Эксплуатация вычислительных
машин, комплексов, систем и сетей»

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

Год обучения 1 семестр 2

Период прохождения практики с 20.06.2025 по 17.07.2025

Кафедра «Вычислительная техника»

Юсупов В.Р. выполнял практическое задание «Сортировка Шелла». На первоначальном этапе были изучен и проанализирован алгоритм сортировки Шелла, был выбран метод решения и язык программирования C++, на котором была написана программа сортировки массива методом Шелла. Также, осуществлял работу с файлами. Протестировал и отладил программу. Оформил отчёт.

Специалист Юсупов В.Р. _____ "___" _____ 2025 г.

Руководитель Карамышева Н.С. _____ "___" _____ 2025 г.
практики

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

ОТЧЁТ

О ПРОХОЖДЕНИИ УЧЕБНОЙ (ОЗНАКОМИТЕЛЬНОЙ) ПРАКТИКИ

(2024/2025 учебный год)

Семёнов Никита Дмитриевич

Направление подготовки (специальность) 09.05.01 «Применение и эксплуатация
автоматизированных систем специального назначения»

Наименование профиля подготовки (специализация) «Эксплуатация вычислительных
машин, комплексов, систем и сетей»

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

Год обучения 1 семестр 2

Период прохождения практики с 20.06.2025 по 17.07.2025

Кафедра «Вычислительная техника»

Семёнов Н.Д. выполнял практическое задание «Сортировка Шелла». На первоначальном этапе были изучен и проанализирован алгоритм сортировки Шелла, был выбран метод решения и язык программирования C++, на котором была написана программа сортировки массива методом Шелла. Также, осуществил работу по заполнению массива случайными значениями. Протестировал и отладил программу. Оформил отчёт.

Специалист Семёнов Н.Д. _____ "___" _____ 2025 г.

Руководитель Карамышева Н.С. _____ "___" _____ 2025 г.
практики

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

ОТЗЫВ

ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ: ОЗНАКОМИТЕЛЬНОЙ ПРАКТИКЕ

(2024/2025 учебный год)

Юсупов Владислав Ренатович

Направление подготовки (специальность) 09.05.01 «Применение и эксплуатация
автоматизированных систем специального назначения»

Наименование профиля подготовки (специализация) «Эксплуатация
вычислительных машин, комплексов, систем и сетей»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 5 лет

Год обучения 1 семестр 2

Период прохождения практики с 20.06.2025 по 17.07.2025

Кафедра «Вычислительная техника»

В процессе выполнения практики Юсупов В.Р. решал следующие задачи:
разработка алгоритма сортировки Шелла, а также работа с файлами.

За период прохождения практики были освоены основные понятия и технологии программирования, изучены основные инструменты языков C/C++. И получены следующие результаты: разработан алгоритм сортировки Шелла, получены результаты работы алгоритма, сделаны выводы эффективности работы алгоритма сортировки Шелла. Во время выполнения работы Юсупов В.Р. показал себя ответственным, добросовестным учеником, знающим свой предмет, имеющим представление о современном состоянии науки, владеющим современными общенаучными знаниями по информатике и вычислительной технике; программированию.

За выполнение работы Юсупов В.Р. заслуживает оценки « ».

Руководитель практики к.т.н., Карамышева Н.С. .« » 2025 г.

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

ОТЗЫВ

ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ: ОЗНАКОМИТЕЛЬНОЙ ПРАКТИКЕ

(2024/2025 учебный год)

Силявко Матвей Алексеевич

Направление подготовки (специальность) 09.05.01 «Применение и эксплуатация
автоматизированных систем специального назначения»

Наименование профиля подготовки (специализация) «Эксплуатация
вычислительных машин, комплексов, систем и сетей»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 5 лет

Год обучения 1 семестр 2

Период прохождения практики с 20.06.2025 по 17.07.2025

Кафедра «Вычислительная техника»

В процессе выполнения практики Силявко М.А. решал следующие задачи: разработка алгоритма сортировки Шелла, а также разработка ручного ввода стартовых значений.

За период прохождения практики были освоены основные понятия и технологии программирования, изучены основные инструменты языков C/C++. И получены следующие результаты: разработан алгоритм сортировки Шелла, получены результаты работы алгоритма, сделаны выводы эффективности работы алгоритма сортировки Шелла. Во время выполнения работы Силявко М.А. показал себя ответственным, добросовестным учеником, знающим свой предмет, имеющим представление о современном состоянии науки, владеющим современными общенаучными знаниями по информатике и вычислительной технике; программированию.

За выполнение работы Силявко М.А. заслуживает оценки « ».

Руководитель практики к.т.н., Карамышева Н.С. .« » 2025 г.

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

ОТЗЫВ

ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ: ОЗНАКОМИТЕЛЬНОЙ ПРАКТИКЕ

(2024/2025 учебный год)

Семёнов Никита Дмитриевич

Направление подготовки (специальность) 09.05.01 «Применение и эксплуатация
автоматизированных систем специального назначения»

Наименование профиля подготовки (специализация) «Эксплуатация
вычислительных машин, комплексов, систем и сетей»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 5 лет

Год обучения 1 семестр 2

Период прохождения практики с 20.06.2025 по 17.07.2025

Кафедра «Вычислительная техника»

В процессе выполнения практики Семёнов Н.Д. решал следующие задачи:
разработка алгоритма сортировки Шелла, анализ результатов сортировки, а также
разработка случайного заполнения массива стартовыми значениями.

За период прохождения практики были освоены основные понятия и
технологии программирования, изучены основные инструменты языков C/C++. И
получены следующие результаты: разработан алгоритм сортировки Шелла,
получены результаты работы алгоритма, сделаны выводы эффективности работы
алгоритма сортировки Шелла. Во время выполнения работы Семёнов Н.Д. показал
себя ответственным, добросовестным учеником, знающим свой предмет, имеющим
представление о современном состоянии науки, владеющим современными
общенаучными знаниями по информатике и вычислительной технике;
программированию.

За выполнение работы Семёнов Н.Д. заслуживает оценки « ».

Руководитель практики к.т.н., Карамышева Н.С. .« » 2025 г.

Содержание

Введение.....	2
1 Постановка задачи.....	3
1.1 Достоинства алгоритма сортировки Шелла.....	3
1.2 Недостатки алгоритма сортировки Шелла.....	3
1.3 Типичные сценарии применения данного алгоритма	3
2 Выбор решения.....	4
3 Описание программы.....	5
4. Схемы программы.....	9
4.1 Блок-схема программы.....	9
4.2 Блок-схема алгоритма.....	10
5 Тестирование программы.....	11
6 Отладка.....	12
7 Совместная разработка	14
Заключение	15
Список используемой литературы	16
Приложение А. Листинг программы.....	17

Введение

Сортировка данных на сегодняшний день при современном развитии компьютерных технологий является одним из наиболее распространенных процессов современной обработки данных. Задачи на сортировку данных встречаются очень часто в различных профессиональных сферах деятельности.

Алгоритмы сортировки образуют отдельный класс алгоритмов, применяются практически во всех задачах обработки информации. При этом они настолько тесно связаны друг с другом, что образуют отдельный класс алгоритмов. Алгоритмы сортировки, как правило, применяются с целью осуществления последующего более быстрого поиска. Например, трудно пользоваться словарями, если бы слова в них не были бы упорядочены по алфавиту.

Важность сортировки основана на том факте, что на ее примере можно показать многие основные фундаментальные приемы и методы построения алгоритмов. Сортировка является хорошим примером огромного разнообразия алгоритмов, которые выполняют одну и ту же задачу. Кроме того, многие из них имеют определенные преимущества друг перед другом. За счет усложнения алгоритма можно добиться существенного увеличения эффективности и быстродействия алгоритма по сравнению с более простыми методами. Как правило, термин сортировка понимают, как процесс перестановки объектов некоторого множества в определенном порядке.

Сортировка Шелла является стабильной. Алгоритм не самый лучший с точки зрения производительности, но традиционно более эффективен, чем большинство других простых алгоритмы, такие как сортировка выбором или же пузырьковая сортировка. Сортировка шелла также используется в гибридной сортировке, которая сочетает в себе различные алгоритмы для повышения производительности.

1 Постановка задачи

Поставленная задача: необходимо заполнить массив из n -ого количества элементов случайными числами, записать данные элементы в отдельный файл. После этого выполнить сортировку Шелла над данными, находящимися в массиве, записать отсортированные данные в другой файл, посчитать время выполнения и количество перестановок значений массива при сортировке.

Использовать сервис GitHub для совместной работы. Создать и выложить коммиты, характеризующие действия, выполненные каждым участником бригады.

Оформить отчет по проведенной практике.

1.1 Достоинства алгоритма сортировки Шелла

- не требует памяти под стек;
- не деградирует при неудачных наборах данных;

1.2 Недостатки алгоритма сортировки Шелла

- медленнее многих аналогов;
- сложен в понимании;
- не рекомендуется для сортировки больших массивов;

1.3 Типичные сценарии применения данного алгоритма

- товары в магазине (сортировка по цене, году выпуска, габаритам, весу, срокам поставки);
- студенты в вузе (сортировка по среднему балу, кол-ву прогулов, уровню IQ, числу хвостов, ФИО);
- города/страны (сортировка по населению, рождаемости, ВВП, ВВП на душу населения);

2 Выбор решения

Нашей бригадой было выбрано вести разработку в среде Microsoft Visual Studio на языке C.

Для написания данной программы будет использован язык программирования Си. Этот язык является распространённым языком программирования. При разработке языка Си был принят компромисс между низким уровнем языка ассемблера и высоким уровнем других языков. Си – это язык программирования общего назначения, хорошо известный своей эффективностью, экономичностью и переносимостью. Указанные преимущества Си обеспечивают хорошее качество разработки почти любого вида программного продукта.

Microsoft Visual Studio — это программная среда по разработке приложений для ОС Windows, как консольных, так и с графическим интерфейсом.

Для удобства совместной разработки был использован сервис WEEEEK. WEEEEK — сервис для управления личными и командными проектами. В основе WEEEEK лежит недельный планер и канбан-методология: доски, колонки и т. д. Проект динамично разрабатывается, регулярно расширяя функционал и возможности. Ведется активная работа с пожеланиями пользователей в еженедельном патчноте WEEEEK Week.

3 Описание программы

При запуске программы выводится меню из 3 пунктов:

- 1) ввод значений массива вручную;
- 2) чтение элементов массива из файла input.txt;
- 3) создание файла с массивом случайных чисел для чтения;

```
printf("Меню программы:\n");  
printf("1: Ввод массива вручную.\n");  
printf("2: Считать массив из файла input.txt.\n");  
printf("3: Создать файл с массивом случайных чисел.\n");
```

Пользователю требуется выбрать тот пункт, который ему требуется. При выборе 1 варианта выводится сообщение, в котором пользователю необходимо ввести содержимое массива через пробел.

```
printf("Введите числа через пробел:\n");  
scanf("%d%c", &arrayNum[currentCount++], &c)
```

После того, как данные были введены, созданный массив отсортировывается методом Шелла и итоговый его вид записывается в файл output.txt. Параллельно происходит подсчёт времени на сортировку.

```
while (scanf("%d%c", &arrayNum[currentCount++], &c) == 2) {  
    if (currentCount >= sizeArray) {  
        sizeArray *= 2;  
        arrayNum = realloc(arrayNum, sizeArray * sizeof(int));  
    }  
    if (c == '\n') break;  
}  
arrayNum = realloc(arrayNum, --currentCount * sizeof(int));  
time_t startTime = clock(); // Начало отсчёта  
shellSort(arrayNum, currentCount); // Сортировка массива  
time_t endTime = clock(); // Конец отсчёта
```

Сама сортировка происходит посредством функции shellSort.

При выборе 2 варианта происходит заполнение массива из файла. Пособством функции readFile.

```
int sizeArr = NULL;  
int* arrayNum = readFile(&sizeArr);
```

Затем программа занимается сортировкой Шелла посредством функции shellSort и подсчитывает время её выполнения. В случае ошибки программа выводит соответствующее сообщение.

```
if (arrayNum != NULL) {
```



```

time_t startTime = clock(); // Начало отсчёта
shellSort(arrayNum, sizeArr); // Сортировка массива
time_t endTime = clock(); // Конец отсчёта

outputFile(arrayNum, sizeArr);
printf("Отсортированный массив записан в файл output.txt");

double totalTime = (double)(endTime - startTime) / CLOCKS_PER_SEC; // Время
выполнения
printf("\nВремя выполнения: %.3f\n", totalTime);
}
else printf("Ошибка чтения файла.");
    При выборе варианта 3 пользователя просят ввести размер случайного
массива. Затем происходит функция createFile.

printf("Введите размер случайного массива:\n");
scanf("%d", &sizeArr);
createFile(sizeArr);
    Затем начинается выполнение функции readFile и если ошибки не
произошло, то начинается выполнение сортировки Шелла посредством функции
shellSort с подсчётом времени на её выполнение. Сгенерированный случайный
массив записывается в файл input.txt, отсортированная версия в output.txt.

if (arrayNum != NULL) {
    time_t startTime = clock(); // Начало отсчёта
    shellSort(arrayNum, sizeArr); // Сортировка массива
    time_t endTime = clock(); // Конец отсчёта

    outputFile(arrayNum, sizeArr);
    printf("Сгенерированный случайный массив записан в файл input.txt,\n а
отсортированный его вариант записан в файл output.txt");

    double totalTime = (double)(endTime - startTime) / CLOCKS_PER_SEC; // Время
выполнения
    printf("\nВремя выполнения: %.3f\n", totalTime);
}
else printf("Ошибка чтения файла.");
    Все функции собраны в отдельном файле mainFunctions.h, где находятся как
createFile, readFile и outputFile, так и алгоритм непосредственно сортировки Шелла
в функции shellSort.

```

Эта функция работает с указателями на основной массив и работает по следующему принципу: в каждую итерацию шаги между сортируемыми элементами уменьшаются в 2 раза, пока не будут меняться местами соседние элементы. Равные элементы также меняются местами. Мы сравниваем так

элементы и найдя тот, который надо поменять, запоминаем его через доп.переменную, затем сдвигаем все остальные ячейки массива пока не найдём место для отложенной и вставляем её.

```
void shellSort(int* arr, int cntEl) {
    for (int step = cntEl / 2; step > 0; step /= 2)
        for (int i = step; i < cntEl; i++) {
            int temp = arr[i];
            int k;
            for (k = i; k >= step && arr[k - step] > temp; k -= step) {
                arr[k] = arr[k - step];
            }
            arr[k] = temp;
        }
}
```

Функция createFile выполняет задачу создания файла input.txt если такового нет или его открытие для редактирования, а затем заполнение его случайными значениями.

```
int createFile(int sizeArray) {
    FILE* file = fopen("input.txt", "w");
    if (file == NULL) {
        printf("\nERROR: file creation error\n");
        fclose(file);
        return 1;
    }

    srand(time(NULL));

    for (int i = 0; i < sizeArray; i++) fprintf(file, "%d\n", rand());

    fclose(file);

    return 0;
}
```

Функция readFile отвечает за открытие файла input.txt для чтения и инициализацию массива значениями из этого файла.

```
int* readFile(int *sizeArr) {
    FILE* file = fopen("input.txt", "r");
    if (file == NULL) {
        printf("\nERROR: file reading error\n");
        fclose(file);
        return NULL;
    }
    int currentCount = 0;
    int sizeArray = 10;
    int* numbers = malloc(sizeArray * sizeof(int));
    int num;
```

```

while (fscanf(file, "%d", &num) == 1) {
    if (++currentCount == sizeArray) {
        sizeArray *= 2;
        numbers = realloc(numbers, sizeArray * sizeof(int));
    }
    numbers[currentCount - 1] = num;
}
numbers = realloc(numbers, currentCount * sizeof(int));

*sizeArr = currentCount;

fclose(file);

return numbers;
}

```

Функция `outputFile` отвечает за открытие файла `output.txt` для редактирования или создание его заново при отсутствии, а затем выводит его содержимое.

```

int outputFile(int* arrayNum, int sizeArray) {
    FILE* file = fopen("output.txt", "w");
    if (file == NULL) {
        fclose(file);
        return 1;
    }

    for (int i = 0; i < sizeArray; i++) fprintf(file, "%d\n", arrayNum[i]);

    fclose(file);

    return 0;
}

```

Подробный алгоритм работы программы и функции сортировки представлен в разделе 4 на рисунках 1, 2.

Листинг программы приведен в приложении А.

4. Схемы программы

4.1 Блок-схема программы

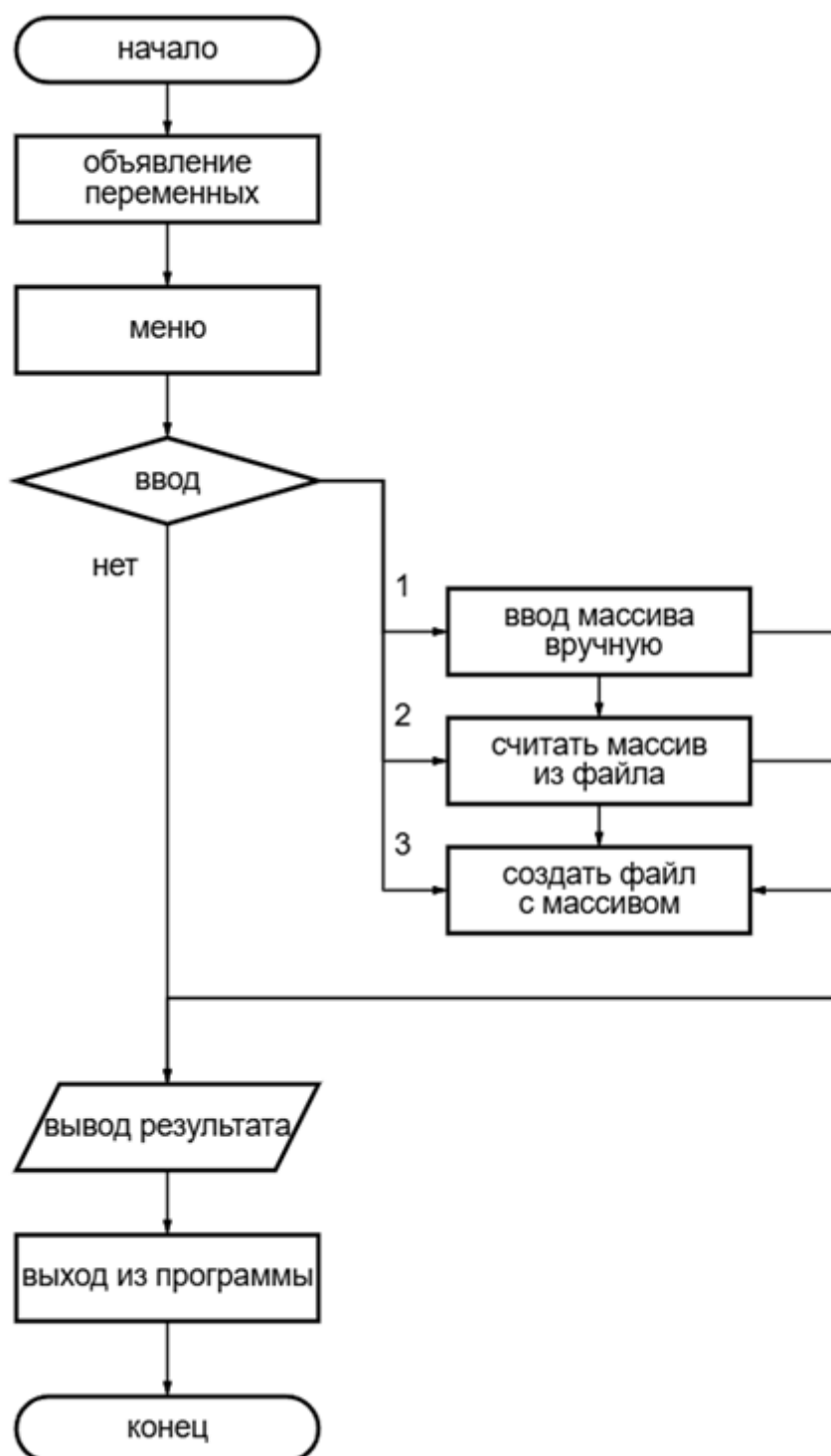


Рисунок 1 - Блок-схема программы

4.2 Блок-схема алгоритма

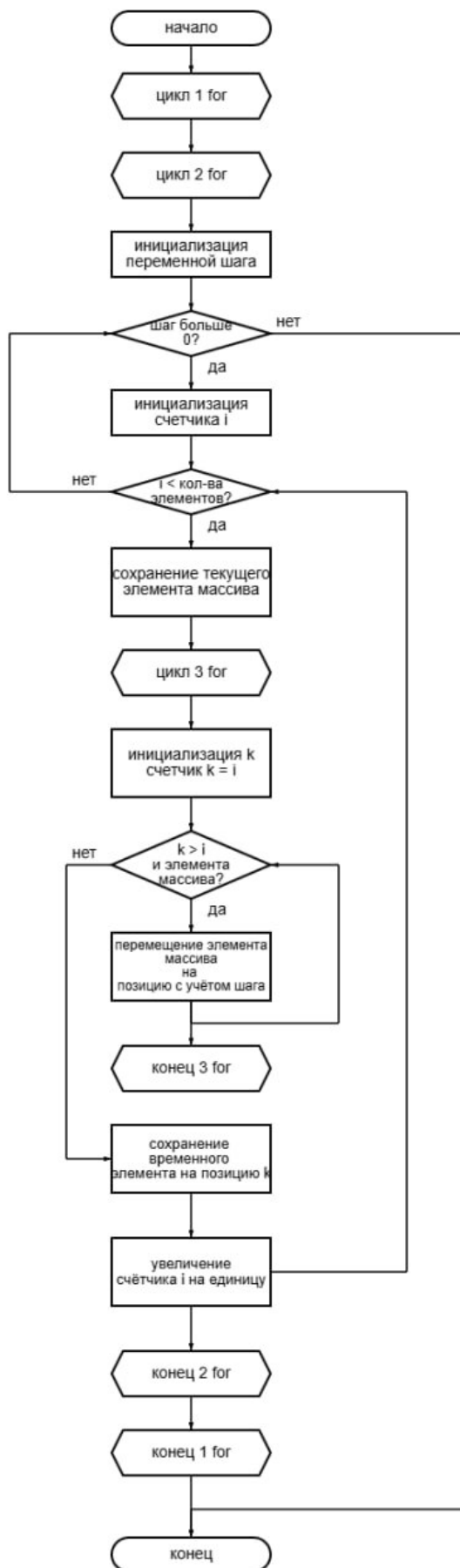


Рисунок 2 - Блок-схема алгоритма с подключенными файлами

5 Тестирование программы

Тестирование показало, что с увеличением количества элементов пропорционально увеличивается время работы программы, ниже представлен график результатов тестирования



Рисунок 3 – Результаты тестирования

6 Отладка

В качестве среды разработки была выбрана программа Microsoft Visual Studio, которая содержит в себе все необходимые средства для разработки и отладки модулей и программ.

Для отладки программы использовались точки останова и пошаговое выполнение кода программы, анализ содержимого локальных переменных.

Точки останова – это прерывание выполнения программы, при котором выполняется вызов отладчика. Отладчик является инструментом для поиска и устранения ошибок в программе, с помощью которого можно исследовать состояние программы.

Был использован метод бинарного поиска, он включает в себя разделение частей кода для упрощения процесса отладки. Это может быть особенно полезно, если причина ошибки находится в начале языка программирования, а фактическая ошибка ближе к концу.

Команда шаг с заходом (step into) выполняет следующую инструкцию в обычном пути выполнения программы, а затем приостанавливает выполнение программы, чтобы мы могли проверить состояние программы с помощью отладчика. Если выполняемый оператор содержит вызов функции, шаг с заходом заставляет программу перескакивать в начало вызываемой функции, где она приостанавливается.

7 Совместная разработка

Для удобства совместной разработки был использован сервис WEEK.

Во время работы над данной практикой наша бригада осуществляла совместную работу в GitHub.

Мною были подключены файлы для записи массивов вручную для последующей сортировки, это было зафиксировано и загружено на удаленный репозиторий Github, на ветку main.

Заключение

При выполнении данной работы были получены навыки совместной работы с помощью сервиса GitHub, навыки использования программы Git Bash. Был изучен алгоритм сортировки вставками.

Мною было создано меню программы, позволяющее выбрать одну из нескольких опций программы. Также я осуществил ввод вручную стартовых значений массива для последующей сортировки.

При выполнении практической работы были улучшены базовые навыки программирования на языке C. Улучшены навыки отладки, тестирования программ и работы со сложными типами данных.

В дальнейшем программу можно улучшить путем подключения упрощающих реализацию данной сортировки библиотек и улучшения графического интерфейса.

7 Совместная разработка

Для удобства совместной разработки был использован сервис WEEK.

Во время работы над данной практикой наша бригада осуществляла совместную работу в GitHub.

Мною были подключены файлы для заполнения массивов случайными значениями для последующей сортировки, это было зафиксировано и загружено на удаленный репозиторий Github, на ветку main.

Заключение

При выполнении данной работы были получены навыки совместной работы с помощью сервиса GitHub, навыки использования программы Git Bash. Был изучен алгоритм сортировки вставками.

Мною было создано меню программы, позволяющее выбрать одну из нескольких опций программы. Также я осуществил заполнение случайных стартовых значений массива для последующей сортировки.

При выполнении практической работы были улучшены базовые навыки программирования на языке C. Улучшены навыки отладки, тестирования программ и работы со сложными типами данных.

В дальнейшем программу можно улучшить путем подключения упрощающих реализацию данной сортировки библиотек и улучшения графического интерфейса.

7 Совместная разработка

Для удобства совместной разработки был использован сервис WEEK.

Во время работы над данной практикой наша бригада осуществляла совместную работу в GitHub.

Мною были подключены файлы для заполнения массивов из текстовых файлов для последующей сортировки, это было зафиксировано и загружено на удаленный репозиторий Github, на ветку main.

Заключение

При выполнении данной работы были получены навыки совместной работы с помощью сервиса GitHub, навыки использования программы Git Bash. Был изучен алгоритм сортировки вставками.

Мною было создано меню программы, позволяющее выбрать одну из нескольких опций программы. Также я осуществил заполнение стартовых значений массива из текстового файла для последующей сортировки.

При выполнении практической работы были улучшены базовые навыки программирования на языке C. Улучшены навыки отладки, тестирования программ и работы со сложными типами данных.

В дальнейшем программу можно улучшить путем подключения упрощающих реализацию данной сортировки библиотек и улучшения графического интерфейса.

Список используемой литературы

1. ГОСТ 19.701 – 90 Схемы алгоритмов, программ, данных и систем.
2. Керниган, Брайан У., Ритчи, Деннис М. Язык программирования С, 2-е издание.: Пер. с англ. – М., 2009.
3. Сортировка Шелла [Электронный ресурс] – URL: <https://ru.wikipedia.org> (дата обращения: 02.07.2025 г)

Приложение А. Листинг программы

main.c:

```
#define _CRT_SECURE_NO_WARNINGS
#define _CRTDBG_MAP_ALLOC
#include "mainFunctions.h"
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
#include <Windows.h>

void main() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);

    printf("Меню программы:\n");
    printf("1: Ввод массива вручную.\n");
    printf("2: Считать массив из файла input.txt.\n");
    printf("3: Создать файл с массивом случайных чисел.\n");

    char operation = getchar();
    getchar();

    switch (operation)
    {
    case '1': {
        printf("Введите числа через пробел:\n");
        int currentCount = 0, sizeArray = 10;
        int* arrayNum = malloc(sizeArray * sizeof(int));
        char c;

        while (scanf("%d%c", &arrayNum[currentCount++], &c) == 2) {
            if (currentCount >= sizeArray) {
                sizeArray *= 2;
                arrayNum = realloc(arrayNum, sizeArray * sizeof(int));
            }
            if (c == '\n') break;
        }
        arrayNum = realloc(arrayNum, --currentCount * sizeof(int));

        time_t startTime = clock(); // Начало отсчёта
        shellSort(arrayNum, currentCount); // Сортировка массива
        time_t endTime = clock(); // Конец отсчёта

        outputFile(arrayNum, currentCount);
        printf("Отсортированный массив записан в файл output.txt");

        double totalTime = (double)(endTime - startTime) / CLOCKS_PER_SEC; // Время
        выполнения
        printf("\nВремя выполнения: %.3f\n", totalTime);
    }
```

```

    free(arrayNum);

    break;
}

case '2': {
    int sizeArr = NULL;
    int* arrayNum = readFile(&sizeArr);
    if (arrayNum != NULL) {
        time_t startTime = clock(); // Начало отсчёта
        shellSort(arrayNum, sizeArr); // Сортировка массива
        time_t endTime = clock(); // Конец отсчёта

        outputFile(arrayNum, sizeArr);
        printf("Отсортированный массив записан в файл output.txt");

        double totalTime = (double)(endTime - startTime) / CLOCKS_PER_SEC; // Время
        выполнения
        printf("\nВремя выполнения: %.3f\n", totalTime);
    }
    else printf("Ошибка чтения файла.");

    break;
}

case '3': {
    int sizeArr;
    printf("Введите размер случайного массива:\n");
    scanf("%d", &sizeArr);
    createFile(sizeArr);

    int* arrayNum = readFile(&sizeArr);
    if (arrayNum != NULL) {
        time_t startTime = clock(); // Начало отсчёта
        shellSort(arrayNum, sizeArr); // Сортировка массива
        time_t endTime = clock(); // Конец отсчёта

        outputFile(arrayNum, sizeArr);
        printf("Сгенерированный случайный массив записан в файл input.txt,\n а
        отсортированный его вариант записан в файл output.txt");

        double totalTime = (double)(endTime - startTime) / CLOCKS_PER_SEC; // Время
        выполнения
        printf("\nВремя выполнения: %.3f\n", totalTime);
    }
    else printf("Ошибка чтения файла.");

    break;
}

default:
    break;
}
}

```


mainFunctions.c:

```
#define _CRT_SECURE_NO_WARNINGS
#include "mainFunctions.h"
```

```
void shellSort(int* arr, int cntEl) {
    for (int step = cntEl / 2; step > 0; step /= 2)
        for (int i = step; i < cntEl; i++) {
            int temp = arr[i];
            int k;
            for (k = i; k >= step && arr[k - step] > temp; k -= step) {
                arr[k] = arr[k - step];
            }
            arr[k] = temp;
        }
}
```

```
int createFile(int sizeArray) {
    FILE* file = fopen("input.txt", "w");
    if (file == NULL) {
        printf("\nERROR: file creation error\n");
        fclose(file);
        return 1;
    }

    srand(time(NULL));

    for (int i = 0; i < sizeArray; i++) fprintf(file, "%d\n", rand());

    fclose(file);

    return 0;
}
```

```
int* readFile(int *sizeArr) {
    FILE* file = fopen("input.txt", "r");
    if (file == NULL) {
        printf("\nERROR: file reading error\n");
        fclose(file);
        return NULL;
    }

    int currentCount = 0;
    int sizeArray = 10;
    int* numbers = malloc(sizeArray * sizeof(int));
    int num;
```

```
    while (fscanf(file, "%d", &num) == 1) {
        if (++currentCount == sizeArray) {
```

```

        sizeArray *= 2;
        numbers = realloc(numbers, sizeArray * sizeof(int));
    }
    numbers[currentCount - 1] = num;
}
numbers = realloc(numbers, currentCount * sizeof(int));

*sizeArr = currentCount;

fclose(file);

return numbers;
}

int outputFile(int* arrayNum, int sizeArray) {
    FILE* file = fopen("output.txt", "w");
    if (file == NULL) {
        fclose(file);
        return 1;
    }

    for (int i = 0; i < sizeArray; i++) fprintf(file, "%d\n", arrayNum[i]);

    fclose(file);

    return 0;
}

```

mainFunctions.h:

```

#pragma once
#include <stdio.h>
#include <time.h>
#include <stdlib.h>

void shellSort(int* arr, int cntEl);
int createFile(int sizeArray);
int* readFile(int *sizeArr);
int outputFile(int* arrayNum, int sizeArray);

```