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**

(2023/2024 учебный год)

\_\_\_\_ Макаров Алексей Сергеевич

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Наименование профиля подготовки «Вычислительные машины, комплексы, системы и сети»

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

Год обучения1семестр 2

Период прохождения практики с 25.06.2024 по 8.07.2024

Кафедра «Вычислительная техника»

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

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

Руководитель практики к.т.н., профессор, 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 |
|  |  |

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

(2023/2024 учебный год)

                                            Макаров Алексей Сергеевич

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Наименование профиля подготовки «Вычислительные машины, комплексы, системы и сети»

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

Год обучения                  1                семестр                 2

Период прохождения практики с 25.06.2024 по 8.07.2024

Кафедра «Вычислительная техника»

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

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

Руководитель практики к.т.н., профессор 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 | 2 | 25.06.24 –  25.06.24 |  |
| 2 | Подбор и изучение материала по теме работы | 15 | 26.06.24 –  29.06.24 |  |
| 3 | Разработка алгоритма | 43 | 29.06.24 –  03.07.24 |  |
| 4 | Описание алгоритма | 18 | 03.07.24 –  05.07.24 |  |
| 5 | Тестирование | 5 | 05.07.24 –  06.07.24 |  |
| 6 | Тестирование и отладка | 38 | 06.07.24 –  07.07.24 |  |
| 7 | Оформление отчёта | 40 | 07.07.24 –  08.07.24 |  |
|  | **Общий объём часов** | 108 |  |  |

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

**ОТЧЁТ**

**О ПРОХОЖДЕНИИ УЧЕБНОЙ (ОЗНАКОМИТЕЛЬНОЙ) ПРАКТИКИ**

(2023/2024 учебный год)

Макаров Алексей Сергеевич

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Наименование профиля подготовки «Вычислительные машины, комплексы, системы и сети»

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

Год обучения1семестр 2

Период прохождения практики с 25.06.2024 по 8.07.2024

Кафедра «Вычислительная техника»

Макаров А.С. выполнял практическое задание «Сортировка вставками». На первоначальном этапе были изучен и проанализирован алгоритм сортировки

вставками, был выбран метод решения и язык программирования С, на

котором была написана программа сортировки массива методом вставки. Также, протестировал программу. Оформил отчёт.

Бакалавр Макаров А.С. \_\_\_\_\_\_\_\_\_\_\_\_ "\_\_\_" \_\_\_\_\_\_\_\_\_\_\_ 2024 г.

Руководитель Карамышева Н.С.\_\_\_\_\_\_\_\_\_\_\_\_ "\_\_\_" \_\_\_\_\_\_\_\_\_\_\_ 2024 г.

практики

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

**ОТЗЫВ**

**О ПРОХОЖДЕНИИ УЧЕБНОЙ (ОЗНАКОМИТЕЛЬНОЙ) ПРАКТИКИ**

(2023/2024 учебный год)

Макаров Алексей Сергеевич

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Наименование профиля подготовки «Вычислительные машины, комплексы, системы и сети»

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

Год обучения1семестр 2

Период прохождения практики с 25.06.2024 по 8.07.2024

Кафедра «Вычислительная техника»

В процессе выполнения практики Макаров А.С. решал следующие задачи:

Создание алгоритма сортировки вставками, анализировал работу алгоритма и реализовал его, сравнение существующих методов сортировки.

За период выполнения практики были освоены основные понятия и технологии двоичной сортировки, реализован метод работы с файлами. Во время выполнения работы Макаров А.С. показал себя ответственным, добросовестным учеником, знающим свой предмет, имеющим представление о современном состоянии науки, владеющим современными общенаучными знаниями по информатике и вычислительной технике.

За выполнение работы Макаров А.С. заслуживает оценки «\_\_\_\_\_\_».

Руководитель практики к.т.н, профессор Карамышева Н.С « » 2024 г.

**Содержание**

Оглавление

Содержание………………………………………………………………………………………………………………………………………….7

[Введение 8](#_Toc170491808)

[Постановка задачи 9](#_Toc170491809)

[Выбор решения 10](#_Toc170491810)

Отладка………………………………………………………………………………………………………………………………………………10

[Описание программы 11](#_Toc170491811)

[Заключение 21](#_Toc170491812)

[Можно с уверенностью сказать, что задача, поставленная в начале работы, выполнена. Программа соответствует заданным требованиям и выполняет свои функции. 21](#_Toc170491813)

[В ходе работы были освоены новые методы программирования. Закреплены старые знания работы с языком С. В ходе подготовки к практической работе были анализированы различные методы и алгоритмы реализации данной программы. 21](#_Toc170491814)

[Не смотря на внешне кажущуюся простоту данной программы, она таит в себе ряд сложностей, которые реализуются с использованием всех основных приемов Visual Studio C++. 21](#_Toc170491815)

[Приложения 22](#_Toc170491816)

[Листинг: 22](#_Toc170491817)

# Введение

Целью данной практической работы является необходимость реализовать и продемонстрировать работу алгоритмов сортировки или поиска элементов из списка. Конечная программа должна реализовывать непосредственно алгоритм, а также выполнять считывание входных данных из файла и запись результатов в файл и отображать все данные в консольном интерфейсе

# Постановка задачи

Поставленная задача: необходимо заполнить массив из n-ого количества элементов случайными числами, записать данные элементы в отдельный файл. После этого выполнить сортировку вставками над данными, находящимися в массиве, записать отсортированные данные в другой файл, подсчитать время выполнения.

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

Оформить отчет по практике

1. **Достоинства алгоритма сортировки вставками**

– алгоритм удобен для работы с массивами небольшого размера или на почти отсортированном наборе данных;

– алгоритм эффективен при работе со списками;

– простая реализация алгоритма.

1. **Недостатки алгоритма сортировки вставками**

– очень много перемещений элементов массива;

– высокая алгоритмическая сложность О(n²);

– не рекомендуется для сортировки больших массивов.

1. **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);

– астрономические объекты (масса, размеры, плотность).

# В**ыбор решения**

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. В качестве среды программирования была выбрана программа Microsoft Visual Studio. Microsoft Visual Studio — это программная среда по разработке приложений для ОС Windows, как консольных, так и с графическим интерфейсом.

# Описание программы

Программа предназначена для генерации и сортировки случайных чисел. Она позволяет пользователю: Ввести размер массива для генерации чисел, создать рандомную последовательность чисел и записать ее в файл, прочитать данные из файла и записать в массив, отсортировать массив используя метод вставки, записать отсортированные данные в файл, а так же программа отображает время затраченное на сортировку в файле.

При запуске программы выводится меню из пяти пунктов:

а) Ввод размера массива чисел;

б) Создание случайной последовательности и записи в файл;

в) Чтение данных из файла;

г) Отсортировать список используя метод Вставками сортировки;

д) Загрузить отсортированные данные в файл.

е) Выход из программы.

1. **writeFile**(int size)

Функция генерирует случайную последовательность целых чисел и записывает их в файл с именем "input.txt". Она принимает целое число size в качестве входного параметра, которое указывает количество случайных чисел для генерации.

Она инициализирует генератор случайных чисел с помощью srand(time(0)).

Она открывает файл с именем "input.txt" в режиме записи ("w").

Она генерирует size случайных целых чисел между -1000 и 1000 с помощью rand().

Она записывает каждое случайное число в файл с помощью fprintf.

Она закрывает файл с помощью fclose.

2. **writeFileData**(int\* numbers, int size)

Функция записывает массив целых чисел в файл с именем "output.txt". Она принимает указатель на массив целых чисел numbers и целое число size в качестве входных параметров.

Она открывает файл с именем "output.txt" в режиме записи ("w").

Она записывает каждый элемент массива numbers в файл с помощью fprintf.

Она закрывает файл с помощью fclose.

3. **readFile**(const char\* filename, int\* numbers, int\* count, int size)

Функция читает данные из файла и записывает их в массив целых чисел. Она принимает имя файла filename, указатель на массив целых чисел numbers, указатель на счетчик count и целое число size в качестве входных параметров.

Она открывает файл с именем filename в режиме чтения ("r").

Она инициализирует счетчик count в ноль.

Она читает данные из файла и записывает их в массив numbers с помощью fscanf.

Она закрывает файл с помощью fclose.

4. **insertion\_sort**(int arr[], int n)

Функция сортирует массив целых чисел с помощью алгоритма вставки. Она принимает массив целых чисел arr и целое число n в качестве входных параметров.

Она измеряет время начала сортировки с помощью clock().

Она сортирует массив с помощью алгоритма вставки.

Она измеряет время конца сортировки с помощью clock().

Она вычисляет время сортировки и записывает его в файл с именем "time.txt" с помощью fprintf.

5**. main**()

Функция main является точкой входа в программу. Она реализует меню для пользователя, которое позволяет выбрать действие: ввести размер массива, создать случайную последовательность и записать ее в файл, прочитать данные из файла, отсортировать список с помощью алгоритма вставки, загрузить отсортированные данные в файл или выйти из программы.

**Схема программы**

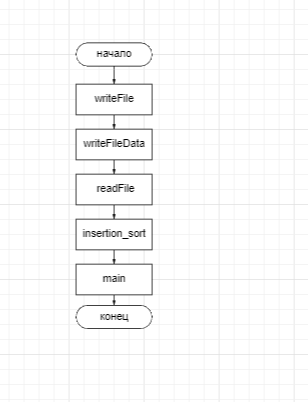
****

Рисунок 1 - Общая схема программы

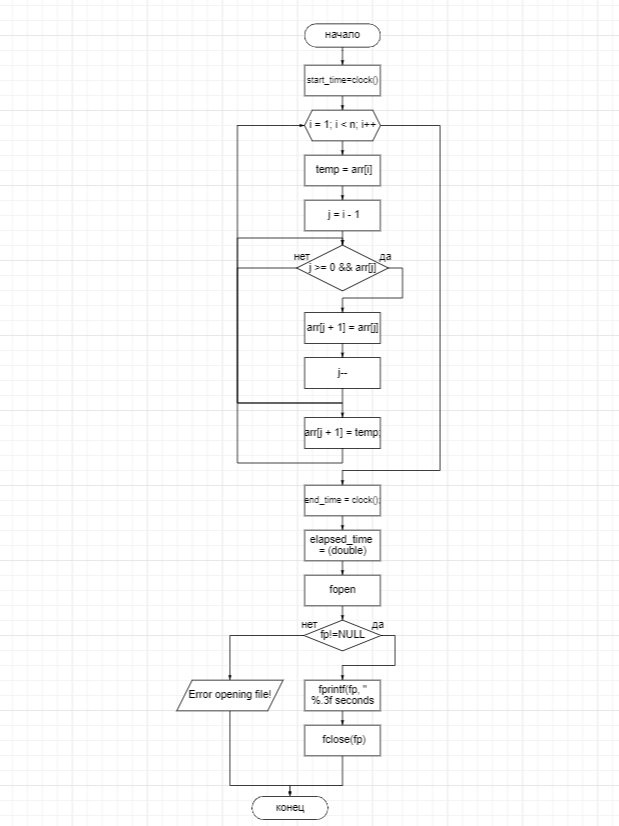


Рисунок 2 - Функция insertion\_sort

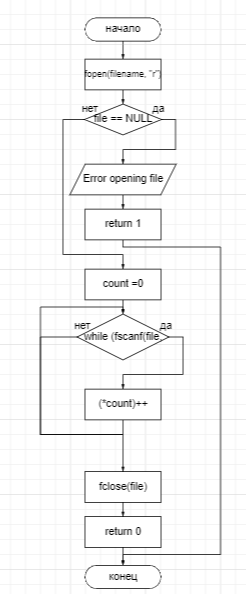


Рисунок 3 - Функция readFile

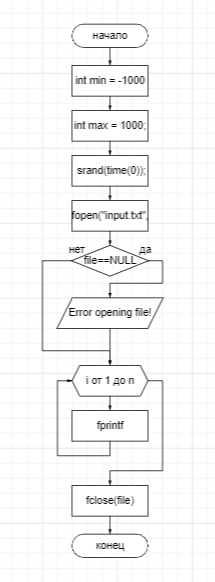


Рисунок 4 – Функция witeFile

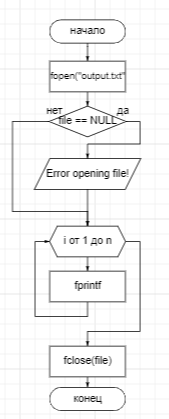


Рисунок 5 - Функция writeFileData

**Тестирование программы**

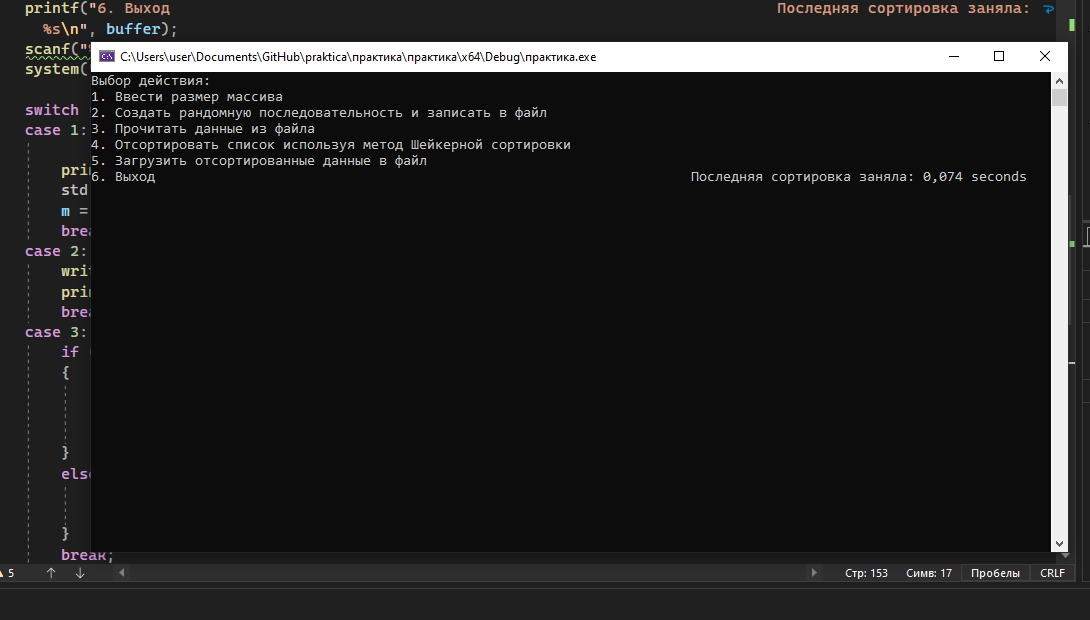
****

Рисунок 1 - Основное меню**.**

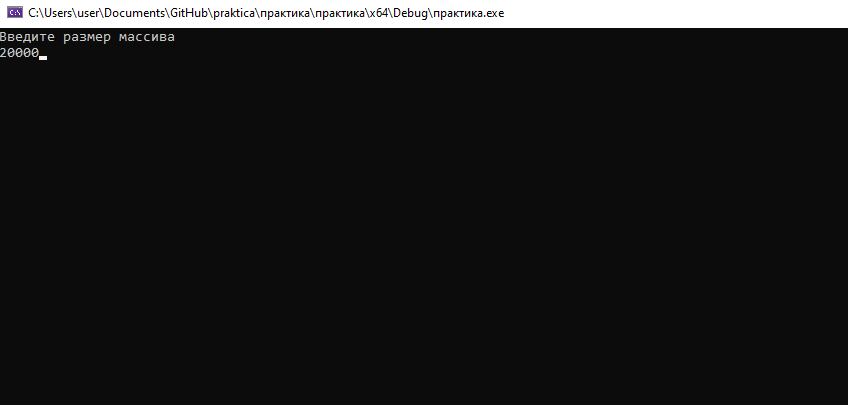
****

Рисунок 2 - Ввод массива.

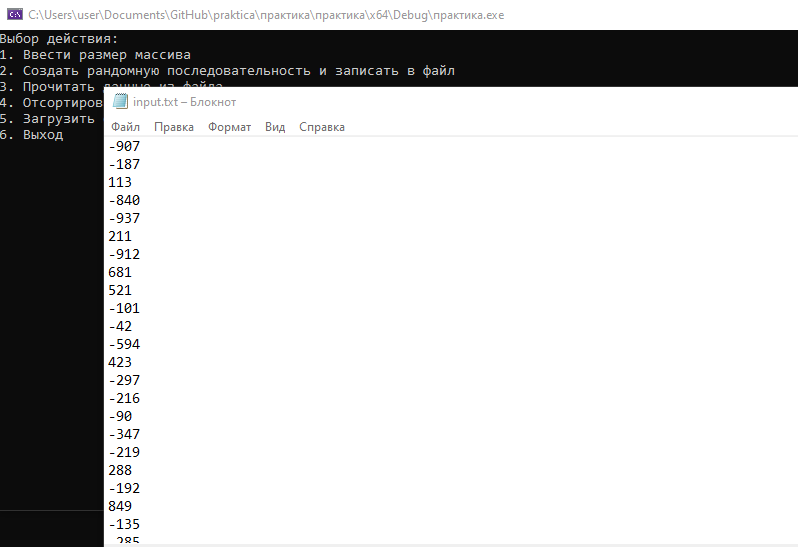
****

Рисунок 3 - Создание массива со случайными числами.

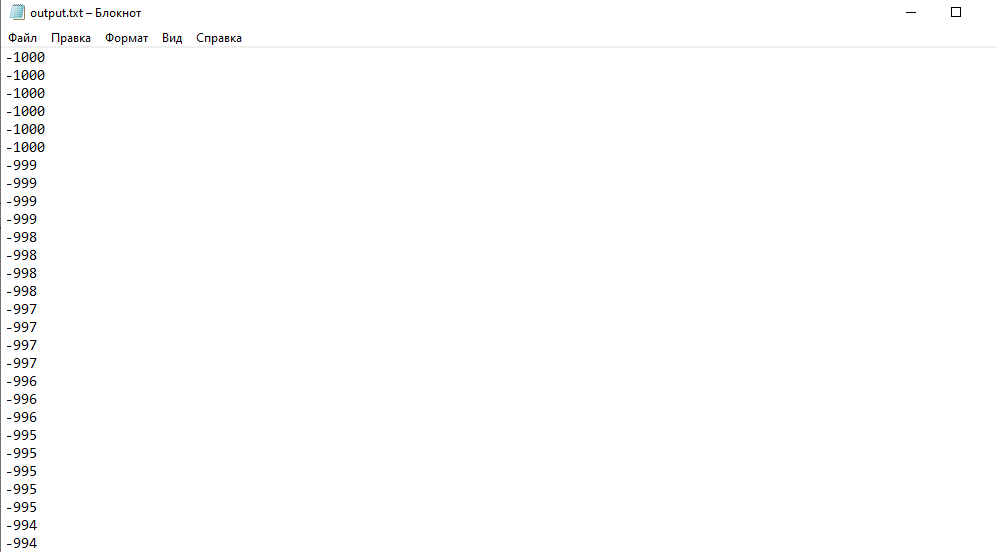
****

Рисунок 4 - Отсортированный массив.

****

Рисунок 5 - Время сортировки.

**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 остановки 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.

**Совместная работа**

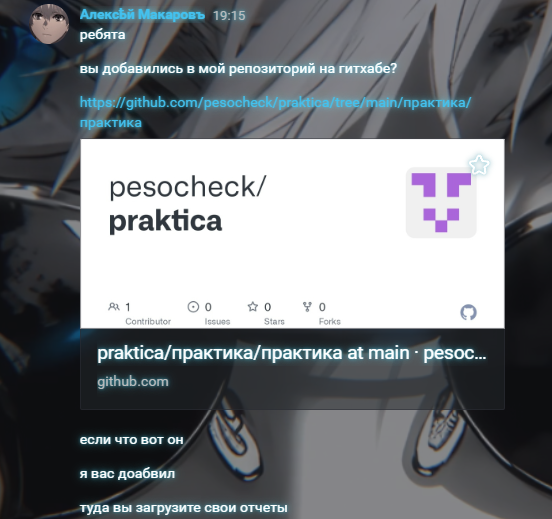
****

Рисунок 1 -Добавление команды в общий репозиторий и обмен информацией

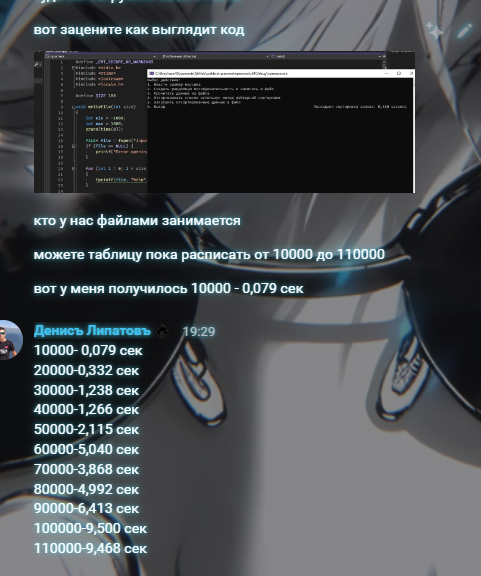
****

Рисунок 2 - Помощь в написании от 10000 тысяч символов до 110000. <https://github.com/pesocheck/praktica/tree/main/практика/практика>

# Заключение

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

Мною был написан алгоритм, осуществляющий сортировку вставками для массива случайно сгенерированных и уже отсортированных чисел, оформлен отчет по данной практике.

При выполнении практической работы были улучшены базовые навыки программирования на языке С. 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.

# Приложения

Листинг:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <ctime>

#include <iostream>

#include <locale.h>

#define SIZE 100

void writeFile(int size)

{

int min = -1000;

int max = 1000;

srand(time(0));

FILE\* file = fopen("input.txt", "w");

if (file == NULL) {

printf("Error opening file!\n");

}

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

{

fprintf(file, "%d\n", (rand() % (max - min + 1) + min));

}

fclose(file);

}

void writeFileData(int\* numbers, int size)

{

FILE\* file = fopen("output.txt", "w");

if (file == NULL) {

printf("Error opening file!\n");

}

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

{

fprintf(file, "%d\n", numbers[i]);

}

fclose(file);

}

int readFile(const char\* filename, int\* numbers, int\* count, int size) {

FILE\* file = fopen(filename, "r");

if (file == NULL) {

printf("Error opening file!\n");

return 1;

}

\*count = 0;

while (fscanf(file, "%d", &numbers[\*count]) == 1 && \*count < size) {

(\*count)++;

}

fclose(file);

return 0;

}

void insertion\_sort(int arr[], int n)

{

clock\_t start\_time = clock();

int i, j, temp;

for (i = 1; i < n; i++) {

temp = arr[i];

j = i - 1;

while (j >= 0 && arr[j] > temp) {

arr[j + 1] = arr[j];

j--;

}

arr[j + 1] = temp;

}

clock\_t end\_time = clock();

double elapsed\_time = (double)(end\_time - start\_time) / CLOCKS\_PER\_SEC;

FILE\* fp = fopen("time.txt", "w");

if (fp != NULL) {

fprintf(fp, "%.3f seconds", elapsed\_time);

fclose(fp);

}

else {

printf("Error opening file!\n");

}

}

int main()

{

setlocale(LC\_ALL, "Rus");

int choice;

int size = 0;

int\* m = nullptr;

int count;

char buffer[256];

while (1) {

FILE\* fp = fopen("time.txt", "r");

if (fp != NULL) {

fgets(buffer, 256, fp);

fclose(fp);

}

printf("Выбор действия:\n");

printf("1. Ввести размер массива\n");

printf("2. Создать рандомную последовательность и записать в файл\n");

printf("3. Прочитать данные из файла\n");

printf("4. Отсортировать список используя метод Шейкерной сортировки\n");

printf("5. Загрузить отсортированные данные в файл\n");

printf("6. Выход Последняя сортировка заняла: %s\n", buffer);

scanf("%d", &choice);

system("cls");

switch (choice) {

case 1:

printf("Введите размер массива\n");

std::cin >> size;

m = new int[size];

break;

case 2:

writeFile(size);

printf("Данные записанный в файл.\n");

break;

case 3:

if (!m)

{

printf("Укажите размер массива");

std::cin.get();

std::cin.get();

}

else {

readFile("input.txt", m, &count, size);

printf("Данные прочитанный с файла.\n");

}

break;

case 4:

insertion\_sort(m, size);

printf("Данные отсортированный.\n");

break;

case 5:

writeFileData(m, size);

printf("Данные записанны в файл.\n");

break;

case 6:

delete[] m;

return 0;

default:

printf("Ошибка.\n Попробуте снова.\n");

}

system("cls");

}

return 0;

}