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**

(2022/2023 учебный год)

Емелин Леонид 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 обучения1семестр 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, Митрохин М.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 ПРАКТИКИ:**

**ОЗНАКОМИТЕЛЬНОЙ ПРАКТИКИ**

(2022/2023 учебный год)

Емелин Леонид 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 обучения1семестр 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, Митрохин М.А.

*(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 | 29.06.2023 -  29.06.2023 |  |
| 2 | Подбор и изучение материала по теме работы | 15 | 30.06.2023 –  02.07.23 |  |
| 3 | Разработка алгоритма | 43 | 02.07.23 –  06.07.23 |  |
| 4 | Описание алгоритма и программы | 18 | 06.07.23 –  08.07.23 |  |
| 5 | Тестирование | 5 | 08.07.23 –  08.07.23 |  |
| 6 | Получение и анализ результатов | 10 | 08.07.23 –  10.07.23 |  |
| 7 | Оформление отчёта | 15 | 10.07.23 –  12.07.2023 |  |
|  | **Общий объём**  **часов** | 108 |  |  |

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

**ОТЧЁТ**

**О ПРОХОЖДЕНИИ УЧЕБНОЙ ПРАКТИКЕ: ОЗНАКОМИТЕЛЬНОЙ ПРАКТИКЕ**

(2022/2023 учебный год)

Емелин Леонид 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 обучения1семестр 2

Период прохождения практики с 29.06.2023 по 12.07.2023

Кафедра «Вычислительная техника»

Емелин Л.А. выполнял практическое задание «Быстрая сортировка». На первоначальном этапе были изучен и проанализирован алгоритм быстрой сортировки, был выбран метод решения и язык программирования С++, на котором была написана программа сортировки массива методом быстрой сортировки. Оформил отчёт.

Специалист Емелин Л.А. \_\_\_\_\_\_\_\_\_\_\_\_ "\_\_\_" \_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Руководитель Митрохин М.А. \_\_\_\_\_\_\_\_\_\_\_\_ "\_\_\_" \_\_\_\_\_\_\_\_\_\_\_ 2023 г.

практики

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

**ОТЗЫВ ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ: ОЗНАКОМИТЕЛЬНОЙ ПРАКТИКЕ**

(2022/2023 учебный год)

Емелин Леонид 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 обучения1семестр 2

Период прохождения практики с 29.06.2023 по 12.07.2023

Кафедра «Вычислительная техника»

В процессе выполнения практики Емелин Л.А. решал следующие задачи:

работа с файлами, анализ работы алгоритма, сравнение существующих методов с разработанным алгоритмом.

За период выполнения практики были освоены основные понятия и технологии быстрой сортировки, реализован метод работы с файлом. Во время выполнения работы Емелин Л.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 и сортировке.

За выполнение работы Емелин Л.А. заслуживает оценки «\_\_\_\_\_\_».

Руководитель практики д.т.н., Митрохин М.А. « »\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г.

Содержание

[Введение 2](#_Toc24122)

[1 Постановка задачи 3](#_Toc24123)

[1.1 Достоинства алгоритма быстрой сортировки 3](#_Toc24124)

[1.2 Недостатки алгоритма быстрой сортировки 3](#_Toc24125)

[1.3 Типичные сценарии применения данного алгоритма 3](#_Toc24126)

[2 Выбор решения 4](#_Toc24127)

[3 Описание программы 5](#_Toc24128)

[4. Схемы программы 7](#_Toc24129)

[4.1 Блок-схема программы 7](#_Toc24130)

[4.2 Блок-схема алгоритма 8](#_Toc24131)

[5 Тестирование программы 9](#_Toc24132)

[6 Отладка 10](#_Toc24133)

[7 Совместная разработка 11](#_Toc24134)

[Заключение 13](#_Toc24135)

[Список используемой литературы 14](#_Toc24136)

[Приложение А. Листинг программы 15](#_Toc24137)

# 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.

Быстрая сортировка — является наиболее эффективным алгоритмом сортировки с помощью прямого обмена. Выбирается опорный элемент массива. Элементы меньше опорного располагаются перед ним, больше опорного или равные — после. Данные операции рекурсивно применяются к подмассивам, но только если подмассив не единичный или не пустой.

# 1 Постановка задачи

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

Необходимо использовать сервис GitHub для совместной работы, создать и выложить коммиты, характеризующие действия, выполненные

каждым участником бригады.

Оформить отчет по проведенной практике.

### **Достоинства алгоритма быстрой сортировки**

* Один из самых быстродействующих из алгоритмов внутренней сортировки;
* Алгоритм очень короткий;
* Работает на связных списках, что обеспечивает одновременный доступ к началу и концу массива;

### **1.2 Недостатки алгоритма быстрой сортировки**

* Сильно деградирует по скорости (до O(n2)). Это может случиться при неудачных вводных данных;
* Прямая реализация в виде функции с использованием двух рекурсий может вызвать ошибку переполнения стека;
* Алгоритм неустойчив.

## 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 долгов, 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.

# 3 Описание программы

При запуске программы выводится сообщение, в котором пользователю необходимо ввести название файла, который будет сортироваться

std::cout << "Введите название файла: ";

std::cin >> filename;

После того, как данные были введены, данные из него сортируются и записываются в файл, название которого пользователь указывает вручную.

std::ofstream fout(filename);

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

{

fout << arr[i] << " ";

}

fout.close();

Далее над этими данными выполняется быстрая сортировка , при которой сначала выбирается опорный элемент. Дальше числа перераспределяются таким образом, что элементы, меньшие опорного, помещаются перед ним, а большие или равные - после. Дальше к перераспределенным числам рекурсивно применяется функция быстрой сортировки.

void shellSort(int array[], int n) {

// Выбирается опорный элемент

for (int i = start + 1; i <= end; i++) {

if (arr[i] <= pivot)

count++;

}

// Применяем быструю сортировку сортировку

if (start >= end)

return;

int p = partition(arr, start, end);

quickSort(arr, start, p - 1);

quickSort(arr, p + 1, end);

После этого отсортированный массив записывается в файл, указанный ранее.

std::cout << time << "ms";

std::ofstream fout(filename);

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

{

fout << arr[i] << " ";

}

fout.close();

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

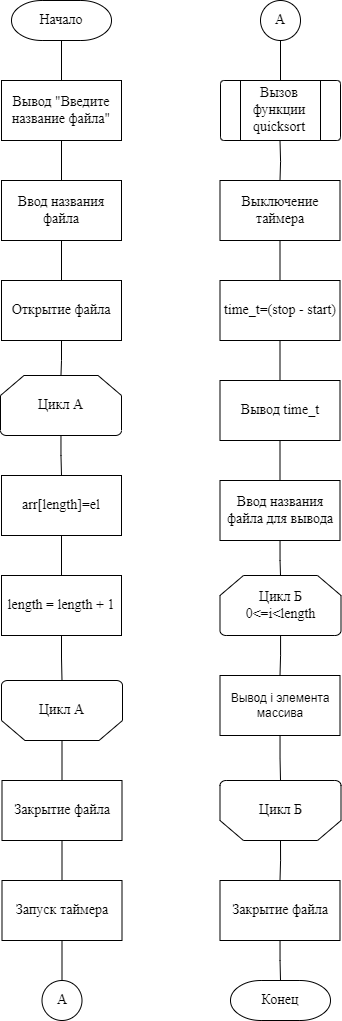


Рисунок 1 **-** Блок-схема программы

## 4.2 Блок-схема алгоритма

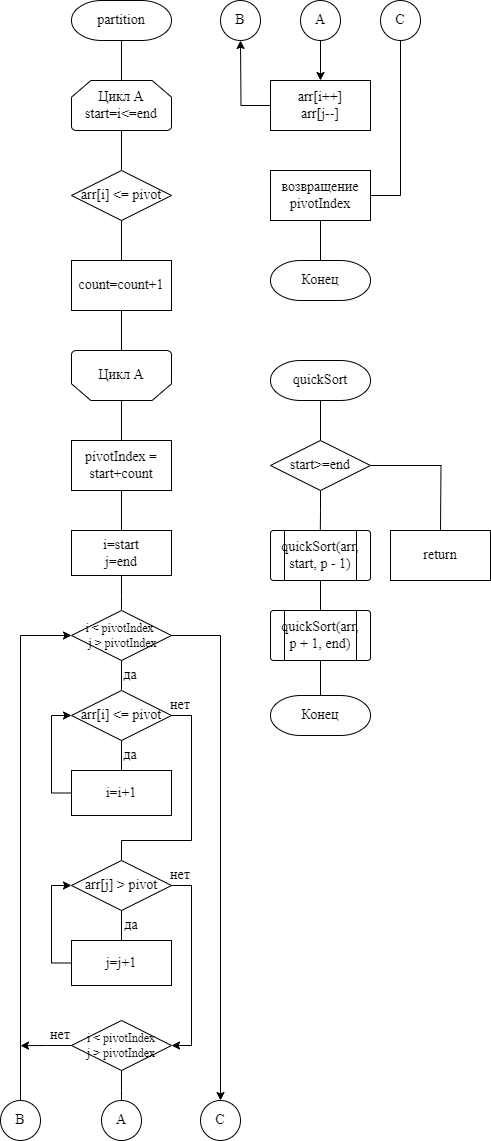


Рисунок 2 - Блок-схема алгоритма с подключенными файлами

# 5 Тестирование программы

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

0

0.2

0.4

0.6

0.8

1

1.2

100000

500000

1000000

1500000

2000000

2500000

3000000

3500000

4000000

Время выполнения сортировки, сек

Время выполнения сортировки, сек

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 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

# 7 Совместная разработка

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

Мною были подключены файлы для записи сгенерированных и отсортированных массивов, это было зафиксировано и загружено на удаленный репозиторий GitHub, на ветку master.

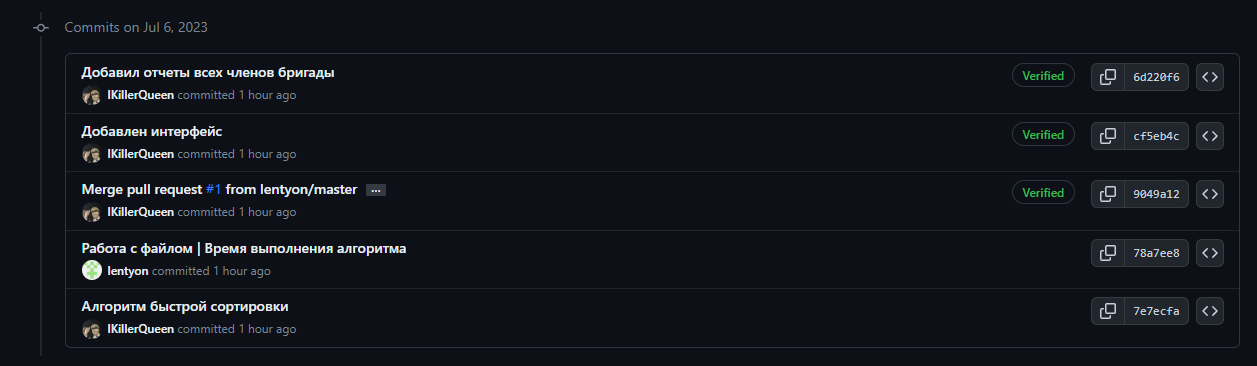


Рисунок 4 – Созданные коммиты

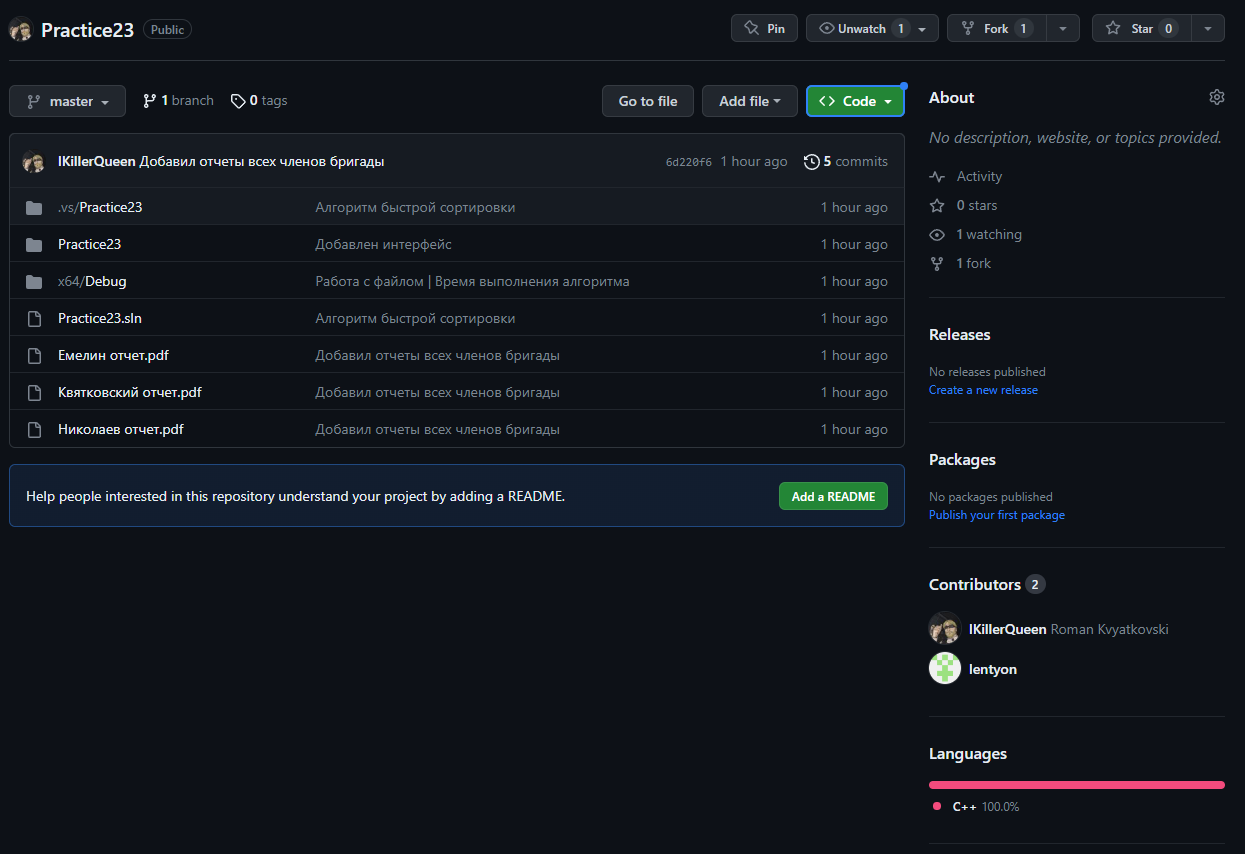


Рисунок 5 – Ветка master

Для загрузки данных на локальный репозиторий, а также отправки данных на удаленный репозиторий было использовано приложение GitHub Desktop.

Ссылка на удаленный репозиторий: https://github.com/lKillerQueen/Practice23

# Заключение

При выполнении данной работы были получены навыки совместной работы с помощью сервисов GitHub и GitHub Desktop. Был изучен алгоритм быстрой сортировки.

Мною была осуществлена разработка алгоритма быстрой сортировки.

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.

# 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. https://ru.wikipedia.org/wiki/Быстрая\_сортировка#Общее\_описание

# Приложение А. Листинг программы

﻿#include <iostream>

#include <stdio.h>

#include <fstream>

#include <time.h>

int partition(int arr[], int start, int end)

{

int pivot = arr[start];

int count = 0;

for (int i = start + 1; i <= end; i++) {

if (arr[i] <= pivot)

count++;

}

int pivotIndex = start + count;

std::swap(arr[pivotIndex], arr[start]);

int i = start, j = end;

while (i < pivotIndex && j > pivotIndex) {

while (arr[i] <= pivot) {

i++;

}

while (arr[j] > pivot) {

j--;

}

if (i < pivotIndex && j > pivotIndex) {

std::swap(arr[i++], arr[j--]);

}

}

return pivotIndex;

}

void quickSort(int arr[], int start, int end)

{

if (start >= end)

return;

int p = partition(arr, start, end);

quickSort(arr, start, p - 1);

quickSort(arr, p + 1, end);

}

int main()

{

std::string filename;

setlocale(LC\_ALL, "");

std::cout << "Введите название файла: ";

std::cin >> filename;

std::ifstream file(filename);

int el;

int length = 0;

int\* arr = NULL;

while (file >> el) {

arr = (int\*)(int\*)realloc(arr, (length + 1) \* sizeof(int)); // Заполнение массива из файла

arr[length] = el;

length += 1;

}

file.close();

time\_t start = clock();

quickSort(arr, 0, length - 1);

time\_t stop = clock();

double time = (stop - start);

std::cout << time << "ms";

std::ofstream fout(filename);

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

{

fout << arr[i] << " "; // Вывод массива в файл

}

fout.close();

int \_;

std::cin >> \_;

return 0;

}