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 (ЭКСПЛУАТАЦИОННОЙ) ПРАКТИКЕ**

(2023/2024 учебный год)

\_\_\_\_ Хабаров Илья Александрович

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 обучения2семестр 4

Период прохождения практики с 20.06.2024 по 17.07.2024

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 (ЭКСПЛУАТАЦИОННОЙ) ПРАКТИКИ**

(2023/2024 учебный год)

                                            Хабаров Илья Александрович

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                  2                 семестр                 4

Период прохождения практики с 20.06.2024 по 17.07.2024

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 звани*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № 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 | 26 | 20.06.24 –  24.06.24 |  |
| 2 | Подбор и изучение материала по теме работы | 26 | 24.06.24 –  26.06.24 |  |
| 3 | Установка виртуальной машины | 26 | 26.06.24 –  28.06.24 |  |
| 4 | Установка операционной системы | 30 | 01.07.24 –  04.07.24 |  |
| 5 | Разработка программы на языке Си | 30 | 04.07.24 –  09.07.24 |  |
| 6 | Тестирование и отладка | 38 | 09.07.24 –  12.07.24 |  |
| 7 | Оформление отчёта | 40 | 12.07.24 –  17.07.24 |  |
|  | **Общий объём часов** | 216 |  |  |

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

**ОТЧЁТ**

**О ПРОХОЖДЕНИИ УЧЕБНОЙ (ЭКСПЛУАТАЦИОННОЙ) ПРАКТИКИ**

(2023/2024 учебный год)

Хабаров Илья Александрович

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 обучения2семестр 4

Период прохождения практики с 20.06.2024 по 17.07.2024

Кафедра «Вычислительная техника»

Хабаров И.А. выполнял практическое задание «Сортировка слияниями». На первоначальном этапе были изучен и проанализирован алгоритм сортировки слияниями, был выбран метод решения и язык программирования С, на котором была написана программа сортировки массива методом слияния. Также, осуществил работу с файлами. Оформил отчёт.

Бакалавр Хабаров И.А.\_\_\_\_\_\_\_\_\_\_\_\_ "\_\_\_" \_\_\_\_\_\_\_\_\_\_\_ 2024 г.

Руководитель Зинкин С.А.\_\_\_\_\_\_\_\_\_\_\_\_ "\_\_\_" \_\_\_\_\_\_\_\_\_\_\_ 2024 г.

практики

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

**ОТЗЫВ**

**О ПРОХОЖДЕНИИ УЧЕБНОЙ (ЭКСПЛУАТАЦИОННОЙ) ПРАКТИКИ**

(2023/2024 учебный год)

Хабаров Илья Александрович

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 обучения2семестр 4

Период прохождения практики с 20.06.2024 по 17.07.2024

Кафедра «Вычислительная техника»

В процессе выполнения практики Прожорин Е.А. решал следующие задачи: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

За период выполнения практики были освоены\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. Во время выполнения работы Прожорин Е. А. 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 Прожорин Е. А. заслуживает оценки «\_\_\_\_\_\_».

Руководитель практики д.т.н., профессор, Зинкин С.А. « » 2024 г.

Оглавление

[Введение 9](#_Toc170761265)

[1 Описание алгоритма 10](#_Toc170761267)

[1.1 Достоинства алгоритма: 10](#_Toc170761268)

[1.2 Недостатки алгоритма 10](#_Toc170761269)

[2 Описание программы 11](#_Toc170761270)

[3 Блок-схемы работы программы 15](#_Toc170761271)

[4 Осуществление командной работы в сфере GitHub 19](#_Toc170761272)

[Заключение 19](#_Toc170761273)

[Список литературы 20](#_Toc170761274)

[Приложение 20](#_Toc170761275)

## **Введение**

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

Функциональная структура среды включает в себя:

* редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего [рефакторинга кода](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D1%84%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3);
* отладчик кода;
* редактор форм, предназначенный для упрощённого конструирования графических интерфейсов;
* веб-редактор;
* дизайнер классов;
* дизайнер схем баз данных.

Visual Studio также позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода, добавление новых наборов инструментов (для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования или инструментов для прочих аспектов процесса разработки программного обеспечения).

Язык Си разрабатывался как язык системного программирования, для которого можно создать однопроходный компилятор. Стандартная библиотека также невелика. Как следствие данных факторов — компиляторы разрабатываются сравнительно легко. Поэтому данный язык доступен на самых различных платформах. К тому же, несмотря на свою низкоуровневую природу, язык ориентирован на переносимость. Программы, соответствующие стандарту языка, могут компилироваться под различные архитектуры компьютеров.

# 1 Описание алгоритма

Основной принцип сортировки слиянием такой: делим массив пополам, каждый из них сортируем слиянием и потом соединяем оба массива. Каждый разделённый массив тоже нарезаем на два подмассива до тех пор, пока в каждом не окажется по одному элементу.

Общая идея алгоритма:

* Сортируемый массив разбивается на две части.
* Каждая из получившихся частей сортируется отдельно тем же самым алгоритмом.
* Разбиение задачи происходит до тех пор, пока размер массива не достигнет единицы.
* Два упорядоченных массива соединяются в один.
* “Слияние” остатка. (Добавляются все оставшиеся элементы второго подмассива, когда один из подмассивов закончился).

# Достоинства алгоритма

Достоинства алгоритма включает в себя:

* Упорядочивание элементов массива (int) при помощи данного алгоритма – это поддержка относительного порядка равных компонентов во входном массиве.
* Метод «разделяй и властвуй» является достаточно простым и понятным.

# Недостатки алгоритма

Недостатки алгоритма включает в себя:

* Реализация рассматриваемого алгоритма требует дополнительной памяти, выделяемой для хранения объединенных подмассивов в процессе работы метода.

# Описание программы

В программе для сортировки слиянием подключены следующие заголовочные файлы *stdio.h*– заголовочный файл с классами, функциями и переменными для организации ввода-вывода; *Windows.h*– специфичный заголовочный файл, необходимый для использования в программе функционала, предоставляемого операционной системой Windows;*time.h*– заголовочный файл стандартной библиотеки C++, содержащий типы и функции для работы с датой и временем.

#pragmawarning(disable : 4996)

#include"menu.h"

#include<time.h>

#include<conio.h>

#include<stdio.h>

#include<Windows.h>

Далее для осуществления сохранения начальных и конечных данных использовали функции save1 и save2.

Функция *save1* сохраняет исходный набор случайных чисел.

void save1(FILE\* F1, char\* F1name, int\* A, int size)

{

F1 = fopen(F1name, "w");

if (F1 != NULL)

{

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

{

fprintf(F1, "%d\n", A[i]);

}

}

else

{

printf("Ошибка записи в файл");

}

fclose(F1);

}

Функция *save2* осуществляет сохранение отсортированного массива чисел.

void save2(FILE\* F2, char\* F2name, int\* A, intsize)

{

F2 = fopen(F2name, "w");

if (F2 != NULL)

{

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

{

fprintf(F2, "%d\n", A[i]);

}

}

else

{

printf("Ошибка записи в файл");

}

fclose(F2);

}

Функцией *random* создается массив из случайного набора чисел, который сохраняется в файл с произвольным названием. После чего происходит сортировка массива и записывается в новый файл.

Void random(void)

{

int size;

FILE\* F1, \* F2;

char F1name[20];

char F2name[20];

printf("Введите файл, в котором будут находиться исходные числа: ");

gets\_s(F1name);

F1 = fopen(F1name, "w");

printf("Введите файл, в котором будут находиться отсортированные числа: ");

gets\_s(F2name);

F2 = fopen(F2name, "w");

printf("Введите размер массива (больше 1): ");

do

{

scanf\_s("%d", &size);

if (size< 2)

{

printf("Неверно ввёден размер массива\n");

printf("Введите размер массива (больше 1): ");

}

} while (size < 2);

int\* A = newint[size];

for (int i = 0; i <size; i++) //Заполнение массива А случайными числами

{

A[i] = (rand() % 20001) - 10000;

}

printf("Исходныечиславфайле \"%s\" \n", F1name);

save1(F1, F1name, A, size);

time\_t start = clock();

Merge\_Sort(A, 0, (size - 1));

time\_t stop = clock();

double time = (stop - start) / 1000.0;

printf("Время сортировки: %f секунд\n", time);

save2(F2, F2name, A, size);

printf("Отсортированные числа в файле \"%s\" \n", F2name);

}

# Блок-схемы работы программы

На рисунке 1 и 2 показана блок схема сохранения начальных и конечных массивов

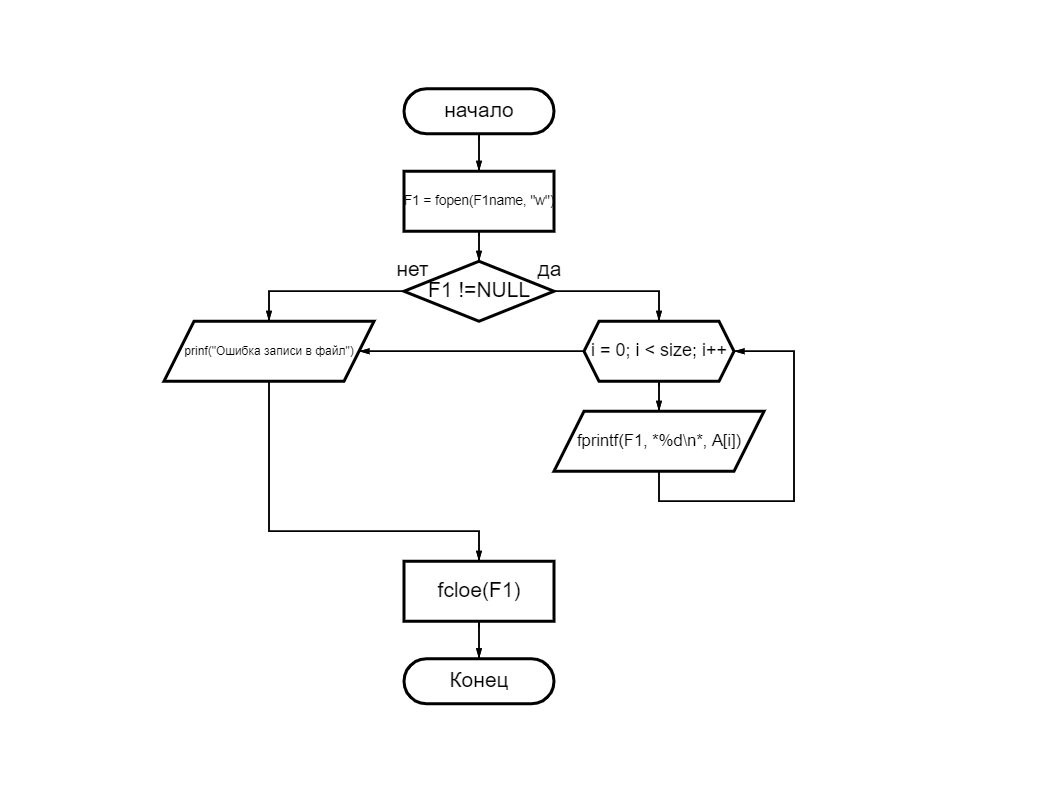
****

Рисунок 1 – Блок-схема функции save1

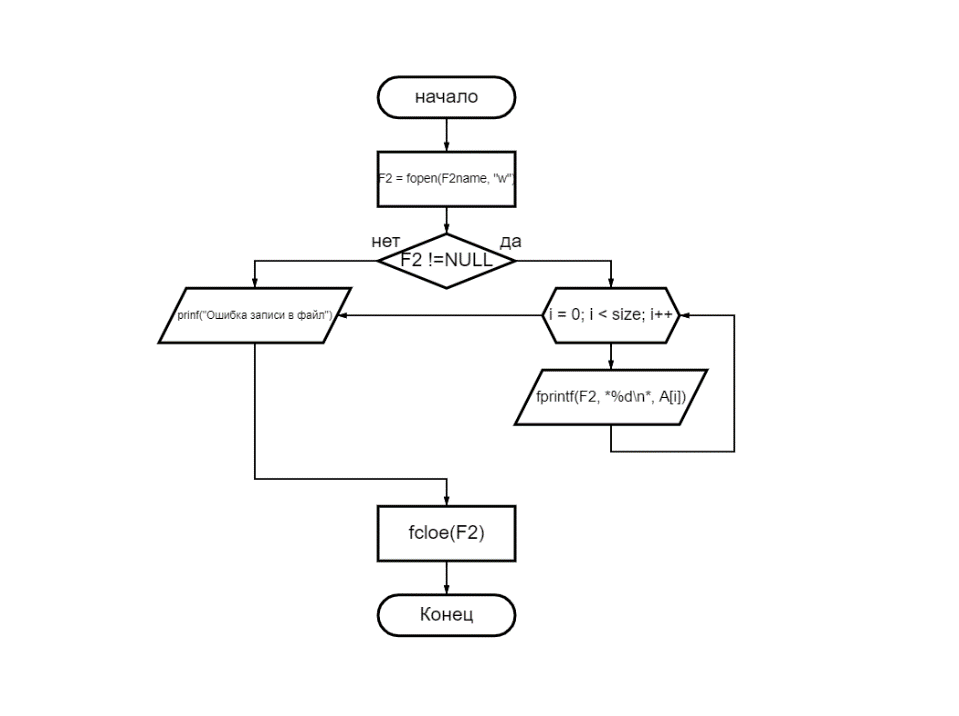
****

Рисунок 2 – Блок-схема функции save2

На рисунке 3 представлена главная блок схема функции random.

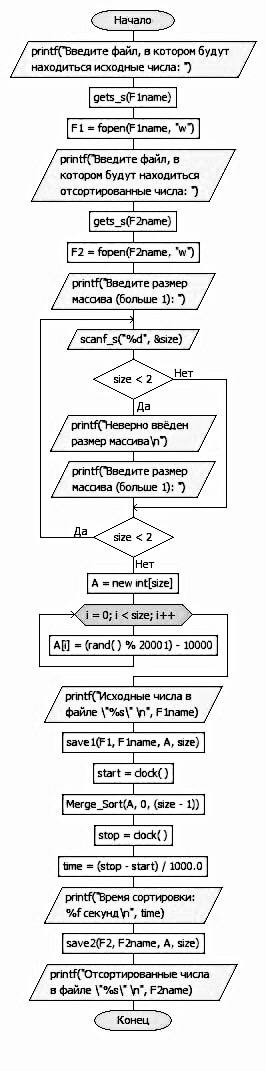
****

Рисунок 3 – Блок-схема функции random

# Осуществление командной работы в сфере GitHub

GitHub — это многофункциональная платформа для командной работы над проектом, состоящая из:

* облачного хранилища, в котором размещаются все файлы;
* социальной сети для разработчиков;
* системы контроля версий, фиксирующей все редакции, вносимые в код тем или иным участником.

Простыми словами GitHub — площадка, где можно разместить общий проект, совместно управлять всеми изменениями, а в случае неудачи — быстро вернуться к исправной версии.

Действия, совершённые мной в GitHub:

1. Создал репозиторий и пригласил членов бригады
2. Создал проект Microsoft Visual Studio и загрузил его в репозиторий
3. Загрузил блок-схемы своего кода
4. Загрузил отчёт практики

# Заключение

В результате проделанной работы была разработана программа, с помощью которой, можно отсортировать числа в порядке возрастания используя метод сортировки слиянием. Были приведены достоинства и недостатки этого кода. Для каждой функции была создана блок схема. А так же была осуществлена работа в бригаде в сфере GitHub.

# Список литературы

1. Википедия. Сортировка слиянием. URL: <https://en.wikipedia.org/wiki/Merge_sort>
2. Дейтел и Пол Харви. Как программировать на С. Интернет-магазин Издательства Бином: [www.binom-press.ru](http://www.binom-press.ru)
3. Никлаус Вирт. Алгоритмы и структуры данных. Новая версия для Оберона + CD / Пер. с англ. Ткачев Ф.В.

# Приложение

**Листинг программы**

#pragmawarning(disable : 4996)

#include"menu.h"

#include<time.h>

#include<conio.h>

#include<stdio.h>

#include<Windows.h>

void Merge(int\* Ap, intp, intq, intr)

{

int n1 = q - p + 1;

int n2 = r - q;

int\* L; int\* R;

L = newint[n1 + 1];

R = newint[n2 + 1];

for (int i = 1; i <= n1; i++)

{

L[i-1] = Ap[p + i - 1];

}

for (int j = 1; j <= n2; j++)

{

R[j-1] = Ap[q + j];

}

for (int i = 0, j = 0, k = p; k <= r; k++) {

if (i < n1 && j < n2) {

if (L[i] <= R[j]) {

Ap[k] = L[i];

i++;

}

else {

Ap[k] = R[j];

j++;

}

}

elseif (i < n1) {

Ap[k] = L[i];

i++;

}

else {

Ap[k] = R[j];

j++;

}

}

}

void Merge\_Sort(int\* Ap, intp, intr)

{

int q;

if (p<r)

{

q = (p + r) / 2;

Merge\_Sort(Ap, p, q);

Merge\_Sort(Ap, q + 1, r);

Merge(Ap, p, q, r);

}

}

void save1(FILE\* F1, char\* F1name, int\* A, intsize)

{

F1 = fopen(F1name, "w");

if (F1 != NULL)

{

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

{

fprintf(F1, "%d\n", A[i]);

}

}

else

{

printf("Ошибка записи в файл");

}

fclose(F1);

}

void save2(FILE\* F2, char\* F2name, int\* A, intsize)

{

F2 = fopen(F2name, "w");

if (F2 != NULL)

{

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

{

fprintf(F2, "%d\n", A[i]);

}

}

else

{

printf("Ошибка записи в файл");

}

fclose(F2);

}

void random(void)

{

int size;

FILE\* F1, \* F2;

char F1name[20];

char F2name[20];

printf("Введите файл, в котором будут находиться исходные числа: ");

gets\_s(F1name);

F1 = fopen(F1name, "w");

printf("Введите файл, в котором будут находиться отсортированные числа: ");

gets\_s(F2name);

F2 = fopen(F2name, "w");

printf("Введите размер массива (больше 1): ");

do

{

scanf\_s("%d", &size);

if (size < 2)

{

printf("Неверно ввёден размер массива\n");

printf("Введите размер массива (больше 1): ");

}

} while (size < 2);

int\* A = newint[size];

for (int i = 0; i < size; i++) //Заполнение массива А случайными числами

{

A[i] = (rand() % 20001) - 10000;

}

printf("Исходные числа в файле \"%s\" \n", F1name);

save1(F1, F1name, A, size);

time\_t start = clock();

Merge\_Sort(A, 0, (size - 1));

time\_t stop = clock();

int time = (stop - start) / 1000;

printf("Время сортировки: %i мс\n", time);

save2(F2, F2name, A, size);

printf("Отсортированные числа в файле \"%s\" \n", F2name);

}

void sort(void)

{

FILE \*F3,\* F4, \*F5, \*F6;

char F3name[20], F4name[20];

printf("Введите название файла: ");

gets\_s(F3name);

F3 = fopen(F3name, "r");

if (F3 == NULL)

{

printf("Невозможно открыть файл \"%s\"\n", &F3name);

}

else

{

int size = 0;

int curDigit = 0;

while (fscanf(F3, "%d", &curDigit) != EOF) {

size++;

}

fseek(F3, 0, SEEK\_SET);

int\* A = newint [size];

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

{

fscanf(F3, "%d", &A[i]);

}

fclose(F3);

time\_t start = clock();

Merge\_Sort(A, 0, size - 1);

time\_t stop = clock();

int time = (stop - start) / 1000;

printf("Время сортировки: %i мс\n", time);

printf("Напишите, в какой файл сохранить отсортированные файлы (текстовый файл!): ");

gets\_s(F4name);

F4 = fopen(F4name, "w");

if (F4 != NULL)

{

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

{

fprintf(F4, "%d\n", A[i]);

}

}

else

{

printf("Ошибка записи в файл");

}

fclose(F4);

}

}

int main(void)

{

SetConsoleCP(1251); //русификация

SetConsoleOutputCP(1251);

srand(time(NULL));

int s = 1;

while (s != 0)

{

menu();

s = \_getch();

switch (s)

{

case'1': system("cls"); sort(); break;

case'2': system("cls"); random(); break;

case'3': system("cls"); exit(0); break;

default: system("cls");

}

}

return 0;

}