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)

Володин Артём Павлович

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 25.06.2024 по 08.07.2024

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., 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

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

(2023/2024 учебный год)

Володин Артём Павлович

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 25.06.2024 по 08.07.2024

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., 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 | 2 | 25.06.24 –  25.06.24 |  |
| 2 | Подбор и изучение материала по теме  работы | 15 | 26.06.24 –  28.06.24 |  |
| 3 | Разработка  алгоритма | 43 | 28.06.24 –  02.07.24 |  |
| 4 | Описание алгоритма и  программы | 18 | 02.07.24 – 04.07.24 |  |
| 5 | Тестирование | 5 | 04.07.24 –  04.07.24 |  |
| 6 | Получение и  анализ результатов | 10 | 04.07.24 –  06.07.24 |  |
| 7 | Оформление  отчёта | 15 | 06.07.24 –  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)

Володин Артём Павлович

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 25.06.2024 по 08.07.2024

Кафедра «Вычислительная техника»

Володин А. П. выполнял практическое задание «Сортировка пузырьком». На первоначальном этапе были изучен и проанализирован алгоритм сортировки пузырьком, был выбран метод решения и язык программирования С, на котором была написана программа сортировки массива методом пузырька. Отладил программу. Оформил отчёт.

Бакалавр Володин А. П. " " 2024 г.



Руководитель Карамышева Н.С.

практики

" " 2024 г.

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)

Володина Артёма Павловича

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 25.06.2023 по 08.07.2023

Кафедра «Вычислительная техника»

В процессе выполнения практики Володин А. П. решал следующие задачи: создание алгоритма пузырьковой сортировки, анализ работы алгоритма, сравнение существующих методов сортировки.

За период выполнения практики были освоены основные понятия и технологии сортировки пузырьком. Во время выполнения работы Володин А. П. 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 Володин А. П. заслуживает оценки « ».

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

**Содержание**

[Введение 7](#_Toc170593459)

[1 Постановка задачи 8](#_Toc170593460)

[2 Выбор решения 9](#_Toc170593461)

[3 Описание программы 10](#_Toc170593462)

[4 Схемы программы 12](#_Toc170593463)

[4.1 Блок-схема программы 12](#_Toc170593464)

[4.2 Блок-схема алгоритма 13](#_Toc170593465)

[5 Тестирование программы 15](#_Toc170593466)

[6 Отладка 16](#_Toc170593467)

[7 Совместная разработка 17](#_Toc170593468)

[Заключение 22](#_Toc170593469)

[Список используемой литературы 23](#_Toc170593470)

[Приложение А. Листинг программы 24](#_Toc170593471)

# 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 очень широко распространяются практически во всех задачах обработки информации. Они образуют отдельный класс алгоритмов, применяются с целью осуществления последующего более быстрого поиска.

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 пузырьком является самым простейшим для понимания и реализации. Алгоритм не самый лучший с точки зрения производительности, но является очень эффективным для сортировки небольших массивов. Сортировка пузырьком также используется в быстрой сортировке, которая сочетает в себе различные алгоритмы для повышения производительности.

# Постановка задачи

Поставленная задача: необходимо заполнить массив из n-ого количества элементов случайными числами, записать данные элементы в отдельный файл. После этого выполнить сортировку пузырьком 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. **Достоинства алгоритма сортировки вставками**
     + алгоритм удобен для работы с массивами небольшого размера;
     + алгоритм не требует дополнительных массивов;
     + простая реализация алгоритма.
  2. **Недостатки алгоритма сортировки вставками**
     + очень много перемещений элементов массива;
     + самый медленный способ сортировки;
     + не рекомендуется для сортировки больших массивов.
  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);
     + астрономические объекты (масса, размеры, плотность).

# Выбор 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. Для удобства совместной разработки был использован сервис WEEK. WEEEK — сервис для управления личными и командными проектами. В основе WEEEK 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 WEEEK Week.

# Описание программы

При запуске программы выводится меню из четырёх пунктов:

1. сортировка случайных чисел;

2. сортировка введенных чисел;

3. информация;

4. ESC – выход.

Пользователю требуется выбрать тот пункт, который ему требуется. При выборе пунктов под цифрами 1-4 выводится сообщение, в котором пользователю необходимо ввести количество значений для сортировки.

printf("Введите количество элементов: "); scanf("%d", &size);

После того, как данные были введены, генерируется массив из случайных чисел.

array = (int\*)malloc(size \* sizeof(int)); for (int i = 0; i < size; i++)

{

array[i] = rand() %100;

}

Эти числа записываются в файл original\_numbers.txt.

Далее над этими данными выполняется сортировка пузырьком, при которой массив постепенно перебирается слева направо. При этом элемент сравнивается со следующим элементом и если левый элемент больше правого, то они меняются местами. Так происходит до тех пор, пока набор входных данных не будет отсортирован по возрастанию.

Алгоритм для сортировки пузырьком:

void bubbleSort(int array[], int size) {

for (int i = 0; i < size - 1; i++) {

for (int j = 0; j < size - i - 1; j++) {

if (array[j] > array[j + 1]) {

int tmp = array[j];

array[j] = array[j + 1];

array[j + 1] = tmp;

}

}

}

}

После этого отсортированный массив записывается в файл sorted\_numbers.txt.

При выборе пункта меню под цифрой 4 программа завершает выполнение.

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

Листинг программы приведен в приложении А.

# Схемы программы

## Блок-схема программы

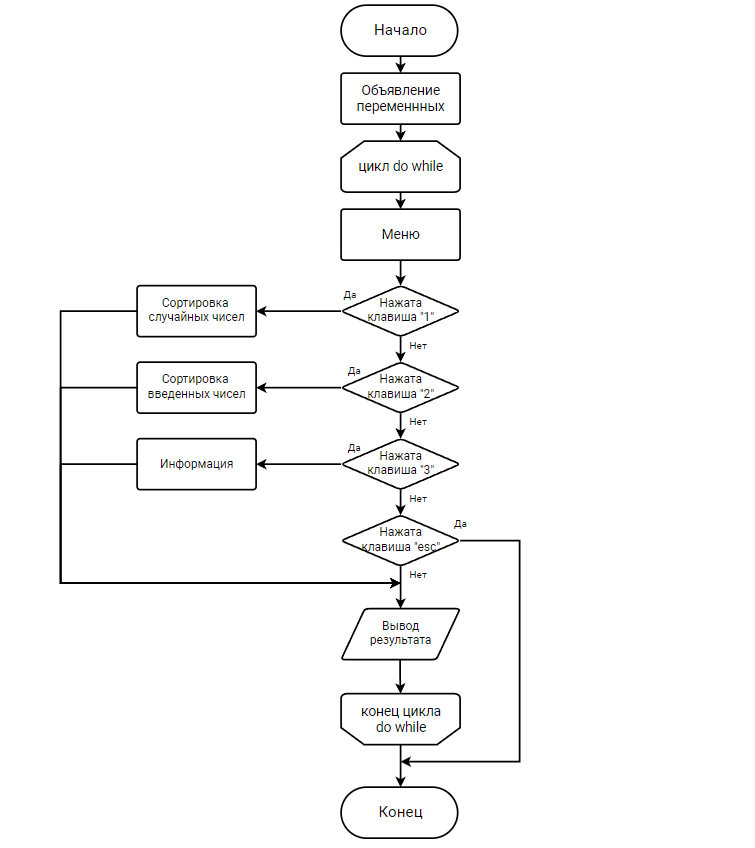


Рисунок 1 – Блок-схема программы

## Блок-схема алгоритма

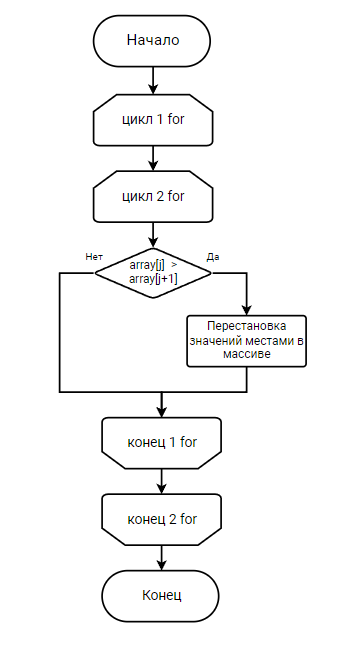


Рисунок 2 – Блок-схема алгоритма

# Тестирование программы

Тестирование показало, что с увеличением количества элементов пропорционально увеличивается время работы программы. График зависимости времени выполнения сортировки от количества элементов в наборе приведен на рисунке 3.

Рисунок 3 – Результаты тестирования

# 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.

# Совместная разработка

Для удобства совместной разработки был использован сервис WEEK. Определили задачи проекта, назначили приоритет задачам.

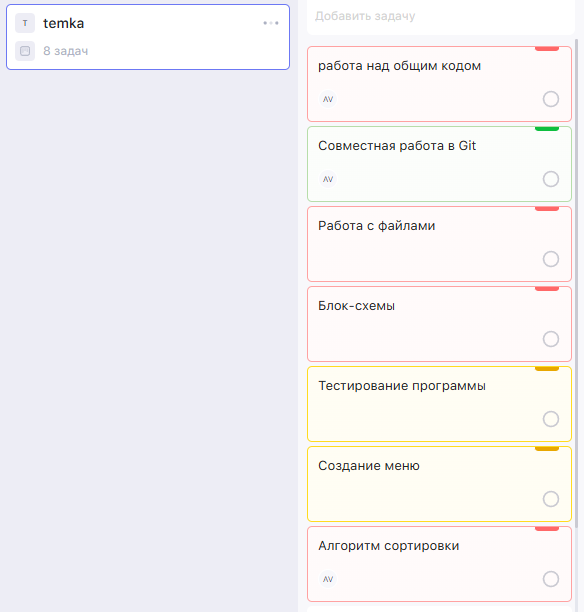


Рисунок 4 – Определение задач проекта

Распределили роли, назначили исполнителей задачам.

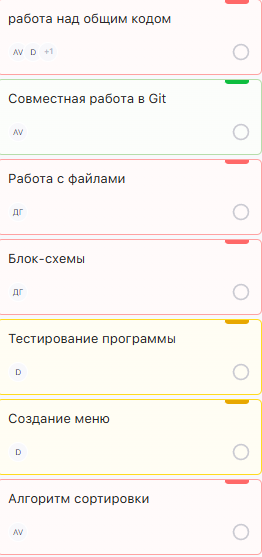


Рисунок 5 – Распределение задач проекта

Обсуждали выполнение задачи на канбан-доске.

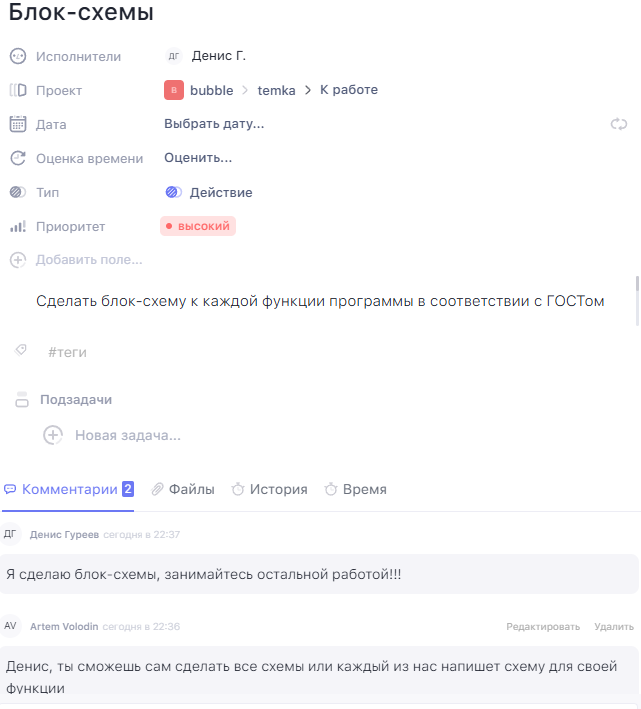


Рисунок 6 – Обсуждение задач проекта

Корректировали статус задач по мере выполнения.

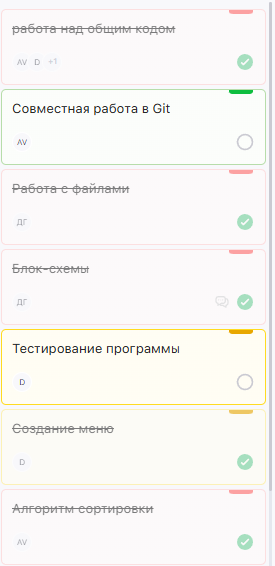


Рисунок 7 – Корректирование задач

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

Мною был написан алгоритм сортировки, она была загружена на удаленный репозиторий Github, на ветку master.

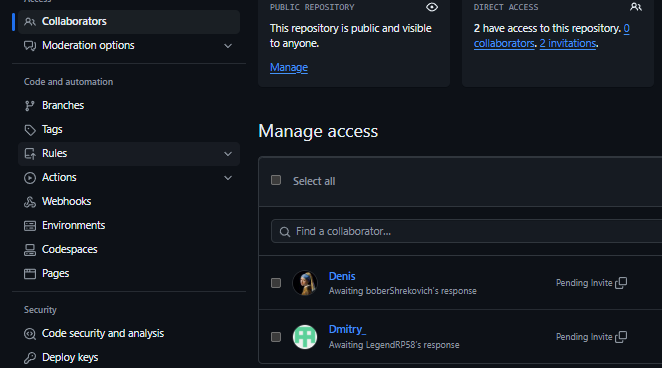


Рисунок 8 – Создание общего репозитория



Рисунок 9 –Инициализация репозитория

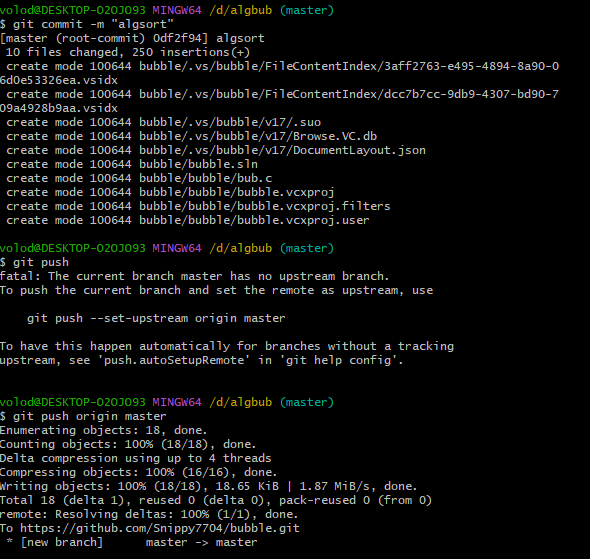


Рисунок 10 – Загрузка алгоритма на удалённый репозиторий

Ссылка на удаленный репозиторий:

<https://github.com/Snippy7704/bubble.git>

# 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 пузырьком для массива случайно сгенерированных и написанных вручную 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.

# 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 пузырьком [Электронный ресурс] – URL: https://ru.wikipedia.org (дата обращения: 01.07.2024 г)

# Приложение А. Листинг программы

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdlib.h>

#include <stdio.h>

#include <conio.h>

#include <locale.h>

#include <time.h>

void bubbleSort(int array[], int size) {

for (int i = 0; i < size - 1; i++) {

for (int j = 0; j < size - i - 1; j++) {

if (array[j] > array[j + 1]) {

int tmp = array[j];

array[j] = array[j + 1];

array[j + 1] = tmp;

}

}

}

}

void writeArray(int array[], int size, const char\* filename) {

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

if (file != NULL) {

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

fprintf(file, "%d ", array[i]);

}

fclose(file);

}

else {

printf("Ошибка открытия файла");

}

}

void printArray(int array[], int size) {

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

printf("%d ", array[i]);

}

}

int main() {

setlocale(LC\_ALL, "");

srand(time(NULL));

int size;

int\* array;

char ch;

do {

system("cls");

printf("\n");

printf(" |><><><><><><>МЕНЮ<><><><><><><|\n");

printf(" 1. Сортировка случайных чисел\n");

printf(" 2. Сортировка введенных чисел\n");

printf(" 3. Информация\n");

printf(" (ESC). Выход\n");

ch = \_getch();

switch (ch) {

case '1':

system("cls");

printf("Введите количество элементов: ");

scanf("%d", &size);

array = (int\*)malloc(size \* sizeof(int));

//Заполнение рандомными числами

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

array[i] = rand() % 100; // от 0 до 100

}

//Запись исходного массива в файл

writeArray(array, size, "original\_numbers.txt");

printf("\nИсходный массив: ");

printArray(array, size);

//Сортировочка с учетом времени работы

time\_t start = clock();

bubbleSort(array, size);

time\_t stop = clock();

double time = (stop - start) / 1000.0;

//Запись отсортированного массива в файл

writeArray(array, size, "sorted\_numbers.txt");

printf("\nОтсортированный массив: ");

printArray(array, size);

printf("\n\nВремя выполнения сортировки: %lf секунд", time);

printf("\n\n\n");

system("pause");

free(array);

break;

case '2':

system("cls");

printf("Введите количество элементов: ");

scanf("%d", &size);

array = (int\*)malloc(size \* sizeof(int));

//Заполнение массива вручную

printf("Введите %d чисел: ", size);

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

scanf("%i", &array[i]);

}

//Запись исходного массива в файл

writeArray(array, size, "original\_numbers.txt");

printf("\nИсходный массив: ");

printArray(array, size);

//Сортировочка с учетом времени работы

time\_t start1 = clock();

bubbleSort(array, size);

time\_t stop1 = clock();

double time1 = (stop1 - start1) / 1000.0;

//Запись отсортированного массива в файл

writeArray(array, size, "sorted\_numbers.txt");

printf("\nОтсортированный массив: ");

printArray(array, size);

printf("\n\nВремя выполнения сортировки: %lf секунд", time1);

printf("\n\n\n");

system("pause");

free(array);

break;

case '3':

system("cls");

printf("\*----------------------------------Информация-----------------------------------\*\n");

printf("Выполнили студенты группы 23ВВВ3\n");

printf("Артем Володин (Реализация алгоритма сортировки,создание общего Git-репозитория)\n");

printf("Батайкин Дмитрий (Тестирование программы, разработка меню)\n");

printf("Гуреев Денис (Работа с файлами, финальная компоновка программы)\n");

printf("\n");

system("pause");

break;

}

} while (ch != 27);

return 0;

}