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)

                                           Назаров Данила Андреевич

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 – очнаяСрок обучения в соответствии с ФГОС – 5лет

Год обучения1семестр 2

Период прохождения практики с 06.02.2023 по 15.07.2023

Кафедра «Вычислительная техника»

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

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

Руководитель практикид.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)

                                           Назаров Данила Андреевич

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 06.02.2023 по 15.07.2023

Кафедра «Вычислительная техника»

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

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

Руководитель практикид.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 (распределенная практика) | 26 | 06.02.23 –  10.03.23 |  |
| 2 | Подбор и изучение материала по теме работы (распределенная практика) | 26 | 10.03.23–  10.04.23 |  |
| 3 | Разработка алгоритма (распределенная практика) | 26 | 10.04.23 –  08.06.23 |  |
| 4 | Описание алгоритма и программы  (сосредоточенная практика) | 30 | 26.06.23 –  30.07.23 |  |
| 5 | Тестирование  (сосредоточенная практика) | 30 | 30.07.23 –  06.07.23 |  |
| 6 | Получение и анализ результатов  (сосредоточенная практика) | 38 | 06.07.23–  12.07.23 |  |
| 7 | Оформление отчёта  (сосредоточенная практика) | 40 | 12.07.23 –  15.07.23 |  |
|  | **Общий объём часов** | 216 |  |  |

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)

                                           Назаров Данила Андреевич

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

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

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

Год обучения1семестр 2

Период прохождения практики с 06.02.2023 по 15.07.2023

Кафедра «Вычислительная техника»

Назаров Д.А. выполнял практическое задание «Сортировка выбором». На первоначальном этапе были изучен и проанализирован алгоритм выборной сортировки, был выбран метод решения и язык программирования С++, на котором была написана программа сортировки массива методом выбора. Также, осуществил работу с файлами. Оформил отчёт.

Бакалавр Назаров Д.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)

                                        Назаров Данила Андреевич

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 06.02.2023 по 15.07.2023

Кафедра «Вычислительная техника»

В процессе выполнения практики Назаров Д.А. решал следующие задачи: разработка алгоритма сортировки выбором, анализ результатов сортировки,

За период прохождения практики были освоены основные понятия и технологии программирования, изучены основные инструменты языков С/С++. И получены следующие результаты: разработан алгоритм сортировки выбором, получены результаты работы алгоритма, сделаны выводы эффективности работы алгоритма сортировки выбором. Во время выполнения работы Назаров Д.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 Назаров Д.А. заслуживает оценки «\_\_\_\_\_\_».

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

Оглавление

[Введение 2](#_Toc138949249)

[1. Постановка задачи 3](#_Toc138949250)

1.1 Достоинства алгоритма……………………………………………………….3

1.2Недостатки алгоритма……..………………………………………………….3

1.3 Типичные сценарии применения……………………………………………3

[2. Выбор решения 4](#_Toc138949251)

[3. Описание программы 5](#_Toc138949252)

[4. Схемы программы 6](#_Toc138949253)

4.1 Блок-схема алгоритма………………………………………………………...6

4.2Блок-схема программы…………………………………………………..….10

[5. Тестирование программы 14](#_Toc138949254)

5.1 Тестирование на разных наборах данных………………………………….14

5.2Анализ полученных результатов тестирования…………………………...15

[6. Отладка 16](#_Toc138949255)

[7. Совместная разработка 17](#_Toc138949256)

[Заключение 19](#_Toc138949257)

[Список используемой литературы 20](#_Toc138949258)

[Приложение А 21](#_Toc138949259)

[Приложение Б “Листинг” 24](#_Toc138949260)

**Введение**

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

Функциональная структура среды включает в себя:

1)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);

2)отладчик кода;

3)редактор форм, предназначенный для упрощённого конструирования графических интерфейсов;

4)веб-редактор;

5)дизайнер классов;

6)дизайнерсхем баз данных.

VisualStudio 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).

В комплект входят следующие основные компоненты:

1. VisualBasic.NET – для разработки приложений на VisualBasic;
2. VisualC++ – на традиционном языке C++;
3. VisualC# – наязыкеC# (Microsoft);
4. VisualF# – наF# (Microsoft Developer Division).

Интегрированная среда разработки (IntegratedDevelopmentEnvironment – IDE) VisualStudio предлагает ряд высокоуровневых функциональных возможностей, которые выходят за рамки базового управления кодом.

Ниже перечислены основные преимущества IDE-среды VisualStudio:

1. Встроенный Web-сервер;
2. Поддержка множества языков при разработке;
3. Интуитивный стиль кодирования;
4. Более высокая скорость разработки;
5. Возможности отладки.

Язык программирования С++ представляет высокоуровневый компилируемый язык программирования общего назначения со статической типизацией, который подходит для создания самых различных приложений. На сегодняшний день С++ является одним из самых популярных и распространенных языков.

Своими корнями он уходит в язык Си, который был разработан в 1969—1973 годах в компании BellLabs программистом Деннисом Ритчи. В начале 1980-х годов датский программист Бьерн Страуструп, который в то время работал в компании BellLabs, разработал С++ как расширение к языку Си. Фактически вначале C++ просто дополнял язык Си некоторыми возможностями объектно-ориентированного программирования. И поэтому сам Страуструп вначале называл его как "C withclasses" ("Си с классами").

С++ является мощным языком, унаследовав от Си богатые возможности по работе с памятью. Поэтому нередко С++ находит свое применение в системном программировании, в частности, при создании операционных систем, драйверов, различных утилит, антивирусов и т.д. К слову сказать, ОС Windows большей частью написана на С++. Но только системным программированием применение данного языка не ограничивается. С++ можно использовать в программах любого уровня, где важны скорость работы и производительность. Нередко он применяется для создания графических приложений, различных прикладных программ. Также особенно часто его используют для создания игр с богатой насыщенной визуализацией. Кроме того, в последнее время набирает ход мобильное направление, где С++ тоже нашел свое применение. И даже в веб-разработке также можно использовать С++ для создания веб-приложений или каких-то вспомогательных сервисов, которые обслуживают веб-приложения. В общем С++ –язык широкого пользования, на котором можно создавать практически любые виды программ.

**1 Постановка задачи**

По программе: Необходимо заполнить массив из n-ого количества элементов случайными числами. Записать неотсортированный массив в файл. После этого выполнить сортировку выбором над данными, находящимися в массиве, записать отсортированные данные в файл.

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

**1.1 Типичные сценарии применения:**

Сортировка результатов спортивного соревнования.

Сортировка списка задач по срокам выполнения.

Сортировка массива студентов по их оценкам.

**1.2 Достоинства алгоритма быстрой сортировки:**

* Один из самых простых в реализации метод.
* Требует лишь O(log⁡ n) дополнительной памяти для своей работы. O(n)
* Осуществляет стабильную сортировку данных.
* Нетрудно анализировать по сравнению с другими алгоритмами сортировки.

**1.3 Недостатки алгоритма:**

* Сильно деградирует по скоростиO(n2) .
* Прямая зависимость от того, упорядочен ли исходный массив.O(n)

# 2 Выбор решения

Нашей бригадой было выбрано вести разработку в среде MicrosoftVisualStudio на языке C/C++.

Все действия, совершаемые программой, происходят в 2 файлах.

Сортировка массива выбором осуществляется так: массив делится на две части. Одна из частей называется отсортированной, а другая неотсортированной. Алгоритм предполагает проход по всему массиву с тем, чтобы длина отсортированной части стала равна длине всего массива. В рамках каждой итерации мы находим минимум в неотсортированной части массива и меняем местами этот минимум с первым элементом неотсортированной части массива. После чего мы увеличиваем длину отсортированной части массива на единицу.

Общая идея алгоритма состоит в следующем:

* Выбрать из массива первый элемент;
* Найти минимальный элемент массива и запомнить его;
* Если номер первого элемента и номер найденного элемента не совпадают, тогда два этих элемента обмениваются значениями, иначе никаких манипуляций не происходит;
* Увеличиваем номер элемента на 1 и продолжаем сортировку оставшейся части массива.

**3 Описание программы**

В программе для сортировки выбором подключены следующие заголовочные файлы:*stdio.h* – заголовочный файл с классами, функциями и переменными для организации ввода-вывода; *stdlib.h*–содержит в себе функции, занимающиеся выделением памяти;***locale.h***– **заголовочный файл,** стандартной библиотеки, используемый для задач локализации;*time.h*– заголовочный файл стандартной библиотеки C++, содержащий типы и функции для работы с датой и временем.

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

#include<locale.h>

Далее идет текст функции сортировки выбором, которая сортируют поданный ей массив:

// сортировкавыбором

Void selectionSort(intarr[], intn) {

inti, j, minIndex, temp;

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

minIndex = i;

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

if (arr[j] <arr[minIndex]) {

minIndex = j;

}

}

temp = arr[minIndex];

arr[minIndex] = arr[i];

arr[i] = temp;

}

}

Функция rand\_massive заполняет массива случайными числами:

Int rand\_massive(intarr[], intn) {

inti;

srand(time(0));

for (i = 0; i<n; i++) {

arr[i] = rand() % 100;

}

return 0;

}

Затем идёт текст функции для записи сгенерированного массива в файл:

in t mark\_rand\_file(intarr[], intn) {

int i;

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

if (filer != NULL) {

for (i = 0; i<n; i++) {

fprintf(filer, "%d \n", arr[i]);

}

fclose(filer);

}

return 0;

}

Потом идёт текст функции для записи сортированного массива в файл:

in t mark\_sort\_file(intarr[], intn) {

int i;

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

if (file != NULL) {

for (i = 0; i<n; i++) {

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

}

fclose(file);

}

return 0;

}

Далее располагается основная функция в которой будут вызваны предыдущие функции для метода сортировки выбором, а так же будет рассчитано время работы программы и произведена локализация платформы на русский язык:

int main() {

int\* arr; // указатель на int для массива

inti, n;

clock\_t timer;

setlocale(LC\_ALL, "RUS");

n = 0;

while ((n < 10000) or (n > 1100000)) {

printf("Введите размер массива от 10000 до 1100000: ");

scanf("%d", &n);

}

// Выделениепамяти

arr = (int\*)malloc(n \* sizeof(int));

// Заполнение массива случайными числами

rand\_massive(arr, n);

// запись рандом массива в файл

mark\_rand\_file(arr, n);

// Сортировкамассива

timer = clock();

selectionSort(arr, n);

timer = clock() - timer;

printf("Time sort = ");

printf("%f", (double)timer / CLOCKS\_PER\_SEC);

// Запись сортированного массива в файл

mark\_sort\_file(arr, n);

return 0;

}

**4 Схемы программы**

**4.1 Блок схема алгоритма сортировки выбором**

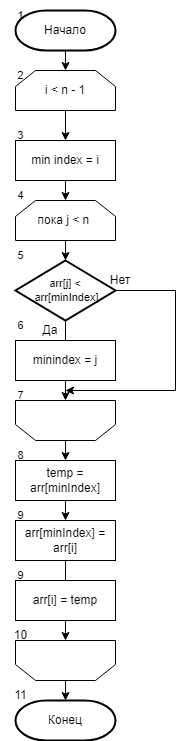


Рисунок 1 “Блок схема алгоритма сортировки массива выбором”

**4.2 Блок схема алгоритма заполнения массива случайными числами**

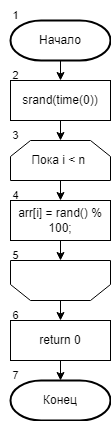


Рисунок 2 “Блок схема алгоритмазаполнения

массива случайными числами ”

**4.3 Блок схема программы**

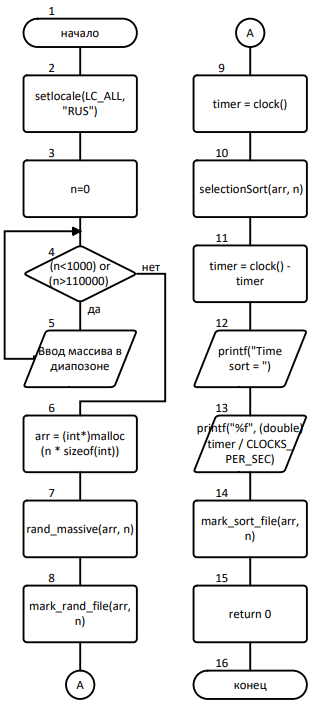
****

Рисунок 3“Блок схема программы”

* 1. **Тестирование программы**

**5.1 Тестирование на разных наборах данных**

Тестовый набор данных представлен в таблице 1. Результаты тестирования приведены в Приложении А на рисунках А.1 - А.11.

Таблица 1 – Тестовый набор данных

|  |  |  |
| --- | --- | --- |
| № теста | Размер массива size | Время выполнения сортировки в секундах |
| 1 | 10000 | 0,189 |
| 2 | 20000 | 0,759 |
| 3 | 30000 | 1,744 |
| 4 | 40000 | 3,063 |
| 5 | 50000 | 4,819 |
| 6 | 60000 | 6,847 |
| 7 | 70000 | 9,221 |
| 8 | 80000 | 11,074 |
| 9 | 90000 | 13,161 |
| 10 | 100000 | 16,024 |
| 11 | 110000 | 18,963 |

**5.2 Анализ полученных результатов тестирования**

На основании анализа данных, полученных в результате тестирования алгоритма быстрой сортировки, можно сделать вывод, что время, затраченное на работу программы относительно количества элементов увеличивается линейно, то есть с увеличением количества элементов пропорционально увеличивается время работы программы. Это можно увидеть на рисунке 4.

Рисунок 4 – Результаты тестирования

# Отладка

В качестве среды разработки была выбрана программа MicrosoftVisualStudio , 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 глобальных и локальных переменных. Визуально это представлено на рисунке 5“Отладка программы”.

Тестирование проводилось в рабочем порядке, в процессе разработки, после завершения написания программы. После завершения написания программы, человеком, выполнявшим тестирование программы, были выявлены и исправлены ошибки.

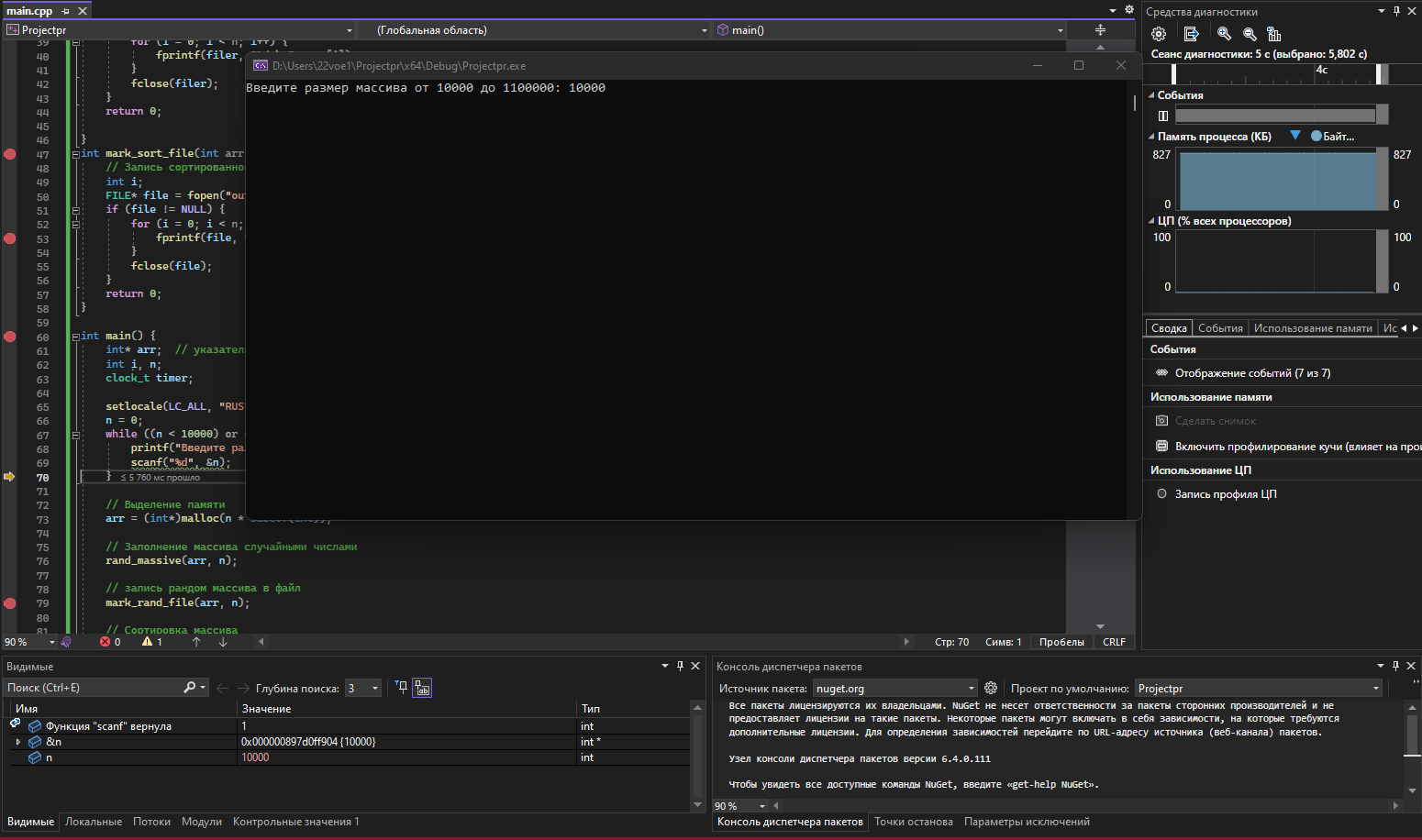


Рисунок 5 “Отладка программы”

# Совместная разработка

Во время работы над данной практикой, нашей бригадой осуществлялась совместная работа в GitHub.

Данная программа была написана – Назаровым Д.А.

После написания программы, она была выгружена на удаленный репозиторий Github, на отдельную ветку под названием practica

После этого, второй участник – Кочетков А.В загрузил данную программу себе на компьютер и добавил в нее работу с файлами. Написал код для записи в файл исходного и отсортированного массивов. После чего выгрузил обновленный код программы на удаленный репозиторий practic.

Ссылка на удаленный репозиторий: <https://github.com/andrsmerch/practic>

# Заключение

Нашей бригадой были получены навыки совместной работы с помощью сервиса GitHub, навыки использования программы GitBash. Нами так же был изученалгоритм сортировки методом выбора. Назаров Д.А. написал программу, выполняющую данную сортировку над массивом случайно сгенерированных чисел, и оформил часть отчета по данной практике. Кочетков А.В. выполнил работу с файлами и оформил часть отчета по данной практике.

Так же при выполнении практической работы были улучшены наши базовые навыки программирования на языках С/C++. Улучшены навыки отладки, тестирования программ и работы со сложными типами данных.

В дальнейшем программу можно улучшить путем подключения упрощающих реализацию данной сортировки библиотеки улучшения консольного 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 вставками [Электронный ресурс] – URL:

<https://ru.wikipedia.org/> (дата обращения: 02.07.2023 г)

**Приложение А**

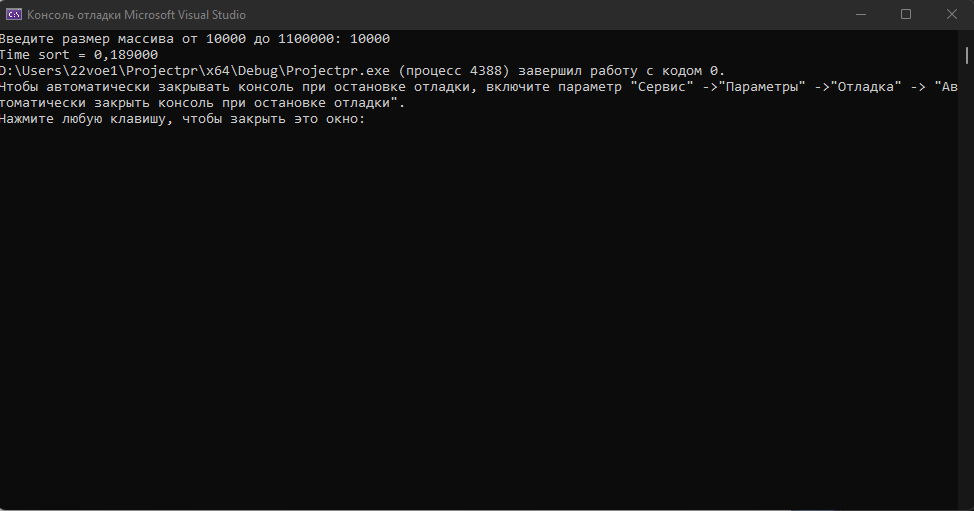


Рисунок А.1

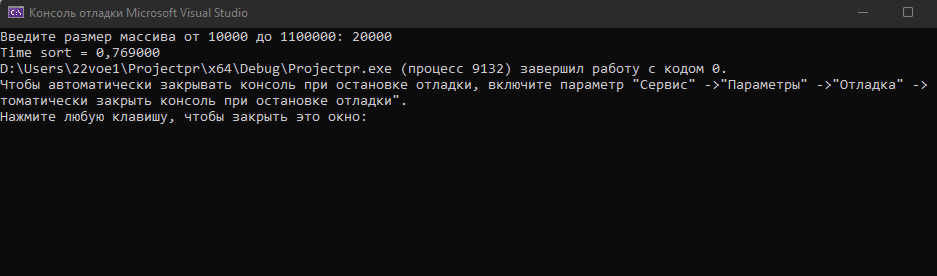


Рисунок А.2

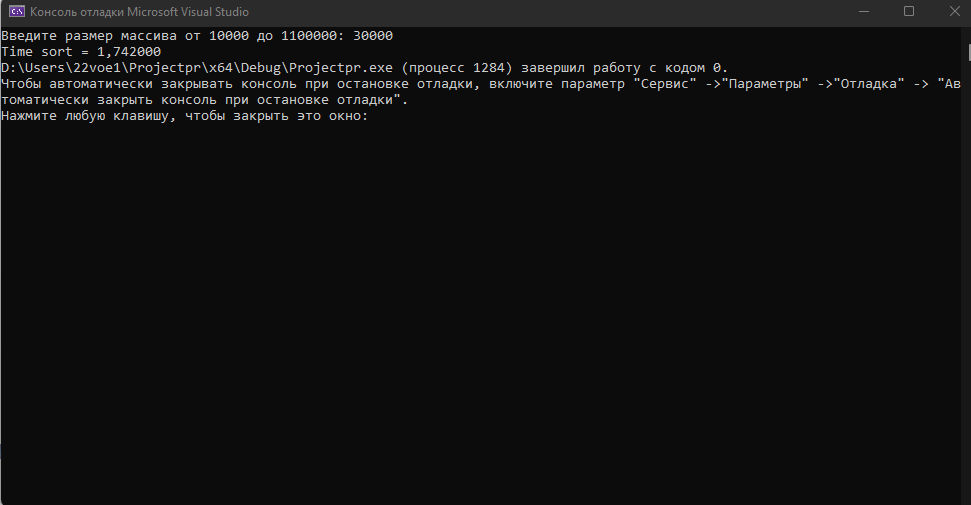


Рисунок А.3

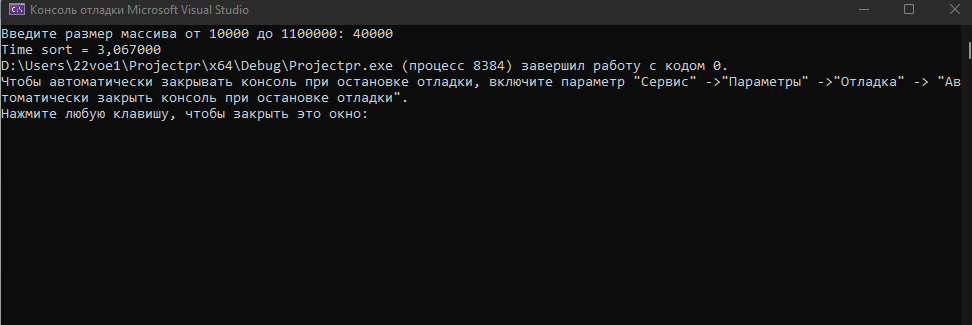


Рисунок А.4

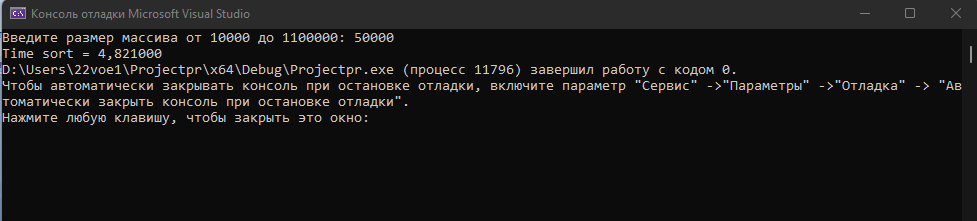


Рисунок А.5

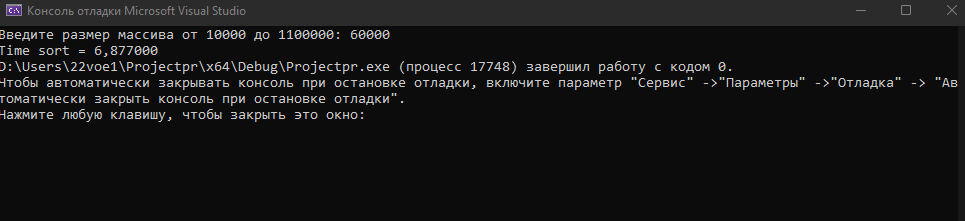


Рисунок А.6

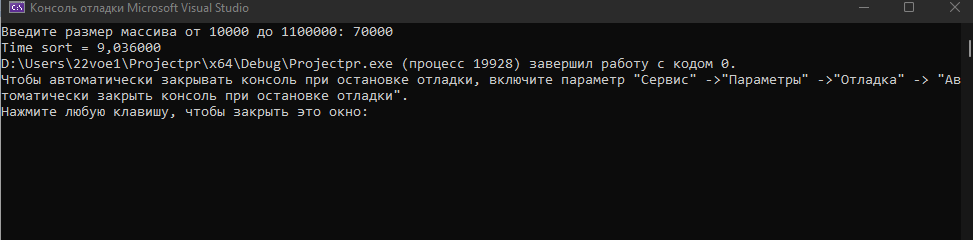


Рисунок А.7

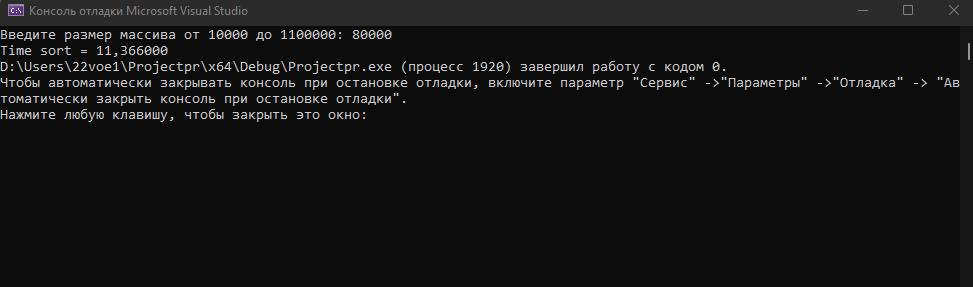


Рисунок А.8

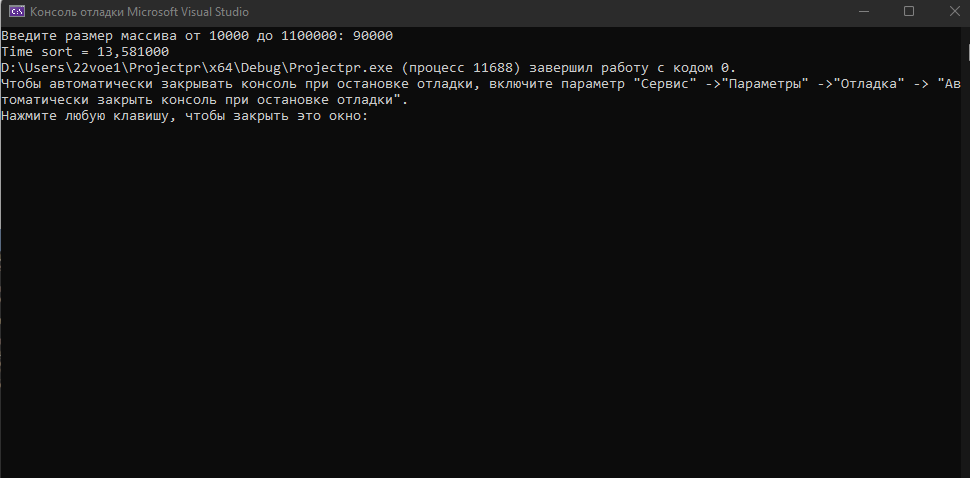


Рисунок А.9

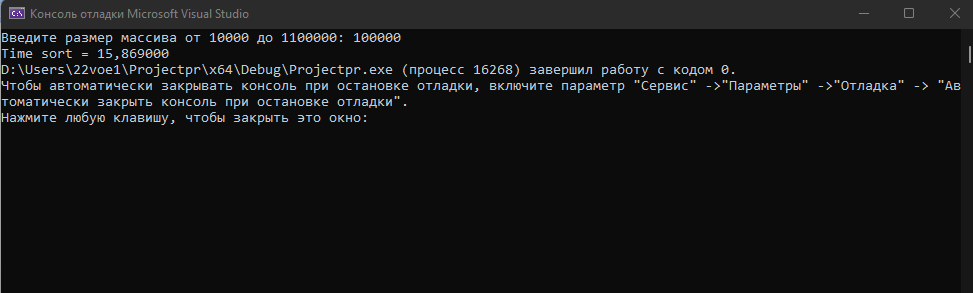


Рисунок А.10

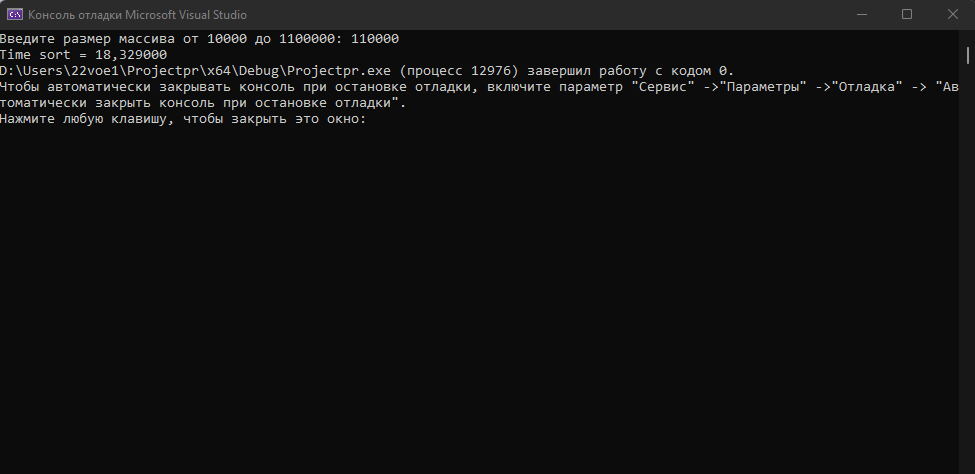


Рисунок А.11

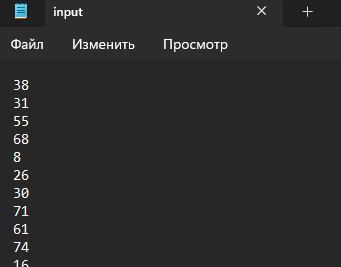
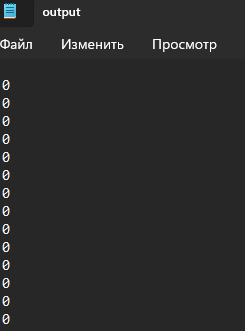


Рисунок А.12



РисунокА.13

# Приложение Б “Листинг”

#define\_CRT\_SECURE\_NO\_WARNINGS

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

#include<locale.h>

void selectionSort(intarr[], intn) {

// сортировкавыбором

int i, j, minIndex, temp;

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

minIndex = i;

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

if (arr[j] <arr[minIndex]) {

minIndex = j;

}

}

temp = arr[minIndex];

arr[minIndex] = arr[i];

arr[i] = temp;

}

}

int rand\_massive(intarr[], intn) {

int i;

// Заполнение массива случайными числами

srand(time(0));

for (i = 0; i <n; i++) {

arr[i] = rand() % 100;

}

return 0;

}

int mark\_rand\_file(intarr[], intn) {

// запись рандом массива в файл

int i;

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

if (filer != NULL) {

for (i = 0; i <n; i++) {

fprintf(filer, "%d \n", arr[i]);

}

fclose(filer);

}

return 0;

}

int mark\_sort\_file(intarr[], intn) {

// Запись сортированного массива в файл

int i;

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

if (file != NULL) {

for (i = 0; i <n; i++) {

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

}

fclose(file);

}

return 0;

}

int main() {

int\* arr; // указатель на int для массива

int i, n;

clock\_t timer;

setlocale(LC\_ALL, "RUS");

n = 0;

while ((n < 10000) or (n > 1100000)) {

printf("Введите размер массива от 10000 до 1100000: ");

scanf("%d", &n);

}

// Выделениепамяти

arr = (int\*)malloc(n \* sizeof(int));

// Заполнение массива случайными числами

rand\_massive(arr, n);

// запись рандом массива в файл

mark\_rand\_file(arr, n);

// Сортировкамассива

timer = clock();

selectionSort(arr, n);

timer = clock() - timer;

printf("Time sort = ");

printf("%f", (double)timer / CLOCKS\_PER\_SEC);

// Запись сортированного массива в файл

mark\_sort\_file(arr, n);

return 0;

}