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**

(2019/2020 учебный год)

                                        Анохин Андрей 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 обучения1семестр 2

Период прохождения практики с 06.07.2020 по 18.07.2020

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**

(2019/2020 учебный год)

                                        Анохин Андрей 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                  1                 семестр                 2

Период прохождения практики с 06.07.2020 по 18.07.2020

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 | 06.07.20 -  06.07.20 |  |
| 2 | Подбор и изучение материала по теме работы | 15 | 06.07. 20 –  07.07. 20 |  |
| 3 | Разработка алгоритма | 43 | 07.07.20 –  11.07. 20 |  |
| 4 | Описание алгоритма и программы | 18 | 13.07.19 –  15.07. 20 |  |
| 5 | Тестирование | 5 | 15.07. 20 –  15.07. 20 |  |
| 6 | Получение и анализ результатов | 10 | 16.07. 20 –  17.07. 20 |  |
| 7 | Оформление отчёта | 15 | 17.07. 20 –  18.07. 20 |  |
|  | **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**

(2019/2020 учебный год)

                                      Анохин Андрей 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 обучения1семестр 2

Период прохождения практики с 06.07.2020 по 18.07.2020

Кафедра «Вычислительная техника»

Анохин А.А. выполнял практическое задание «Сортировка пузырьком». На первоначальном этапе был изучен способ написания алгоритма программы, был выбран метод решения и язык программирования С, на котором был написан алгоритм программы сортировки массива методом пузырька. Протестировал и отладил программу. Оформил отчёт.

Бакалавр Анохин А.А. \_\_\_\_\_\_\_\_\_\_\_\_ "\_\_\_" \_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Руководитель Карамышева Н.С. \_\_\_\_\_\_\_\_\_\_\_\_ "\_\_\_" \_\_\_\_\_\_\_\_\_\_\_ 2020 г.

практики

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

**ОТЗЫВ**

**О ПРОХОЖДЕНИИ УЧЕБНОЙ (ОЗНАКОМИТЕЛЬНОЙ) ПРАКТИКИ**

(2019/2020 учебный год)

                                        Анохин Андрей 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 обучения1семестр 2

Период прохождения практики с 06.07.2020 по 18.07.2020

Кафедра «Вычислительная техника»

В процессе выполнения практики Анохин А.А. решал следующие задачи: создание алгоритма пузырьковой сортировки.

За период выполнения практики были освоены основные понятия и технологии. Во время выполнения работы Анохин А.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 Анохин А.А. заслуживает оценки «\_\_\_\_\_\_».

Руководитель практики к.т.н., доцент, Карамышева Н.С. « » 2020 г.

Оглавление

[1 Введение](#_Toc45550651) 6

[2 Постановка задачи](#_Toc45550652) 6

[3 Разработка и описание алгоритма](#_Toc45550653) 7

[4 Описание алгоритма программы](#_Toc45550655) 9

[5 Тестирование интерфейса](#_Toc45550656) 15

[6 Заключение](#_Toc45550657) 17

[7 Литература](#_Toc45550658) 18

[Приложение 1 – Листинг всей программы](#_Toc45550659) 18

## 1 Введение

Как известно, процесс проникновения информационных технологий практически во все сферы человеческой деятельности продолжает развиваться и углубляться. Помимо уже привычных и широко распространенных персональных компьютеров, общее число которых достигло многих сотен миллионов, становится все больше и встроенных средств вычислительной техники.

Пользователей всей этой разнообразной вычислительной техники становится все больше, причем наблюдается развитие двух вроде бы противоположных тенденций. С одной стороны, информационные технологии все усложняются, и для их применения, и тем более дальнейшего развития, требуется иметь очень глубокие познания. С другой стороны, упрощаются интерфейсы взаимодействия пользователей с компьютерами.

Компьютеры и информационные системы становятся все более дружественными и понятными даже для человека, не являющегося специалистом в области информатики и вычислительной техники. Это стало возможным прежде всего потому, что пользователи и их программы взаимодействуют с вычислительной техникой посредством специального (системного) программного обеспечения - через операционную систему. Операционная система предоставляет интерфейсы и для выполняющихся приложений, и для пользователей.

## 2 Постановка задачи

В ходе учебной практики перед нашей бригадой была поставлена задача, разработать программу, сортирующую массив чисел по возрастанию алгоритмом “Пузырьковая сортировка”. Для того, чтобы реализовать эту задачу, мы написали несколько мелких подзадач:

* Изучение синтаксиса языка C;
* Проектирование логики работы программы;
* Проектирование графического интерфейса;
* Разработка алгоритма;
* Разработка функции для работы с файлами;
* Разработка программы сортировки массивов (её графического интерфейса);
* Тестирование и отладка программы;

Данная программа должна обладать следующим функционалом:

1. Ввод массива пользователем и его сортировка;
2. Автоматическое создание массива с выбором кол-ва элементов и указанием предела элементов и их сортировка;
3. Загрузка массива из \*.txt файла и последующая его сортировка;
4. Вывод на экран изначального и отсортированного массива;

Моя задача состоит в том, чтобы спроектировать и разработать графический интерфейс нашей программы.

## 3 Описание алгоритма

Алгоритм сортировки пузырьком довольно прост.

Метод его работы состоит в следующем: на первом шаге сортировки мы проходим циклом от первого элемента массива к последнему, по пути мы просматриваем пары соседних элементов, если элементы некоторой пары находятся в неправильном порядке, то мы их меняем. То есть, если элемент [n-1] больше элемента [n], то на место n-го элемента мы ставим элемент [n-1].

После нулевого прохода на нулевом месте в массиве оказывается самое маленькое число. На следующем проходе в массиве на первом месте по аналогии оказывается число, которое больше числа на нулевом элементе, но меньше всех остальных чисел в массиве. Для работы алгоритма используется вложенный цикл, количество итераций внешнего и внутреннего цикла равно [длина массива – 1].

Далее приведена схема алгоритма сортировки пузырьком:

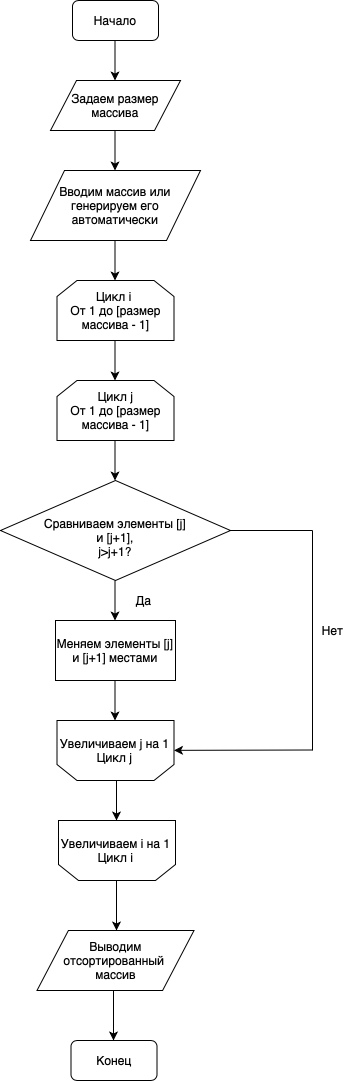


Рисунок 1 – алгоритм сортировки пузырьком

**Преимущества и недостатки алгоритма:**

Преимущества:

Простейший для понимания и реализации

На его основе построены более эффективные алгоритмы сортировки – шейкерная сортировка, сортировка Шелла и т.д.

Недостатки:

Имеет сложность O(n^2), что делает его довольно медленным

Эффективен лишь для небольших массивов

Считается учебным и практически не применяется вне учебной литературы

## 4 Описание алгоритма программы

Для создания алгоритма программы сортировки пузырьком было использовано несколько библиотек, которые представлены дальше.

**stdlib**.**h** — заголовочный файл стандартной библиотеки языка Си, который содержит в себе функции, занимающиеся выделением памяти, контролем процесса выполнения программы, преобразованием типов и другие.

**windows.h –** специфический файл заголовка для C и C++ языки программирования, который содержит объявления для всех функций в Windows API, все общие макросы используются программистами для Windows, а также все типы данных, используемые различными функциями и подсистемы. Он определяет очень большое количество специфических функций Windows, которые могут быть использованы в С.

**locale.h** – заголовочный файл стандартной библиотеки языка программирования С, который используется для задач, связанных с локализацией.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Наименование функции | Действие | Блок-Схема |
| 1 | bubble | Сортирует массив | Рисунок 2 |
| 2 | autowrite | Генерирует случайный массив исходя из его размеров и чисел для определения его диапазона, затем выводит отсортированный массив | Рисунок 3 |
| 3 | selfwrite | Дает пользователю вручную ввести размер массива и его элементы, затем выводит отсортированный массив | Рисунок 4 |
| 4 | load | Загружает массив из файла, затем сортирует и выводит его | Рисунок 5 |
| 5 | menu | Графический интерфейс | Рисунок 6 |
| 6 | main | Обрабатывает нажатие кнопок для взаимодействия с программой в графическом интерфейсе | Рисунок 7 |

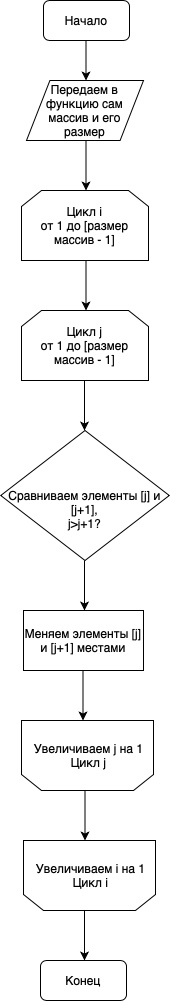


Рисунок 2 – функция bubble

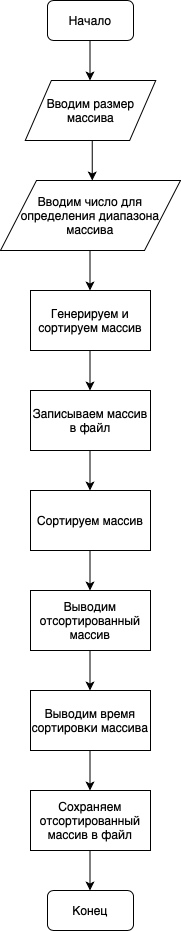


Рисунок 3 - функция autowrite

## 

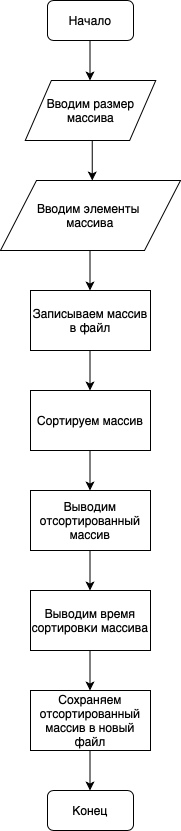


Рисунок 4 – функция selfwrite



Рисунок 5 –функция load



Рисунок 6 –функция Menu

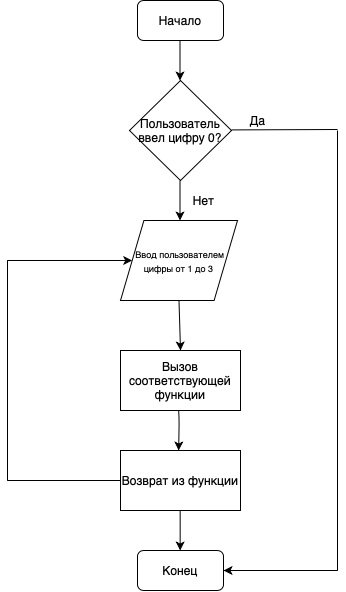


Рисунок 7 – функция main

## 5 Тестирование алгоритма

Алгоритм был протестирован в работе с большим количеством элементов в массиве (Рисунок 7, Рисунок 8,Рисунок 9), но ошибок выявлено не было. Рисунок 7- интерфейс при 1000

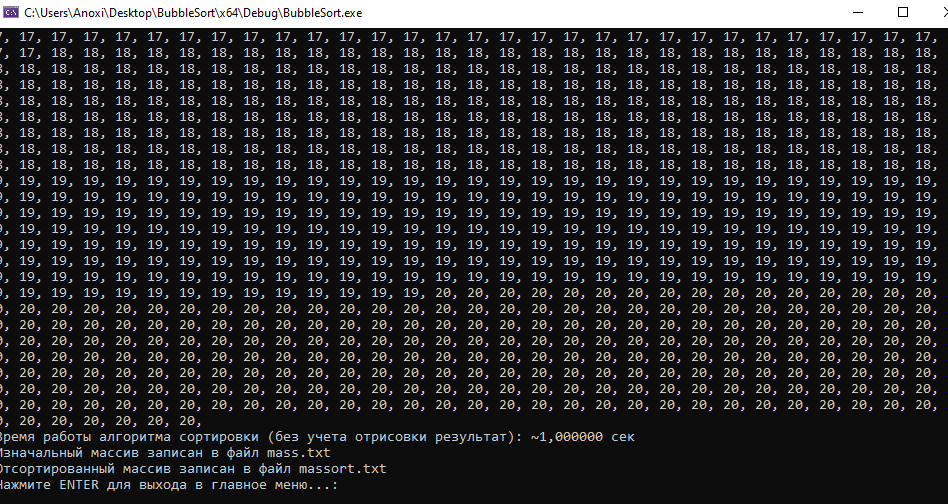
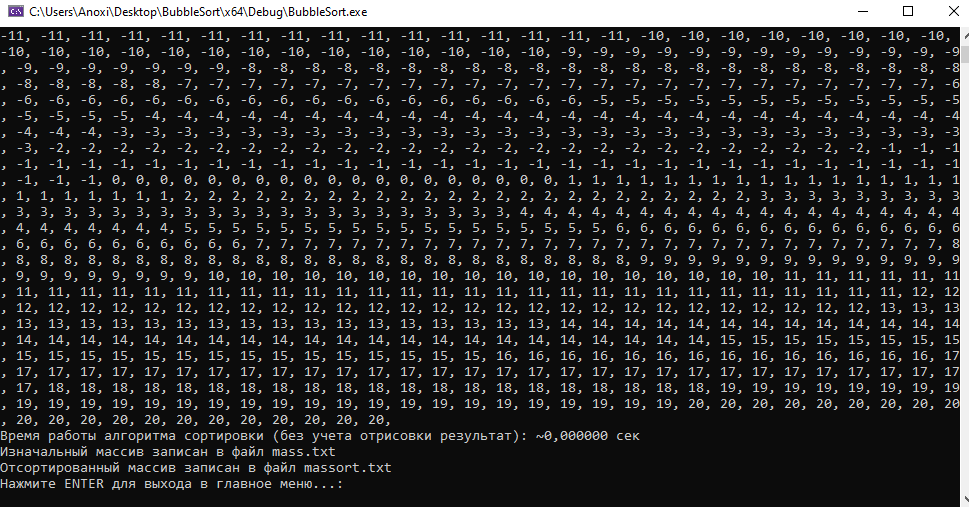


Рисунок 8 – интерфейс при 10000

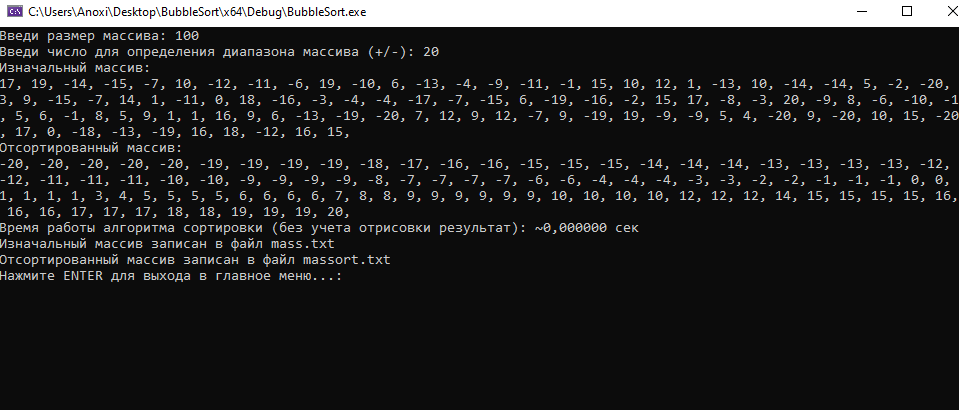


Рисунок 9 – интерфейс при 100

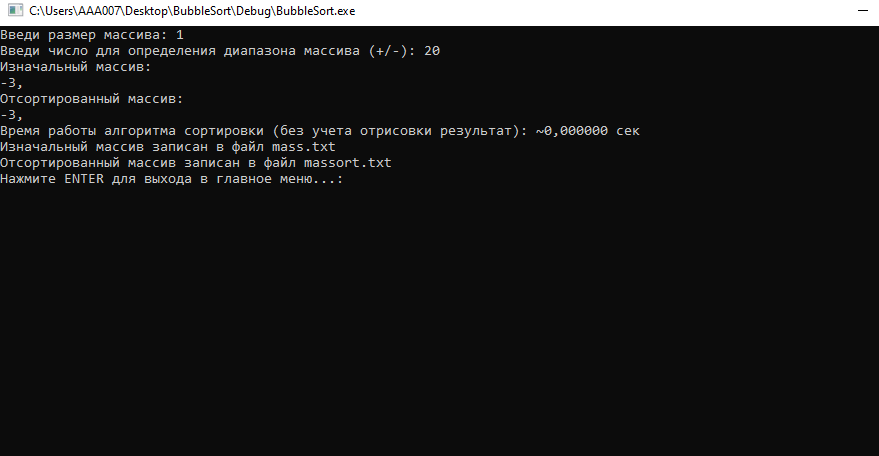
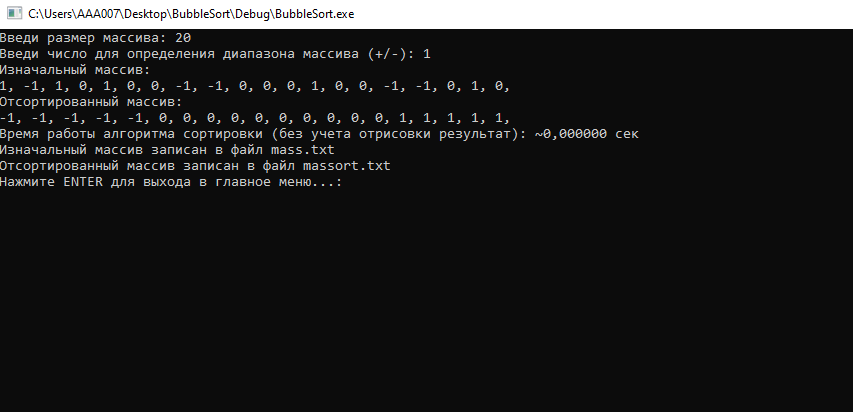


Рисунок 10-массив от 1 до 20

Рисунок 11-массив от 20 до 1



## 6 Заключение

В результате выполнения практической работы были получены навыки командной работы по работе над проектом, изучены различны алгоритмы сортировки.

Была разработана программа с графическом интерфейсом, предназначенная для сортировки массива чисел. Для реализации этой функции на языке C был написан известный алгоритм сортировки “пузырьком”. Программа была отлажена и неоднократна протестирована для полной уверенности в ее работоспособности.

В дальнейшем программа может быть доработана: может быть добавлено логирование, изменен графический интерфейс и расширен функционал сортировки.

## 7 Литература

1. Айра Пол Объектно-ориентированное программирование на С++ / Айра Пол. - М.: Не указано, Бином, Невский Диалект, 2001. - 464 c.
2. Александр Чиртик Программирование на C++. Трюки и эффекты / Александр Чиртик. - М.: "Издательство "Питер", 2010. - 352 c.
3. Бьерн Страуструп Дизайн и эволюция языка С++ / Бьерн Страуструп. - М.: ДМК Пресс, 2014. - 446 c.
4. Герберт Шилдт Искусство программирования на C++ / Герберт Шилдт. - М.: БХВ-Петербург, 2005. - 496 c.
5. Грег Перри Программирование на C для начинающих / Грег Перри , Дин Миллер. - М.: Эксмо, 2015. - 368 c

## 

## Приложение 1 – Листинг всей программы

#define \_CRT\_SECURE\_NO\_WARNINGS // для корректной работы scanf()

#include <conio.h>

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <locale.h>

#include <windows.h>

int n, i, mas;

void bubble(int\* num, int size)

{

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

{

for (int j = (size - 1); j > i; j--)

{

if (num[j - 1] > num[j])

{

int temp = num[j - 1];

num[j - 1] = num[j];

num[j] = temp;

}

}

}

}

int autowrite(void)

{

int tb, te, i, n, e, \* mas;

printf("Введи размер массива: ");

scanf("%d", &n);

printf("Введи число для определения диапазона массива (+/-): ");

scanf("%d", &e);

e++;

srand(time(0));

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

printf("Изначальный массив: \n");

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

mas[i] = rand() % (2\*e-1)+(-e+1);

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

}

FILE\* F;

F = fopen("mass.txt", "wt");

fprintf(F, "%d ", n);

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

fprintf(F, "%d ", mas[i]);

fclose(F);

time\_t start, end;

time(&start);

bubble(mas, n);

time(&end);

printf("\nОтсортированный массив: \n");

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

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

FILE\* F1;

double seconds = difftime(end, start);

printf("\nВремя работы алгоритма сортировки (без учета отрисовки результат): ~%f сек", seconds);

printf("\nИзначальный массив записан в файл mass.txt");

printf("\nОтсортированный массив записан в файл massort.txt");

F1 = fopen("masssort.txt", "wt");

fprintf(F1, "%d ", n);

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

fprintf(F1, "%d ", mas[i]);

fclose(F1);

printf("\nНажмите ENTER для выхода в главное меню...: \n");

getchar(); getchar();

return 0;

}

int selfwrite(void)

{

int i, n, \* mas;

printf("Введи размер массива: ");

scanf("%d", &n);

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

printf("Введи элементы массива: \n");

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

{

printf("a[%d] = ", i);

scanf("%d", &mas[i]);

}

FILE\* F;

F = fopen("mass.txt", "wt");

fprintf(F, "%d ", n);

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

fprintf(F, "%d ", mas[i]);

fclose(F);

bubble(mas, n);

printf("\nОтсортированный массив: \n");

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

{

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

}

FILE\* F1;

printf("\nИзначальный массив записан в файл mass.txt");

printf("\nОтсортированный массив записан в файл massort.txt");

F1 = fopen("masssort.txt", "wt");

fprintf(F1, "%d ", n);

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

fprintf(F1, "%d ", mas[i]);

fclose(F1);

printf("\nНажмите ENTER для выхода в главное меню... \n");

getchar(); getchar();

return 0;

}

void load(void)

{

int i, n, e, \*mas;

// определяем указатель на файл

FILE\* fin;

// открываем файл на чтение

fin = fopen("mass.txt", "r");

// построчное считывание из файла

fscanf(fin, "%d", &n);

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

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

{

fscanf(fin, "%d", &mas[i]);

}

printf("Массив из файла загружен и отсортирован!\nИзначальный массив:\n");

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

{

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

}

bubble(mas, n);

printf("\nОтсортированный массив: \n");

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

{

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

}

printf("\nНажмите ENTER для выхода в главное меню...\n");

\_getch();

fclose(fin);

}

int Menu(void)

{

system("CLS");

int c = 0;

setlocale(LC\_ALL, "Rus");

while (c < '0' || c > '3')

{

printf(

"::=================================::\n"

"|| 1 : Ввести массив ||\n"

"|| 2 : Автоматический ввод массива ||\n"

"|| 3 : Загрузить из файла ||\n"

"|| 0 : Закрыть программу ||\n"

"::=================================::\n"

"> ");

c = \_getch();

system("cls");

}

return c;

}

void main(void)

{

int Select;

while ((Select = Menu()) != '0' && Select != 27)

switch (Select)

{

case '1':

selfwrite();

break;

case '2':

autowrite();

break;

case '3':

load();

break;

}

}