

КАФЕДРА №

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

должность, уч. степень, звание

подпись, дата

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

Обработка данных в файлах

по курсу: ОСНОВЫ ПРОГРАММИРОВАНИЯ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

подпись, дата

инициалы, фамилия

Санкт-Петербург 2022

1.Цель работы: Целью работы является изучение файловых операций ввода-вывода

2.Задачи работы:

Написать программу, которая считывает текст из файла и записывает его в другой файл, добавляя после каждого предложения, сколько раз встретилось в нём введённое с клавиатуры слово.

3.Описание функций:

1)Имя: **main** для **fstream** и **cstdio**

Назначение: ввод пользователем имя файла и слова для поиска, вывод на экран количества повторений слова в предложении после каждого предложения

Входные данные: нет

Выходные данные: нет

Побочный эффект: отсутствует.

Тестовые данные:

Ввод: 3D

Вывод: What is 3D printing?(1)

Прототип: int main()

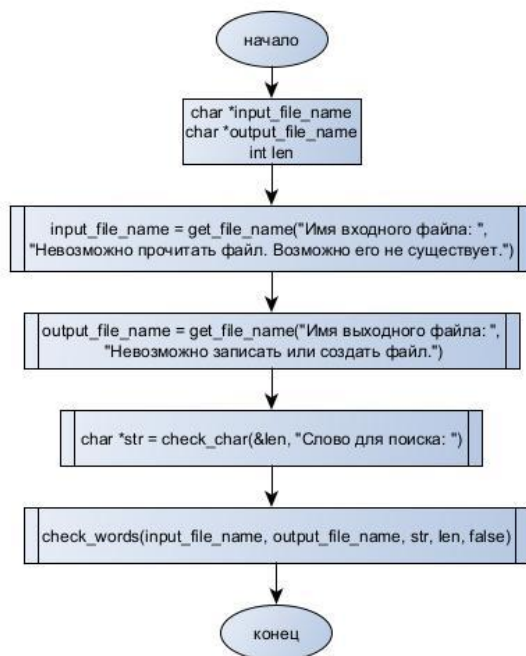
Псевдокод:

ввод имени файла

Подсчет количества найденных слов

Запись в файл

Блок-схема:



2)Имя: **check_words()** для **fstream**

Назначение: функция для подсчета слов в предложениях

Входные данные: `*input_file_name`, `*output_file_name`, `*str`, `len`, `data`

Выходные данные: нет

Побочный эффект: отсутствует.

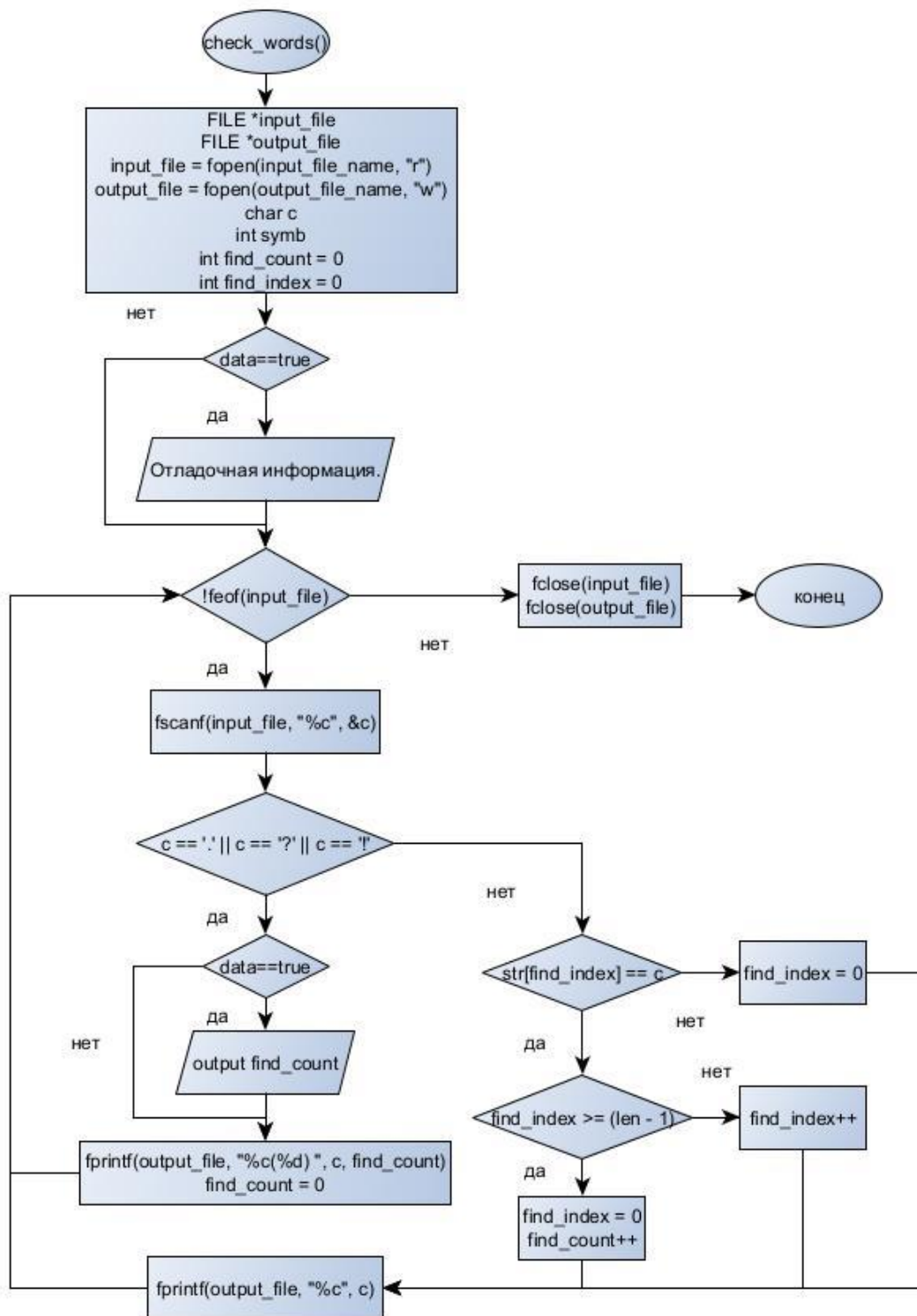
Прототип: void check_words()

Псевдокод:

если встречаем конец предложения, то записываем в него количество слов и обнуляем переменные для подсчёта

в остальных случаях ищем слова
если слово полностью совпадает, то обновляем счётчик
сохранение и закрытие файла

Блок-схема:



3)Имя: check_words() для cstdio

Назначение: функция для подсчета слов в предложениях

Входные данные: *input_file_name, *output_file_name, *str, len, data

Выходные данные: нет

Побочный эффект: отсутствует.

Прототип: `void check_words()`

Псевдокод:

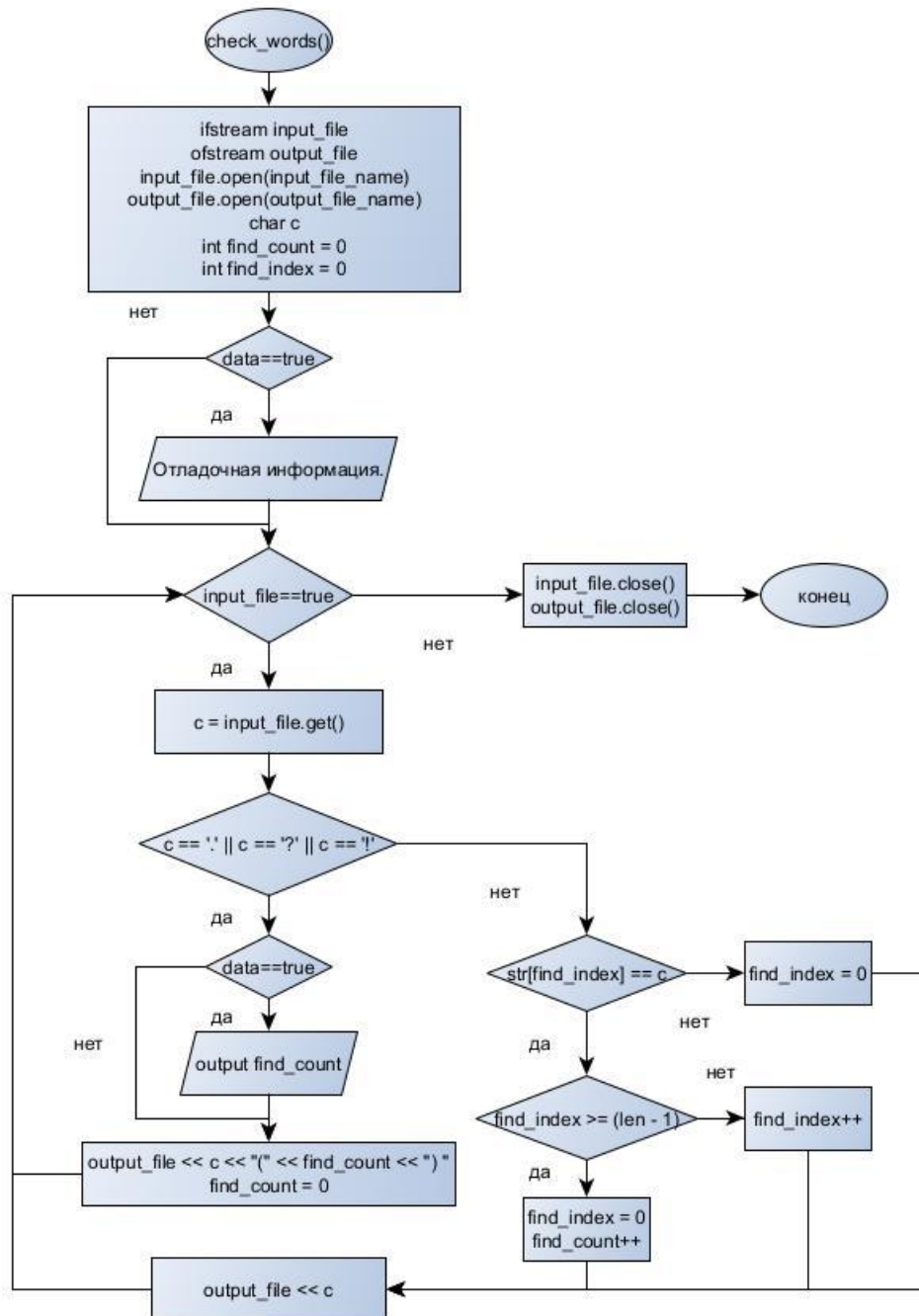
если встречаем конец предложения, то записываем в него количество слов и обнуляем переменные для подсчёта

в остальных случаях ищем слова

если слово полностью совпадает, то обновляем счётчик

сохранение и закрытие файла

Блок-схема:



4)Имя: `get_file()` для `cstdio`

Назначение: функция для подсчета слов в предложениях

Входные данные: `*input_file_name`, `*output_file_name`, `*str`, `len`, `data`

Выходные данные: нет

Побочный эффект: отсутствует.

Прототип: `char* get_file()`

Псевдокод:

Ввод названия файла

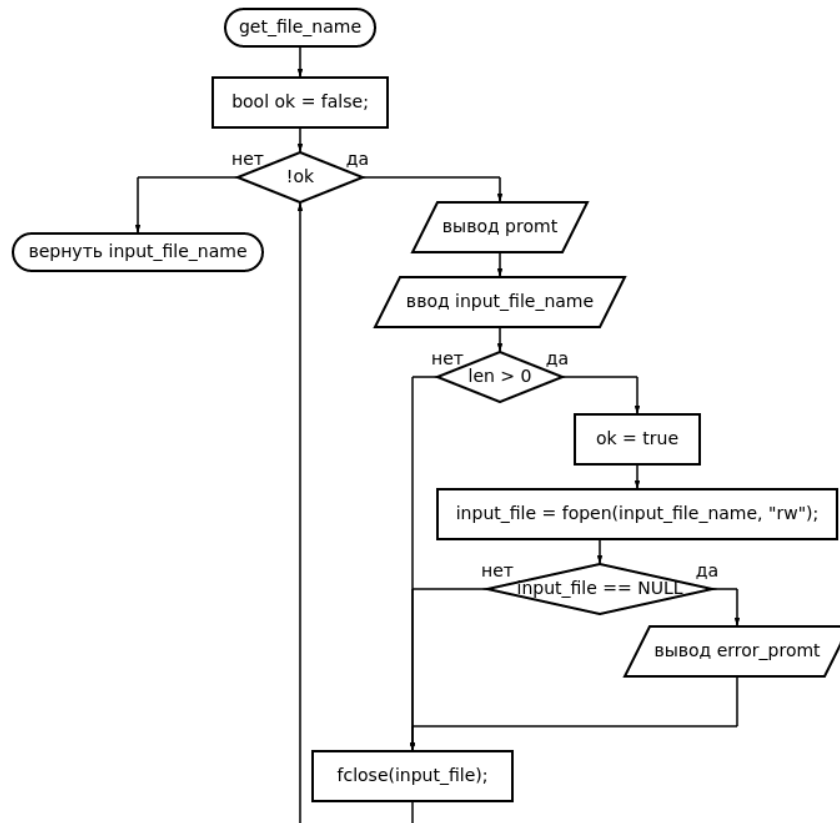
Открываем файл

Если файла не существует, либо его нельзя записать, то повторяем всё заново

Закрываем файл

Возвращаем название файла

Блок-схема:



4)Имя: get_file() для fstream

Назначение: функция для подсчета слов в предложениях

Входные данные: *input_file_name, *output_file_name, *str, len, data

Выходные данные: нет

Побочный эффект: отсутствует.

Прототип: char* get_file()

Псевдокод:

Ввод названия файла

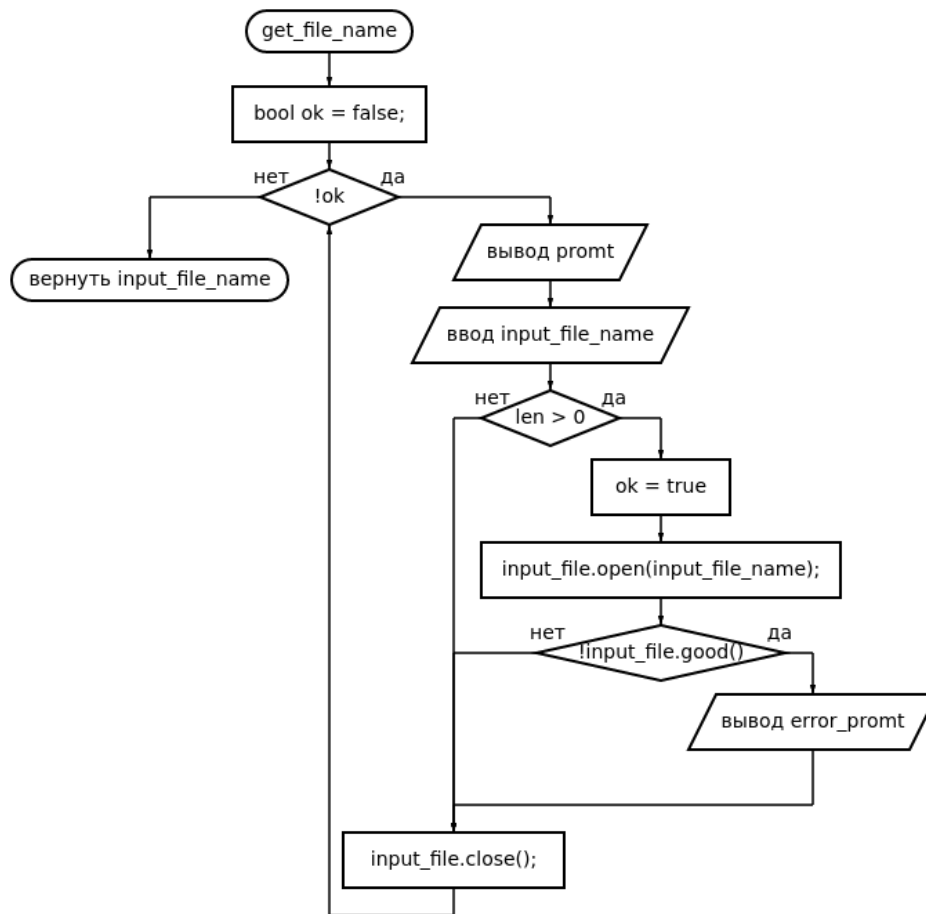
Открываем файл

Если файла не существует, либо его нельзя записать, то повторяем всё заново

Закрываем файл

Возвращаем название файла

Блок-схема:



5.ЛИСТИНГ КОДА:

1)Для fstream

```

#include <iostream>
using namespace std;

#include "libs/lib.h"
#include <cmath>
#include <time.h>
#include <fstream>
#include <cctype>

#include "libs/simple_char.h"

#include "fstream_files.h"

char *get_file_name(const char *prompt = "", const char *error_prompt = "", bool
new_file = false) {

    ifstream input_file;

    int len;
    char *input_file_name;

    bool ok = false;
    while (!ok) {
        cout << prompt;
        input_file_name = get_string(&len);
        if (len > 0) {
            ok = true;

```

```

        input_file.open(input_file_name);

        if (!input_file.good()) {
            if (new_file) {
                ofstream ost(input_file_name);
            } else {
                cout << error_promt << endl;
                ok = false;
            }
        }

        input_file.close();

    }
}

return input_file_name;
}

// функция для подсчёта слов
void check_words(char *input_file_name, char *output_file_name, char *str, int
len, bool data = false) {

    ifstream input_file;
    ofstream output_file;

    // открываем файлы
    input_file.open(input_file_name);
    output_file.open(output_file_name);

    char c;

    if (data) {
        draw_line(20);
        cout << "Отладочная информация." << endl;
        draw_line(20);
    }

    int find_count = 0;
    int find_index = 0;

    // цикл по всему файлу
    while (input_file) {

        // берём из файла 1 символ
        c = input_file.get();

        // если встречаем конец предложения, то записываем в него количество слов и
        обнуляем переменные для подсчёта
        if (c == '.' || c == '?' || c == '!') {
            if (data) cout << c << "(" << find_count << ")" ";
            output_file << c << "(" << find_count << ")" ";
            find_count = 0;
        } else {
            // в остальных случаях ищем слова
            if (str[find_index] == c) {
                // если слово полностью совпадает, то обновляем счётчик
                if (find_index >= (len - 1)) {
                    find_index = 0;
                    find_count++;
                } else {

```

```

        find_index++;
    }
} else {
    find_index = 0;
}
//
output_file << c;
}
}

input_file.close();
output_file.close();

}

int main() {
    system("chcp 65001");

    //clear_scr();

    char *input_file_name;
    char *output_file_name;

    // вводим имена файлов (с проверкой)
    input_file_name = get_file_name("Имя входного файла: ", "Невозможно прочитать
    файл. Возможно его не существует.");
    output_file_name = get_file_name("Имя выходного файла: ", "Невозможно записать
    или создать файл.", true);

    draw_line(20);
    int len;
    // вводим слово для поиска
    char *str = check_char(&len, "Слово для поиска: ");

    // подсчитываем найденные слова
    check_words(input_file_name, output_file_name, str, len, false);

    draw_line(20);

    cout << "В файл " << output_file_name << " был успешно записан результат." <<
    endl;

    return 0;
}

```

2)Для cstdio

```

char *get_file_name(const char *prompt = "", const char *error_prompt = "") {

    FILE *input_file;

    int len;
    char *input_file_name;

    bool ok = false;
    while (!ok) {
        cout << prompt;
        input_file_name = get_string(&len);
        if (len > 0) {
            ok = true;

            input_file = fopen(input_file_name, "rw"); // если выдаёт ошибку поставь
            "r" вместо "rw"

            if (input_file == NULL) {

```



```

        cout << error_promt << endl;
        ok = false;
    }

    fclose(input_file);

}

}

return input_file_name;
}

void check_words(char *input_file_name, char *output_file_name, char *str, int
len, bool data = false) {

    FILE *input_file;
    FILE *output_file;

    input_file = fopen(input_file_name, "r");
    output_file = fopen(output_file_name, "w");

    char c;
    int symb;

    if (data) {
        draw_line(20);
        cout << "Отладочная информация." << endl;
        draw_line(20);
    }

    int find_count = 0;
    int find_index = 0;

    while (!feof(input_file)) {
        fscanf(input_file, "%c", &c);

        if (c == '.' || c == '?' || c == '!') {
            if (data) cout << c << "(" << find_count << ") ";
            fprintf(output_file, "%c(%d) ", c, find_count);
            find_count = 0;
        } else {
            if (str[find_index] == c) {
                if (find_index >= (len - 1)) {
                    find_index = 0;
                    find_count++;
                } else {
                    find_index++;
                }
            } else {
                find_index = 0;
            }
            fprintf(output_file, "%c", c);
        }
    }

    fclose(input_file);
    fclose(output_file);

}

int main() {
    // смена кодировки
    system("chcp 65001");

    // очистка терминала

```

```

//clear_scr();

char *input_file_name;
char *output_file_name;

input_file_name = get_file_name("Имя входного файла: ", "Невозможно прочитать
файл. Возможно его не существует.");
output_file_name = get_file_name("Имя выходного файла: ", "Невозможно записать
или создать файл.");

draw_line(20);
int len;
char *str = check_char(&len, "Слово для поиска: ");

check_words(input_file_name, output_file_name, str, len, false);

draw_line(20);

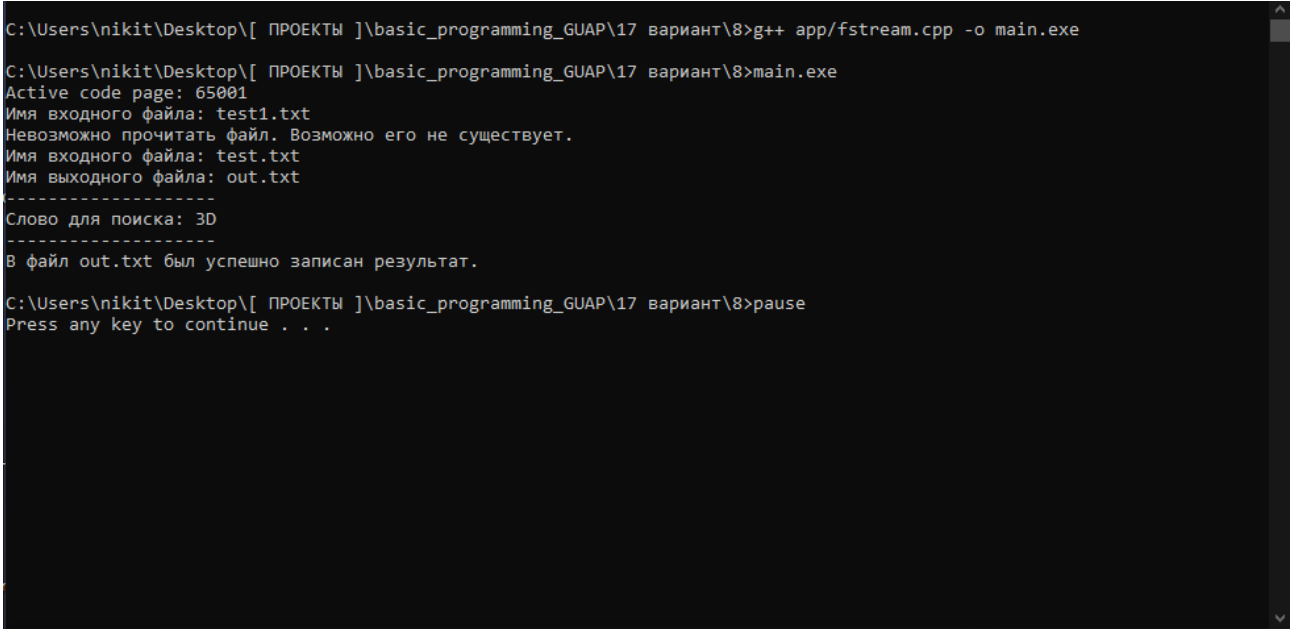
cout << "В файл " << output_file_name << " был успешно записан результат." <<
endl;

return 0;
}

```

6.Пример выполнения программы:

main



```

C:\Users\nikit\Desktop\[ ПРОЕКТЫ ]\basic_programming_GUAP\17 вариант\8>g++ app/fstream.cpp -o main.exe
C:\Users\nikit\Desktop\[ ПРОЕКТЫ ]\basic_programming_GUAP\17 вариант\8>main.exe
Active code page: 65001
Имя входного файла: test1.txt
Невозможно прочитать файл. Возможно его не существует.
Имя входного файла: test.txt
Имя выходного файла: out.txt
-----
Слово для поиска: 3D
-----
В файл out.txt был успешно записан результат.
C:\Users\nikit\Desktop\[ ПРОЕКТЫ ]\basic_programming_GUAP\17 вариант\8>pause
Press any key to continue . . .

```

out.txt – Блокнот

Файл Правка Формат Вид Справка

What is 3D Printing?(1)
3D printing or additive manufacturing is a process of making three dimensional solid objects from a digital file.(1)
The creation of a 3D printed object is achieved using additive processes.(1) In an additive process an object is created by laying down successive layers of material until the object is created.(0) Each of these layers can be seen as a thinly sliced cross-section of the object.(0)
3D printing is the opposite of subtractive manufacturing which is cutting out / hollowing out a piece of metal or plastic with for instance a milling machine.(1)
3D printing enables you to produce complex shapes using less material than traditional manufacturing methods

7.Анализ результатов и выводы:

В ходе этой лабораторной работы мы изучили структуру данных одномерных массивов и перезаписывать их.

Из плюсов можно отметить, что программа работает корректно.
Существенных минусов не обнаружено.