Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

(ПНИПУ)

ОТЧЁТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №8

по теме:

БЛОКОВЫЙ ВВОД-ВЫВОД

Выполнила: студентка группы РИС-22-1б

Черкасова А.А\_\_\_\_\_\_\_\_\_\_\_\_\_

Проверил: доцент кафедры ИТАС

Полякова О.А\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_/\_\_\_\_\_\_\_/\_\_\_\_\_г.

Пермь 2023

**Содержание**

[**Введение** 3](#_Toc127379672)

[**Анализ задачи** 3](#_Toc127379673)

[**Блок – схема** 4](#_Toc127379674)

[**Приложение А** 6](#_Toc127379675)

[**Приложение Б** 7](#_Toc127379676)

# **Введение**

**Постановка задачи:** Сформировать файл из элементов структуры: "Книга":

* название;
* автор;
* год издания;
* количество страниц,

распечатать его содержимое. Удалить 3 элемента из начала файла, добавить элемент перед элементом с указанным названием.

Формирование, печать, добавление и удаление элементов оформить в виде функций. Предусмотреть сообщения об ошибках при открытии файла и выполнении операций ввода/вывода.

**Цель:** Работа с двоичными файлами, организация ввода-вывода структурированной информации и ее хранение на внешних носителях.

Для достижения поставленной цели, необходимо решить следующие **задачи**:

* Провести анализ задачи
* Реализовать задачу на языке С++
* Составить блок-схему

# **Анализ задачи**

1. Задаётся структура *Book* в соответствие с постановкой задачи.

2. В функции *main* происходят вызовы функции*.*

3. В функции *Filling* происходит заполнение массива структур через цикл *for* Пользователь может задать количество структур – *N*. Функция возвращает массив структур.

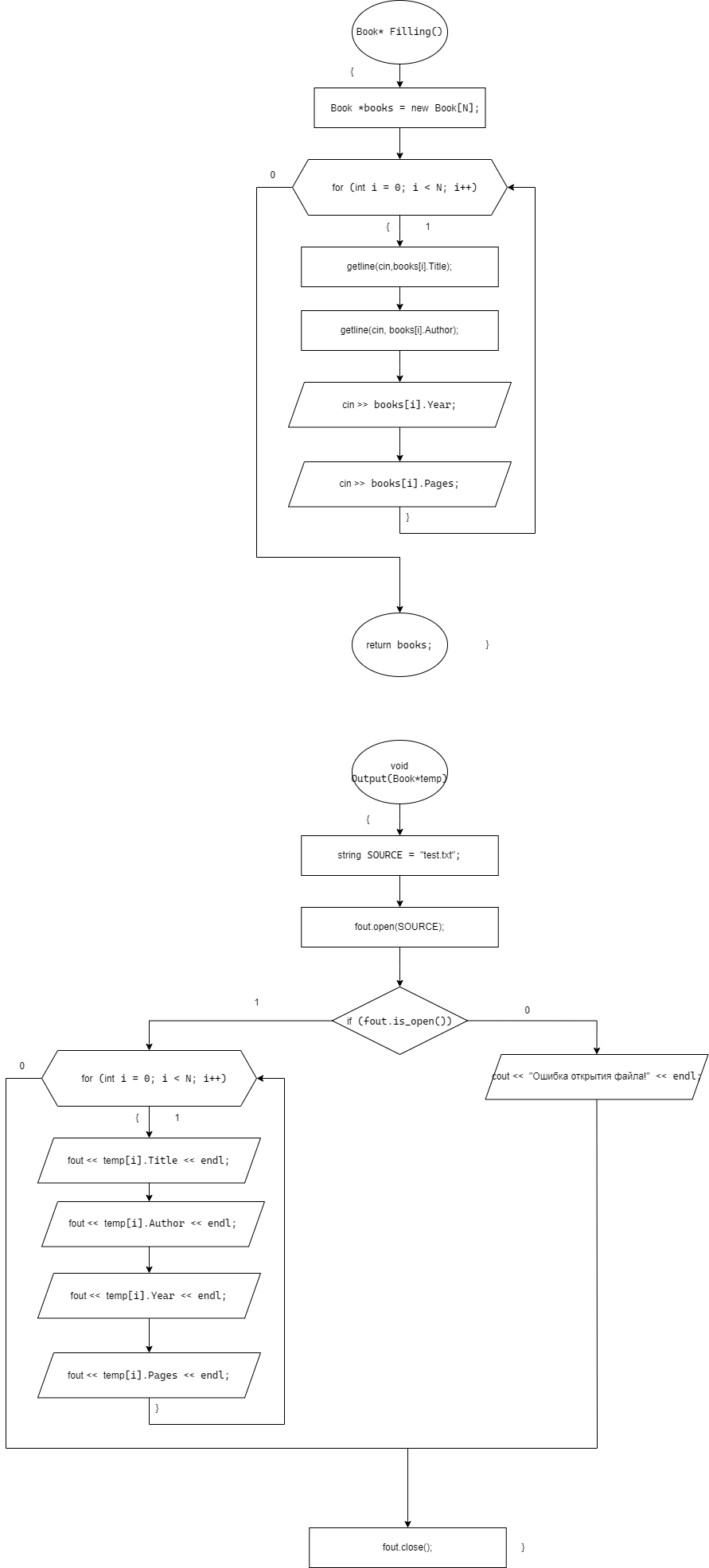
4. *Output –* функция, которая выводит данные в текстовый файл. Работа с файлом осуществляется при помощи библиотеки *fstream.* Создаётся особый указатель на файл *ofstream* в глобальной области видимости. Функция *open* открывает поток для записи в файл. Далее идёт проверка на открытие файла, если файл открылся, то туда записываются последовательно структуры из массива. После того, как запись в файл произошла, требуется закрыть поток.

5. Функция *DEL* удаляет три первых структуры из файла по желанию пользователя. Здесь создаётся временный динамический массив, размер которого *N – 3,* куда записываются «оставшиеся» элементы (структуры)

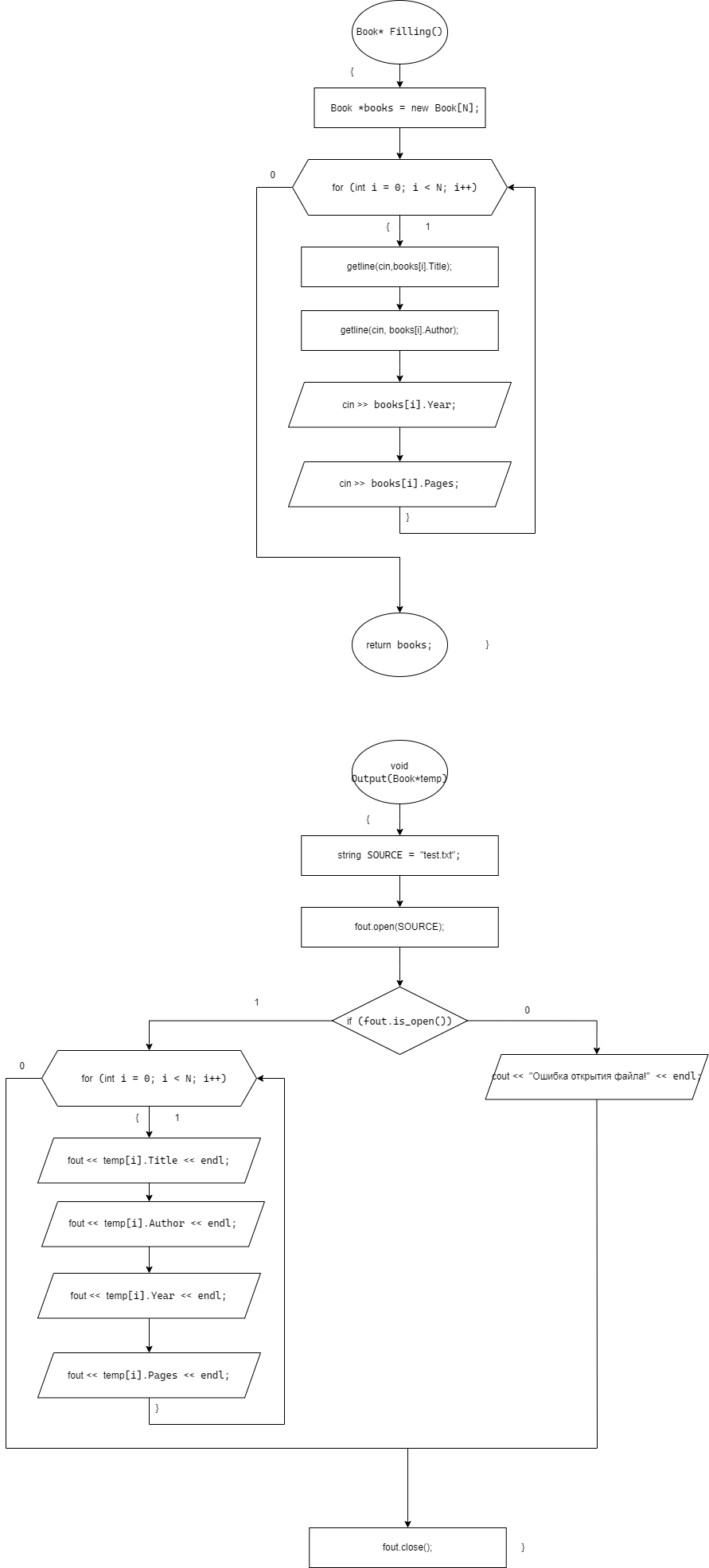
6. *ADD* – функция, в которой происходит добавление элемента по желанию пользователя. Вводится название книги, перед которой необходимо вставить «новые» данные о другой книге, и они также добавляются в массив.

# **Блок – схема**

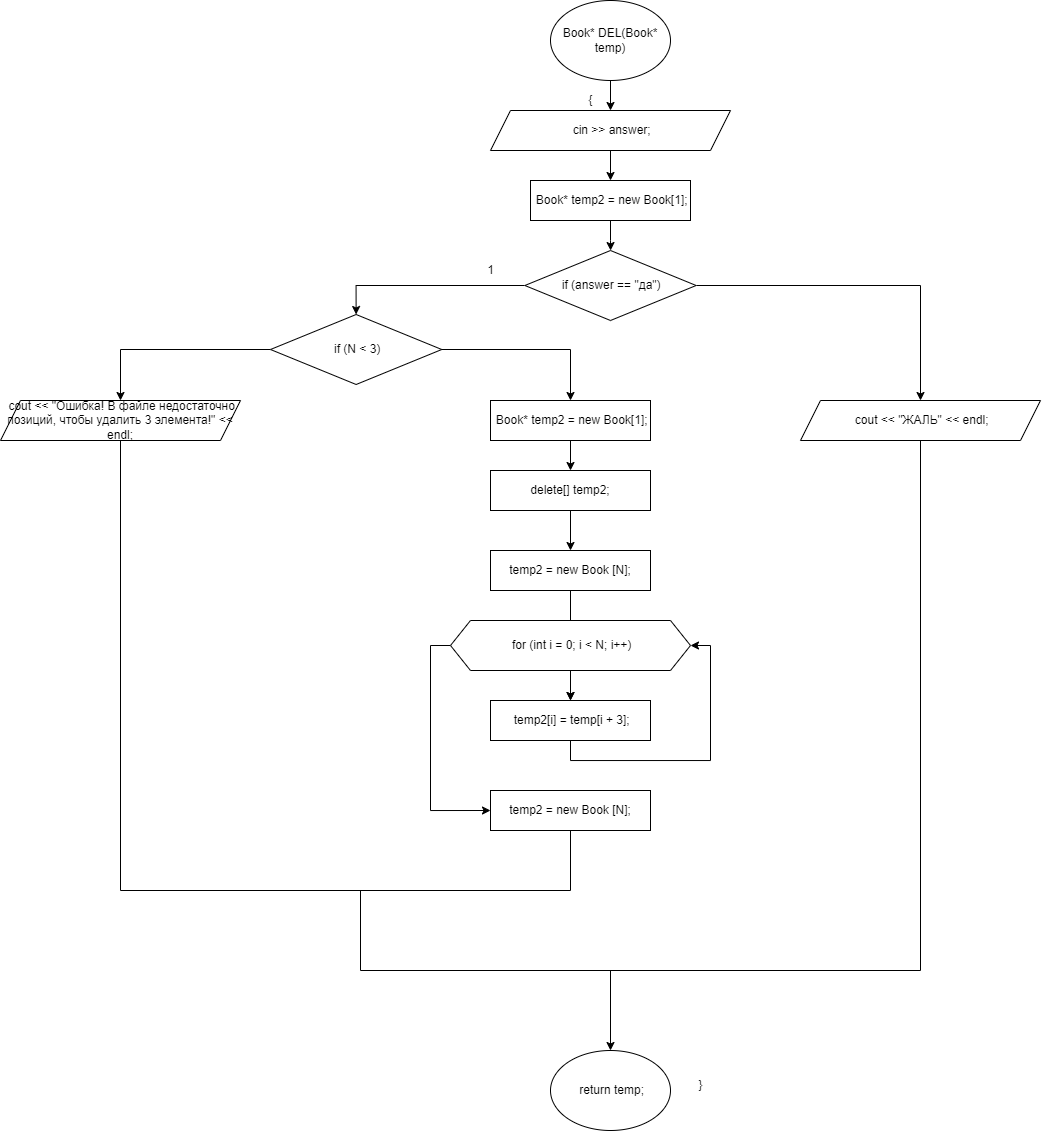
A) Функция *Filling( ):*

**

B) Функция *Output( ):*

**

C) Функция *DEL():*

****

**Приложение А**

**Листинг программы**

#include <iostream>

#include <string>

#include <fstream>

using namespace std;

int N; // длина диамического массива структур

struct Book

{

string Title;

string Author;

int Year;

int Pages;

};

ofstream fout;

Book\* Filling() //создание и заполнение массива структур

{

cout << "Введите количество экземпляров: ";

cin >> N;

Book \*books = new Book[N];

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

{

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "Название: ";

getline(cin, books[i].Title);

getline(cin, books[i].Title);

cout << "Автор: ";

getline(cin, books[i].Author);

cout << "Год издания: ";

cin >> books[i].Year;

cout << "Количество страниц: ";

cin >> books[i].Pages;

cout << endl;

}

return books;

}

void Output(Book\*temp)

{

string SOURCE = "test.txt";

fout.open(SOURCE);

if (fout.is\_open())

{

cout << "OUTPUT" << endl;

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

{

fout << "Название: ";

fout << temp[i].Title << endl;

fout << "Автор: ";

fout << temp[i].Author << endl;

fout << "Год издания: ";

fout << temp[i].Year << endl;

fout << "Количество страниц: ";

fout << temp[i].Pages << endl;

fout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

fout << endl;

}

}

else

{

cout << "Ошибка открытия файла!" << endl;

}

fout.close();

}

Book\* DEL(Book\* temp)

{

cout << "Хотите удалить 3 элемента из начала файла?" << endl;

string answer;

cout << "да ИЛИ нет" << endl;

cin >> answer;

cin.ignore();

Book\* temp2 = new Book[1];

if (answer == "да")

{

if (N < 3)

{

cout << "Ошибка! В файле недостаточно позиций, чтобы удалить 3 элемента!" << endl;

}

else

{

N -= 3;

delete[] temp2;

temp2 = new Book [N];

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

{

temp2[i] = temp[i + 3];

}

temp = temp2;

}

}

else

{

cout << "ЖАЛЬ" << endl;

}

return temp;

}

Book\* ADD(Book\* temp)

{

cout << "Хотите добавить информацию о книге?" << endl;

string answer;

cout << "да ИЛИ нет" << endl;

cin >> answer;

cin.ignore();

string BookName;

cout << "Введите название книги, перед которой вы хотите добавить позцию: ";

cin >> BookName;

Book\* temp2 = new Book[1];

if (answer == "да")

{

bool fl = false; // найдена ли книга с таким названием

int i = 0;

int index = 0;

while (fl == false and i < N)

{

if (temp[i].Title == BookName)

{

fl = true;

index = i;

}

i++;

}

if (fl == false)

{

cout << "Книги с таким названием не найдено!" << endl;

}

else

{

N++;

delete[] temp2;

temp2 = new Book[N];

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

temp2[i] = temp[i];

cout << "Название: ";

getline(cin, temp2[index].Title);

getline(cin, temp2[index].Title);

cout << "Автор: ";

getline(cin, temp2[index].Author);

cout << "Год издания: ";

cin >> temp2[index].Year;

cout << "Количество страниц: ";

cin >> temp2[index].Pages;

cout << "------------------------------------------------------------" << endl;

cout << endl;

for (int i = index + 1; i < N; i++)

temp2[i] = temp[i - 1];

temp = temp2;

}

}

else

{

cout << "В другой раз!" << endl;

}

return temp;

}

int main()

{

system("chcp 1251>NULL");

Book\* books = Filling();

Output(books); // вывод массива структур изначально заданных

books = DEL(books);

Output(books);

books = ADD(books);

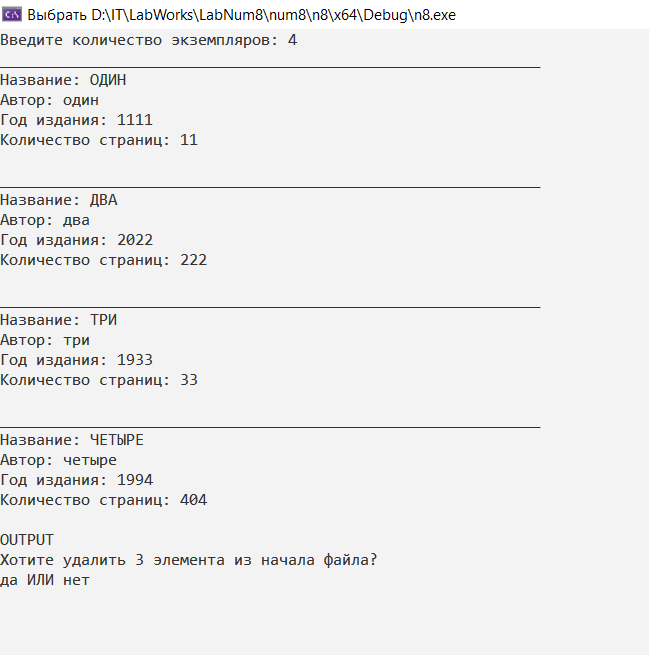
Output(books);

}

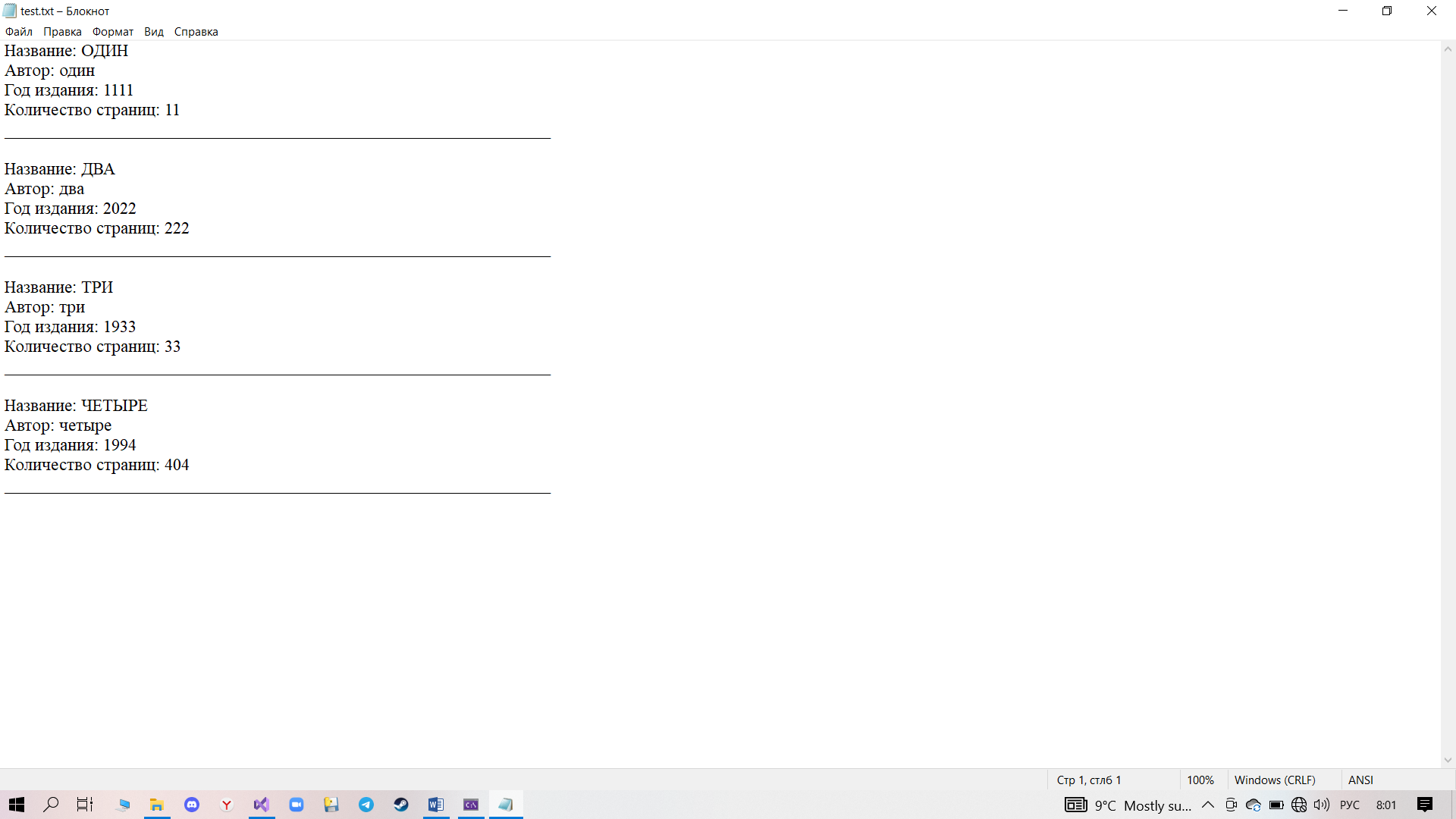
**Приложение Б**

**Результаты выполнения программы**

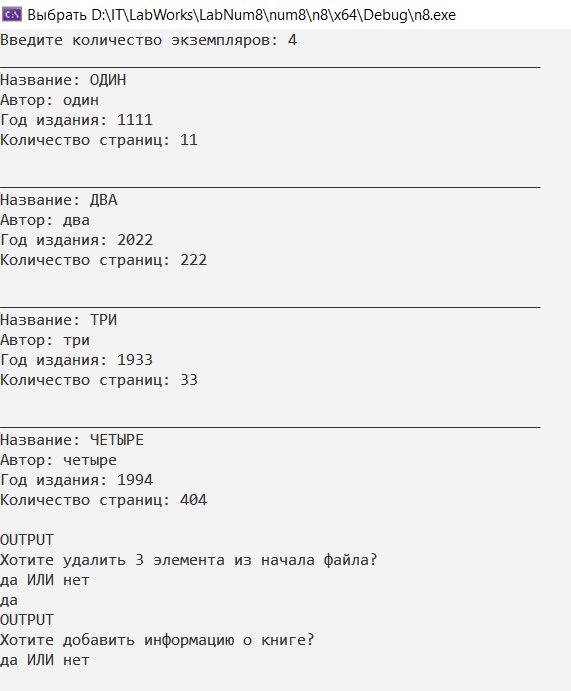
1) Ввод данных через консоль



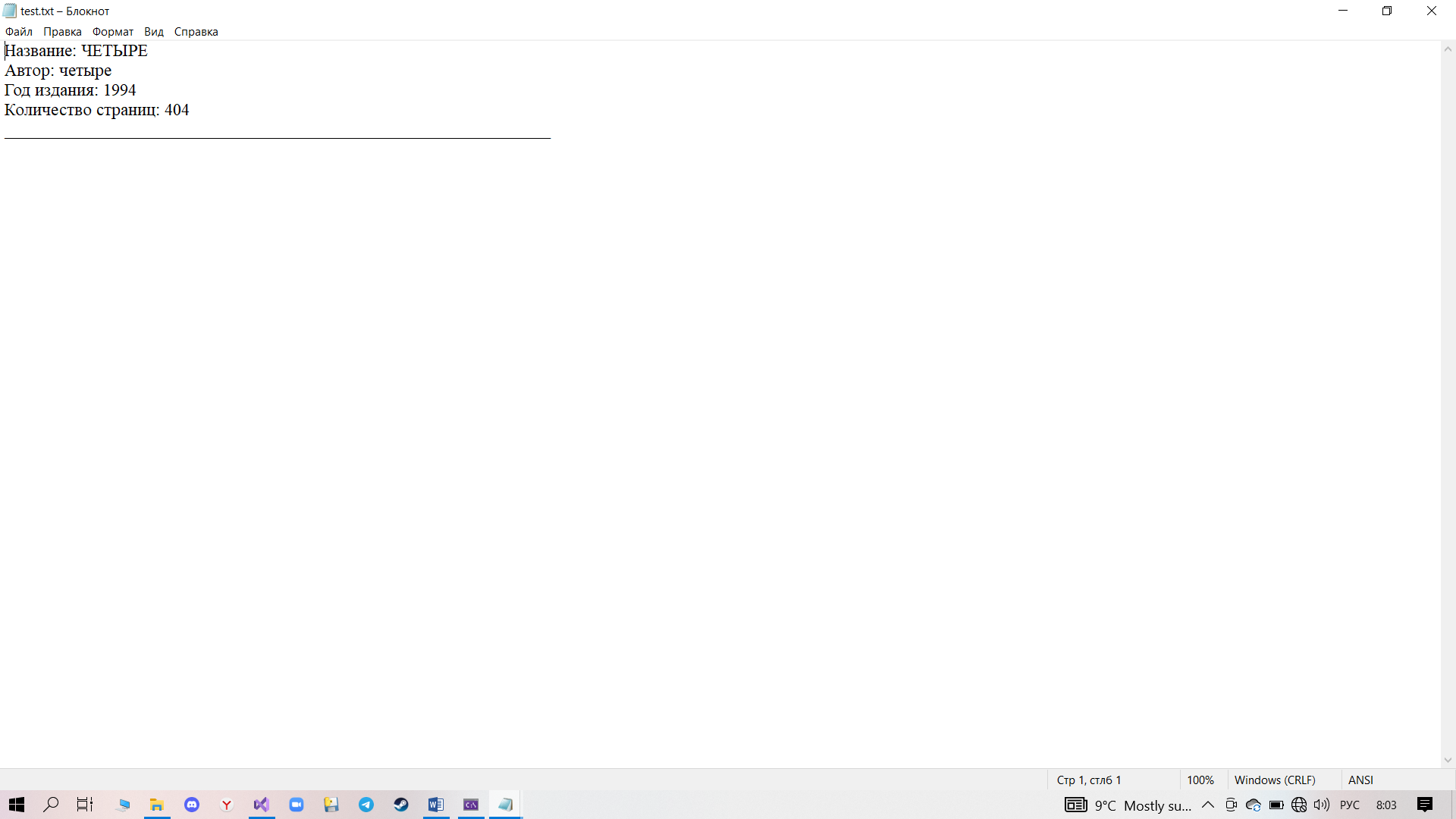
2) Запись данных в файл



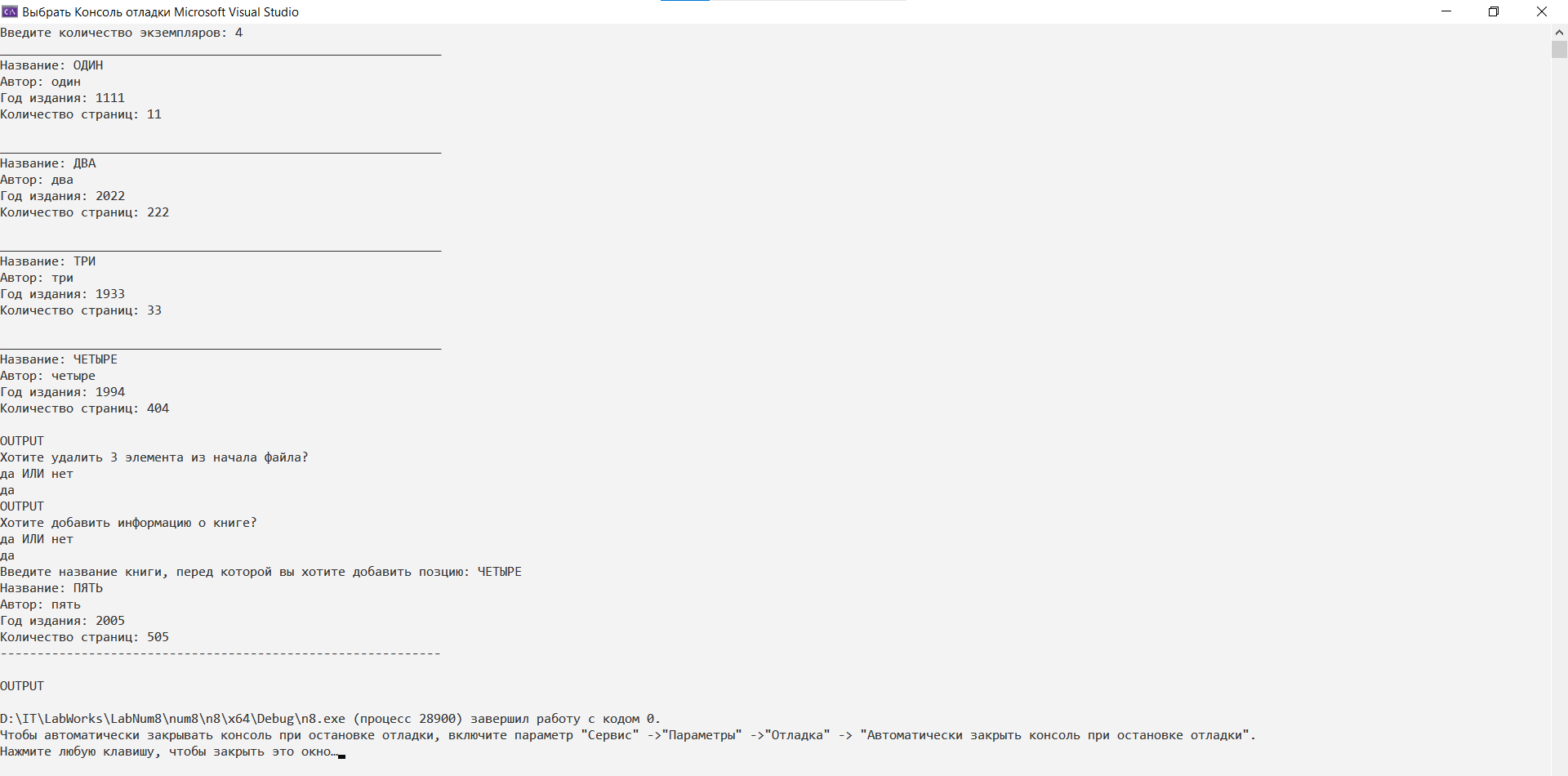
3) Возможность удаления первых трёх структур в файле



4) Изменения в файле



5) Возможность добавления структуры в файл



6) Изменения в файле

