Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №8**

Дисциплина: основы алгоритмизации и программирования

Тема: Блоковый ввод-вывод

Выполнил работу

студент группы РИС-20-1б

Ремянников А. В.

Проверила

Доцент кафедры ИТАС

к.т.н Полякова О. А.

Пермь, 2021

**Цель работы**

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

**Постановка задачи**

(Вариант 19) Сформировать двоичный файл из элементов, заданной в варианте структуры, распечатать его содержимое, выполнить удаление и добавление элементов в соответствии со своим вариантом, используя для поиска удаляемых или добавляемых элементов функцию. Формирование, печать, добавление и удаление элементов оформить в виде функций. Предусмотреть сообщения об ошибках при открытии файла и выполнении операций ввода/вывода.

Структура "Фильм":

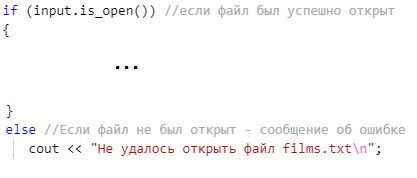
* название;
* режиссер;
* страна;
* приносимая прибыль.

Удалить 2 элемента из конца файла, добавить элемент после элемента с указанным названием.

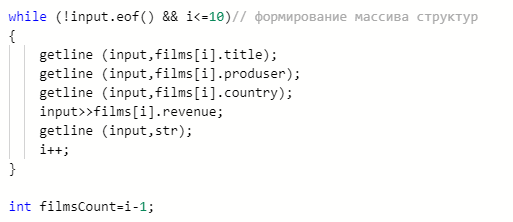
**Анализ задачи**

**1.** Для решения задачи необходимо:

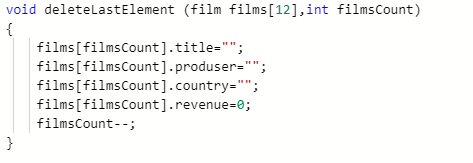
**1.1.** Проверить, существует ли файл ввода и в случае его отсутствия сообщить об ошибке.



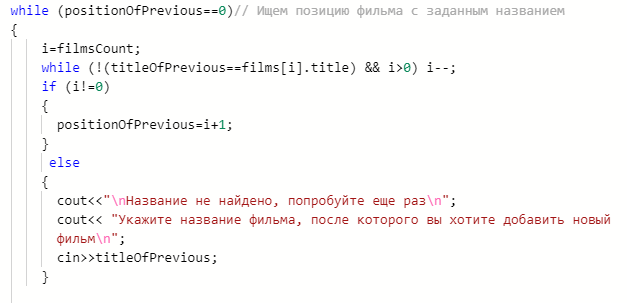
**1.2.** Создать структуру и заполнить ее данными из файла Films.txt с входными данными и запомнить количество фильмов.



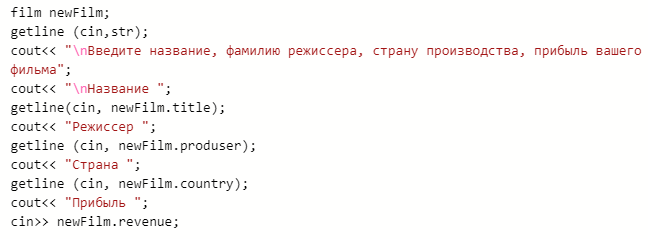
**1.3.** Разработать функцию, позволяющую удалять элементы массива из структур film.



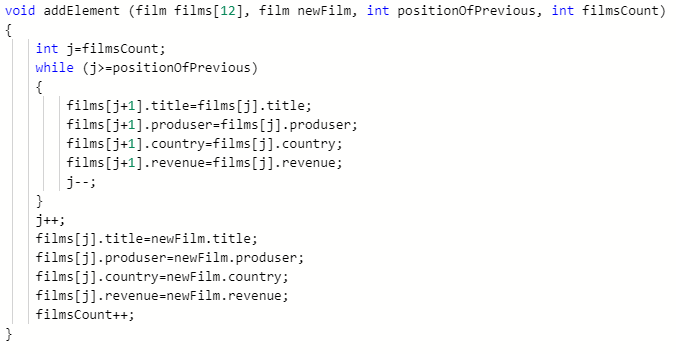
**1.4.** Разработать код, по поиску нужного элемента по названию фильма.



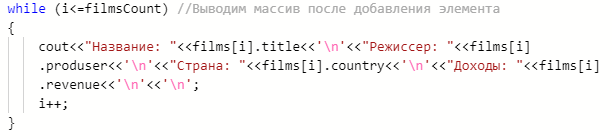
**1.5.** Позволить пользователю ввести данные о новом фильме.

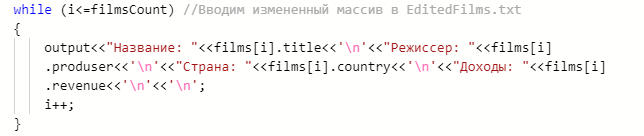


**1.6.** Разработать функцию, добавляющую на указанную позицию новый элемент, введенный с клавиатуры.



**1.7.** Обеспечить вывод информации на консоль и в файл EditedFilms.txt



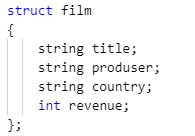


**2.** В ходе работы были использованы следующие типы данных:

**2.1.** Для хранения информации о фильмах используется массив films, состоящий из структур film:



**2.2.** Структура film: Для записи названия, имени режиссера и страны используется тип string, для доходов- int:



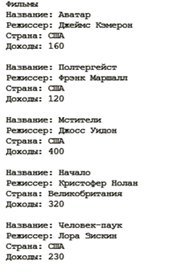
**2.3.** Для вывода данных в файл используется класс ofstream:



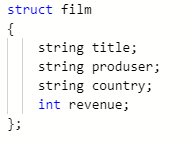
**2.4.** Для получения данных из файла используется класс ifstream:



**3.** Для решения задачи данные будут представлены в виде списка с пояснениями для каждой строки:

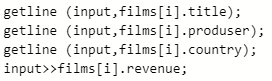


**4.** При работе со структурой использовались следующие поля:

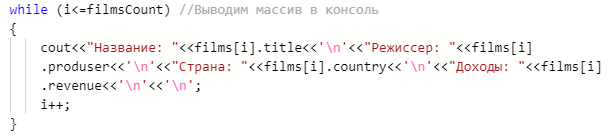


**5.** Для операций ввода и вывода использовались следующие операторы и функции:

**5.1.** Ввод данных из файла Films.txt реализован с помощью функции getline, для поля revenue использован оператор ввода из файла:

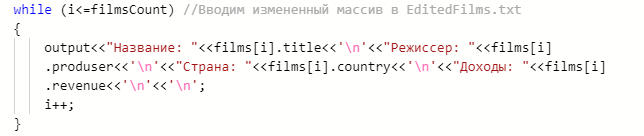


**5.2.** Вывод данных на консоль реализован с помощью оператора cout:



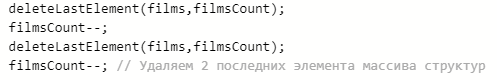


**5.3.** Вывод данных в файл реализован с помощью оператора вывода в файл:

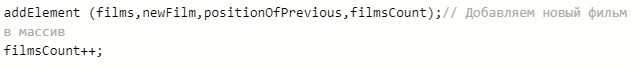


**6.** Поставленные задачи будут решены следующими действиями:

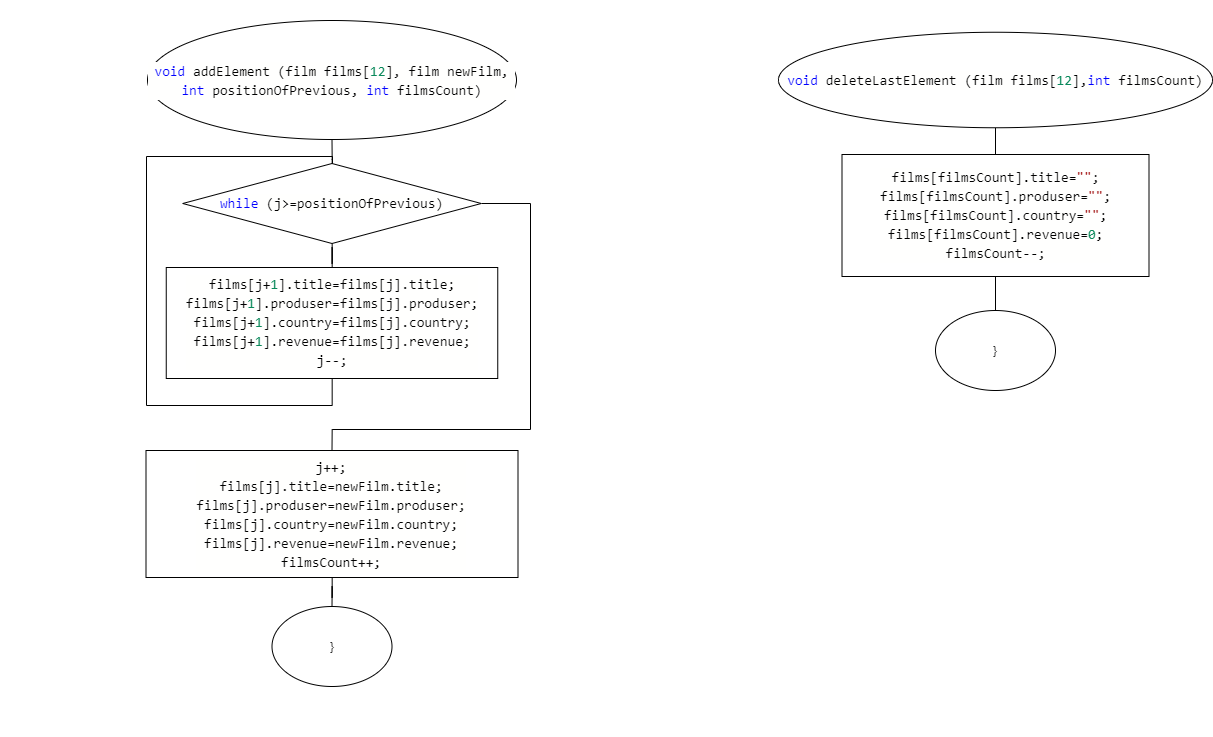
**6.1.** Для удаления 2х последних элементов массива использована функция deleteLastElement:

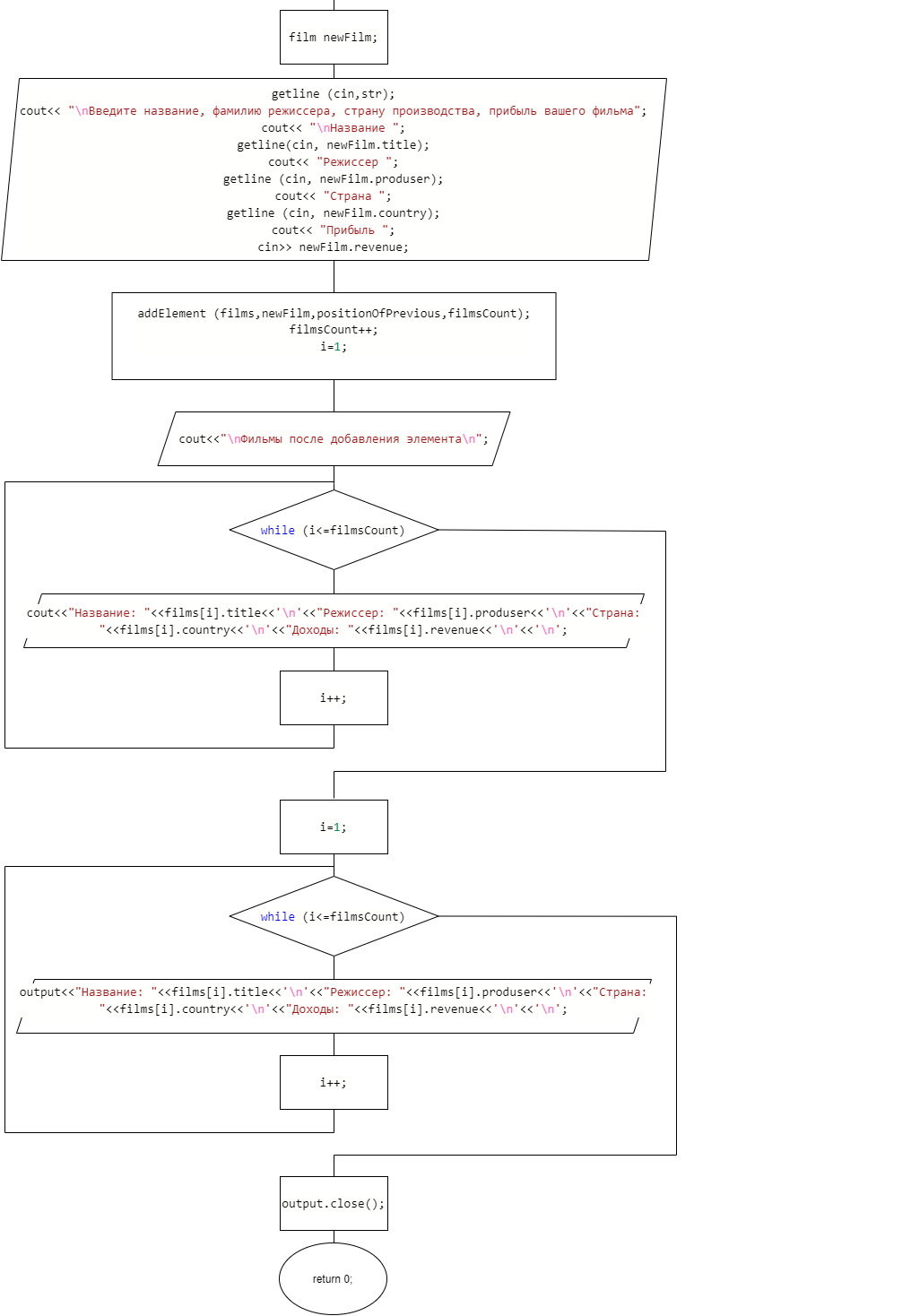
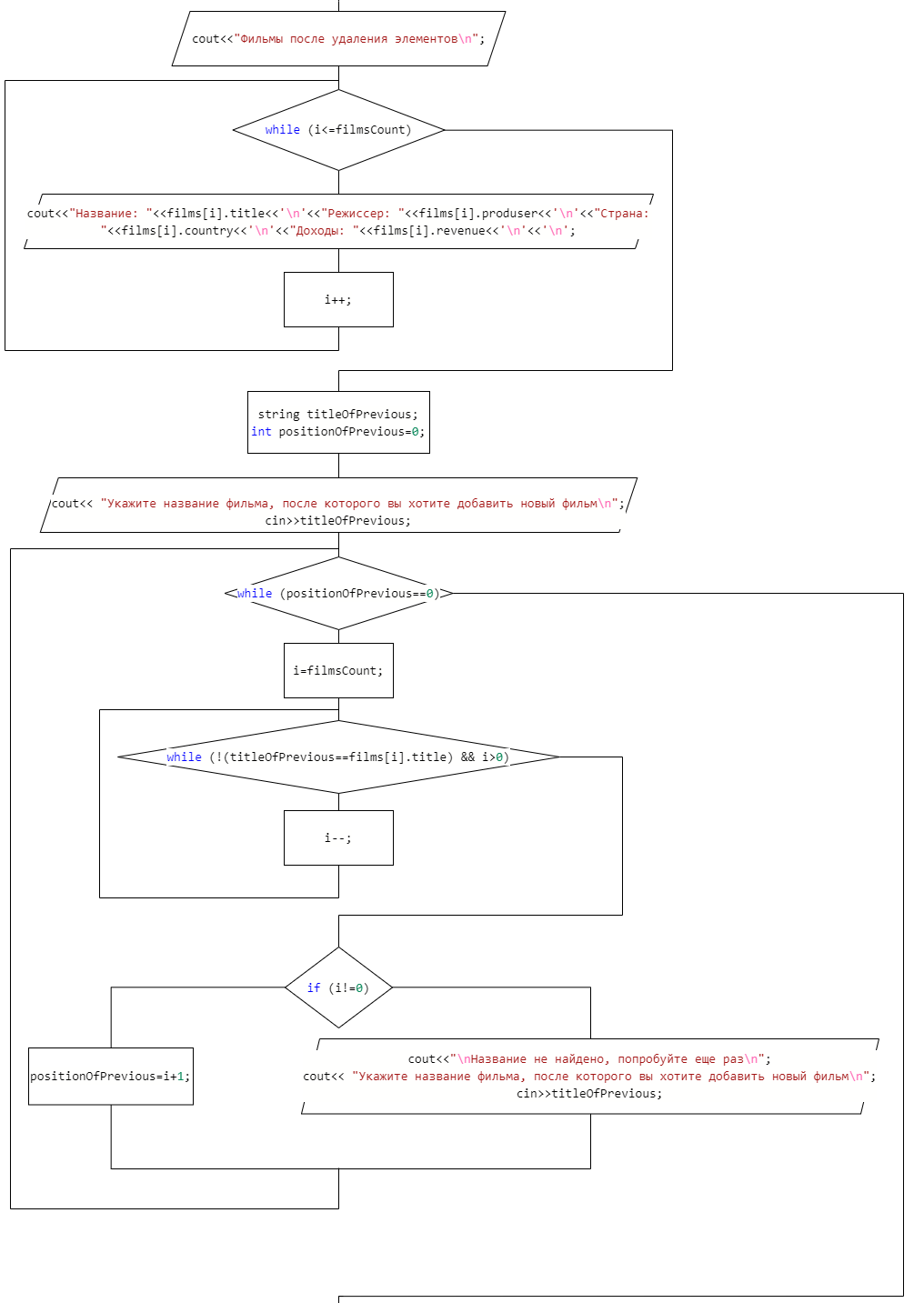
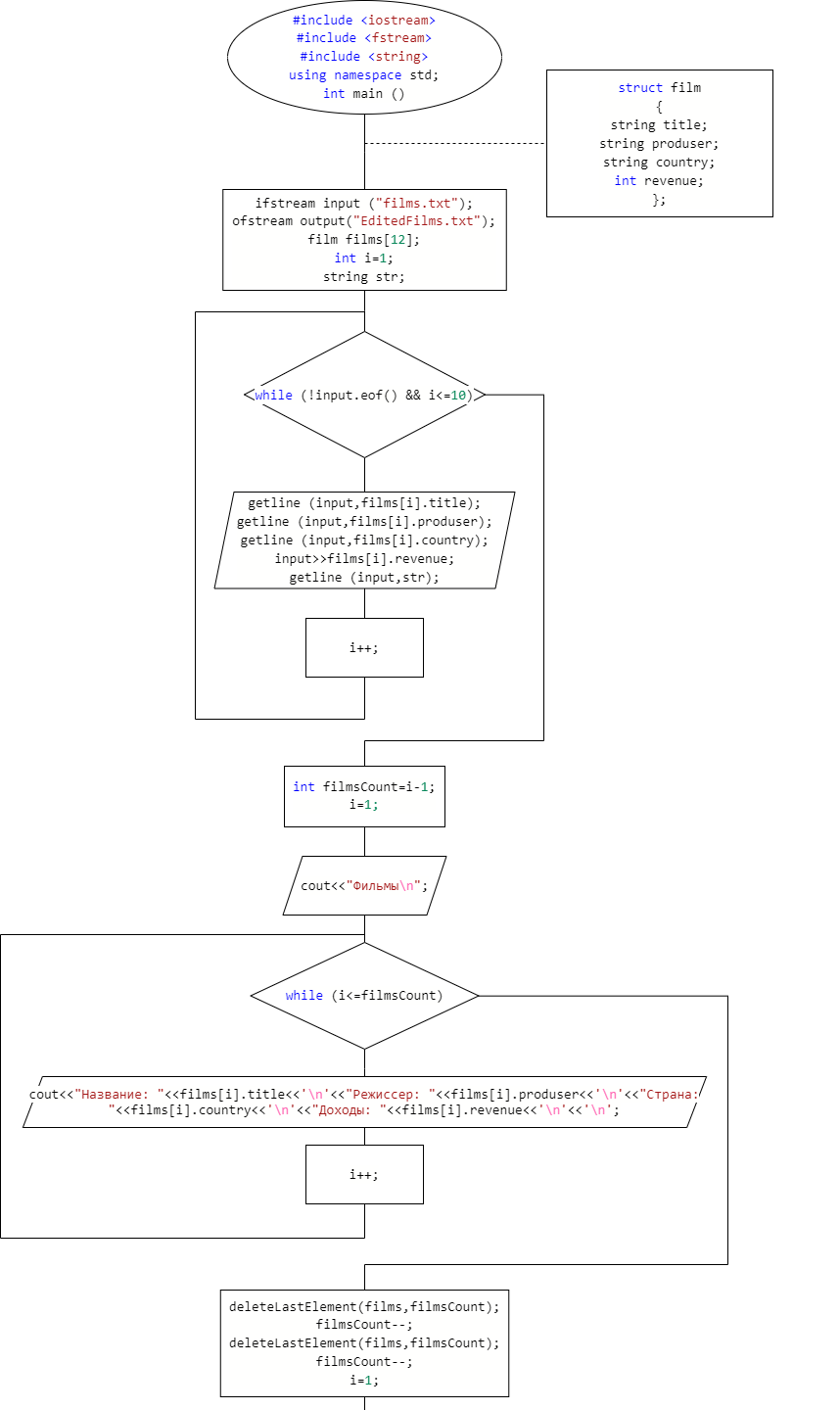


**6.2.** Для добавления элемента используется функция addElement:



**Блок-схема программы**

****

****

**Решение**

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

struct film

{

string title;

string produser;

string country;

int revenue;

};

void deleteLastElement (film[12],int); // Удаление последнего элемента массива структур

void addElement (film[12],film,int,int); // Добавление элемента в массив структур

int main ()

{

setlocale(LC\_ALL, "Russian");

ifstream input ("films.txt");

if (input.is\_open()) //если файл был успешно открыт

{

ofstream output("EditedFilms.txt"); //открываем файл для записи

film films[12];

int i=1;

string str;

while (!input.eof() && i<=10)// формирование массива структур

{

getline (input,films[i].title);

getline (input,films[i].produser);

getline (input,films[i].country);

input>>films[i].revenue;

getline (input,str);

i++;

}

int filmsCount=i-1;

i=1;

cout<<"Фильмы\n";

while (i<=filmsCount) //Выводим массив в консоль

{

cout<<"Название: "<<films[i].title<<'\n'<<"Режиссер: "<<films[i].produser<<'\n'<<"Страна: "<<films[i].country<<'\n'<<"Доходы: "<<films[i].revenue<<'\n'<<'\n';

i++;

}

deleteLastElement(films,filmsCount);

filmsCount--;

deleteLastElement(films,filmsCount);

filmsCount--; // Удаляем 2 последних элемента массива структур

i=1;

cout<<"Фильмы после удаления элементов\n";

while (i<=filmsCount) //Выводим массив после удаления элементов

{

cout<<"Название: "<<films[i].title<<'\n'<<"Режиссер: "<<films[i].produser<<'\n'<<"Страна: "<<films[i].country<<'\n'<<"Доходы: "<<films[i].revenue<<'\n'<<'\n';

i++;

}

cout<< "Укажите название фильма, после которого вы хотите добавить новый фильм\n";

string titleOfPrevious;

int positionOfPrevious=0;

cin>>titleOfPrevious;

while (positionOfPrevious==0)// Ищем позицию фильма с заданным названием

{

i=filmsCount;

while (!(titleOfPrevious==films[i].title) && i>0) i--;

if (i!=0)

{

positionOfPrevious=i+1;

}

else

{

cout<<"\nНазвание не найдено, попробуйте еще раз\n";

cout<< "Укажите название фильма, после которого вы хотите добавить новый фильм\n";

cin>>titleOfPrevious;

}

}

film newFilm;

getline (cin,str);

cout<< "\nВведите название, фамилию режиссера, страну производства, прибыль вашего фильма";

cout<< "\nНазвание ";

getline(cin, newFilm.title);

cout<< "Режиссер ";

getline (cin, newFilm.produser);

cout<< "Страна ";

getline (cin, newFilm.country);

cout<< "Прибыль ";

cin>> newFilm.revenue;

addElement (films,newFilm,positionOfPrevious,filmsCount);// Добавляем новый фильм в массив

filmsCount++;

cout<<"\nФильмы после добавления элемента\n";

i=1;

while (i<=filmsCount) //Выводим массив после добавления элемента

{

cout<<"Название: "<<films[i].title<<'\n'<<"Режиссер: "<<films[i].produser<<'\n'<<"Страна: "<<films[i].country<<'\n'<<"Доходы: "<<films[i].revenue<<'\n'<<'\n';

i++;

}

i=1;

while (i<=filmsCount) //Вводим измененный массив в EditedFilms.txt

{

output<<"Название: "<<films[i].title<<'\n'<<"Режиссер: "<<films[i].produser<<'\n'<<"Страна: "<<films[i].country<<'\n'<<"Доходы: "<<films[i].revenue<<'\n'<<'\n';

i++;

}

output.close(); //Закрываем файл для записи

}

else //Если файл не был открыт - сообщение об ошибке

cout << "Не удалось открыть файл films.txt\n";

input.close(); //Закрываем файл с исходными данными

}

void deleteLastElement (film films[12],int filmsCount)

{

films[filmsCount].title="";

films[filmsCount].produser="";

films[filmsCount].country="";

films[filmsCount].revenue=0;

filmsCount--;

}

void addElement (film films[12], film newFilm, int positionOfPrevious, int filmsCount)

{

int j=filmsCount;

while (j>=positionOfPrevious)

{

films[j+1].title=films[j].title;

films[j+1].produser=films[j].produser;

films[j+1].country=films[j].country;

films[j+1].revenue=films[j].revenue;

j--;

}

j++;

films[j].title=newFilm.title;

films[j].produser=newFilm.produser;

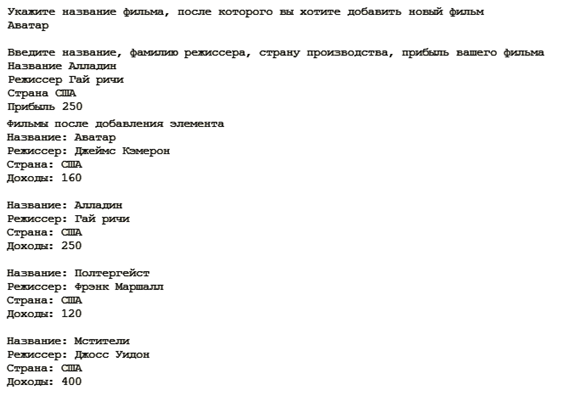
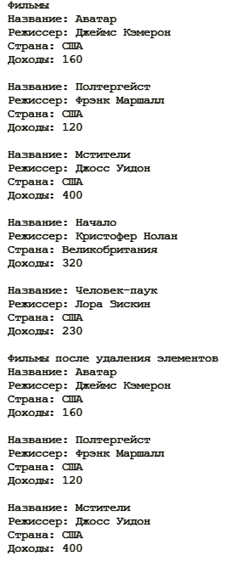
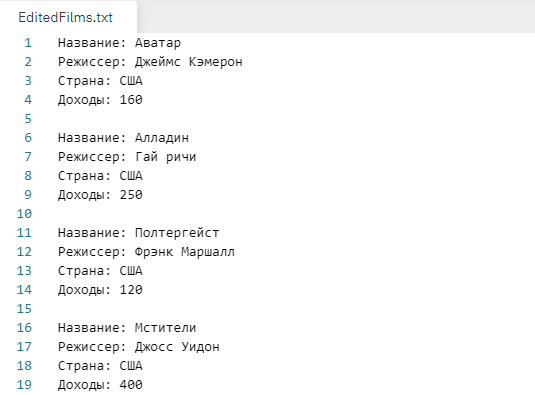
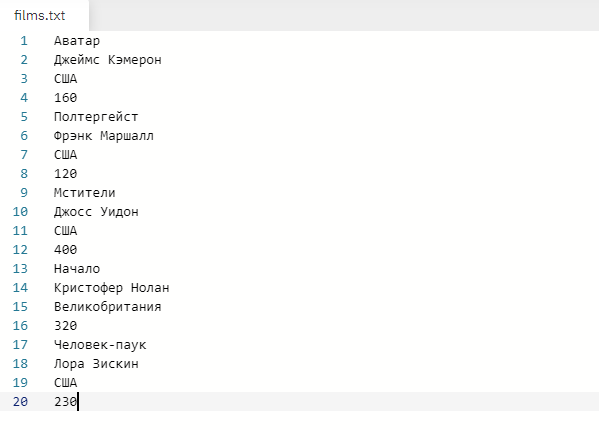
films[j].country=newFilm.country;

films[j].revenue=newFilm.revenue;

filmsCount++;

}

**Скриншоты результатов работы программы**

****

**GitHub**

<https://github.com/sugarrrfqs/pnrpuLab8>