Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет

**Лабораторная работа №8**

**"Блоковый ввод-вывод"**

**19 вариант**

Выполнила: студентка   
гр. ИВТ-20-2б

Лялина Анастасия Владиславовна

Проверила: доцент кафедры

ИТАС Полякова О. А.

Пермь

2020

**Постановка задачи**

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

* 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. Будут использоваться следующие типы данных:

Name - Название фильма, тип - string;

Producer - Имя режиссера, тип - string;

Country - Страна производства, тип - string;

Income - Приносимая прибыль, тип - int;

n - Количество элементов структуры, тип - int;

r – Номер элемента структуры, после которого будет вставлен новый элемент, тип – int;

i – Вспомогательная переменная для работы циклов, тип – int.

1. При работе с данными первоначально будет создана структура Film. После этого будет считана переменная n и заполнена структура из n элементов. Затем удаляются последние два элемента структуры, считывается переменная r и заполняется новый элемент структуры, записывающийся после элемента под номером r. В конце вся полученная структура выводится на экран и в выходной файл Result.txt.
2. Для решения будет представлена структура данных Film и файл Result.txt.
3. Структура Film имеет 4 поля (Name, Producer, Country, Income) и 3 метода:
   1. Input – используется для заполнения элемента структуры;
   2. Output – используется для вывода полей элемента структуры в консоль;
   3. DeletePrelasts – используется для удаления полей элемента структуры.
4. Ввод будет организован операторами cin и getline:
   1. Ввод переменных n,r, и полей Income осуществляется с помощью cin;
   2. Ввод полей Name, Producer, Country осуществляется с помощью getline();

Вывод будет организован оператором cout и записью в текстовый файл Result.txt:

1. Все выводы внутри программы осуществляется с помощью cout;
2. В конце работы программы вся структура записывается в файл Result.txt с помощью ofstream.
3. В ходе работы будет создана новая функция adding, которая с определенного элемента структуры смещает все последующие на 1 вперед. Данная функция используется при записи нового элемента.

В программе используется цикл while для проверки правильности ввода количества элементов - n и номера элемента r.

while (n <= 2 || n > 100)

{

cout << "Enter the amount of films (Min is 3, Max is 100): ";

cin >> n;

}  
Также несколько раз используется цикл for – когда идет работа со всеми элементами структуры.

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

{

arr[i].Input();

}

**Код программы**

#include <iostream>

#include <string>

#include <Windows.h>

#include <fstream>

using namespace std;

struct Film

{

string Name;

string Producer;

string Country;

int Income;

void Input()

{

cout << "Enter the name of the film: ";

getline(cin, Name);

cout << "Enter the producer of the film: ";

getline(cin, Producer);

cout << "Enter the country of the film: ";

getline(cin, Country);

cout << "Enter the income of the film: ";

cin >> Income;

cin.get();

}

void Output()

{

cout << "Name: \t" << Name << "\nProducer:\t" << Producer << "\nCountry: \t" << Country << "\nIncome: \t" << Income << "\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

}

void DeletePrelasts()

{

Name = "";

Producer = "";

Country = "";

Income = NULL;

}

};

void adding(struct Film str[101], int n, int r)

{

for (int i = n; i >= r; i--)

{

str[i + 1].Name = str[i].Name;

str[i + 1].Producer = str[i].Producer;

str[i + 1].Country = str[i].Country;

str[i + 1].Income = str[i].Income;

}

str[r].Name = "";

str[r].Producer = "";

str[r].Country = "";

str[r].Income = 0;

}

int main()

{

setlocale(LC\_ALL, "Russian");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

ofstream exit("Result.txt");

Film arr[101];

int r,n = 2;

exit << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n\tList of Films:\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

while (n <= 2 || n > 100)

{

cout << "Enter the amount of films (Min is 3, Max is 100): ";

cin >> n;

}

cin.get();

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

{

arr[i].Input();

}

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n\tPRINTING:\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

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

{

arr[i].Output();

}

cout << "Deleting the last 2 elements: \n";

arr[n - 1].DeletePrelasts();

arr[n - 2].DeletePrelasts();

n = n - 2;

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n\tPRINTING:\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

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

{

arr[i].Output();

}

cout << "Select the film: \n";

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

{

cout <<"("<< i+1 <<") "<<arr[i].Name<<endl;

}

cin >> r;

while (r <= 0 || r > n)

{

cout << "Wrong input, try again: ";

cin >> r;

}

cin.get();

n++;

adding(arr, n, r);

cout << "Enter the film: \n";

arr[r].Input();

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n\tPRINTING:\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

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

{

arr[i].Output();

exit << "Name: \t" << arr[i].Name << "\nProducer:\t" << arr[i].Producer << "\nCountry: \t" << arr[i].Country << "\nIncome: \t" << arr[i].Income << "\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

}

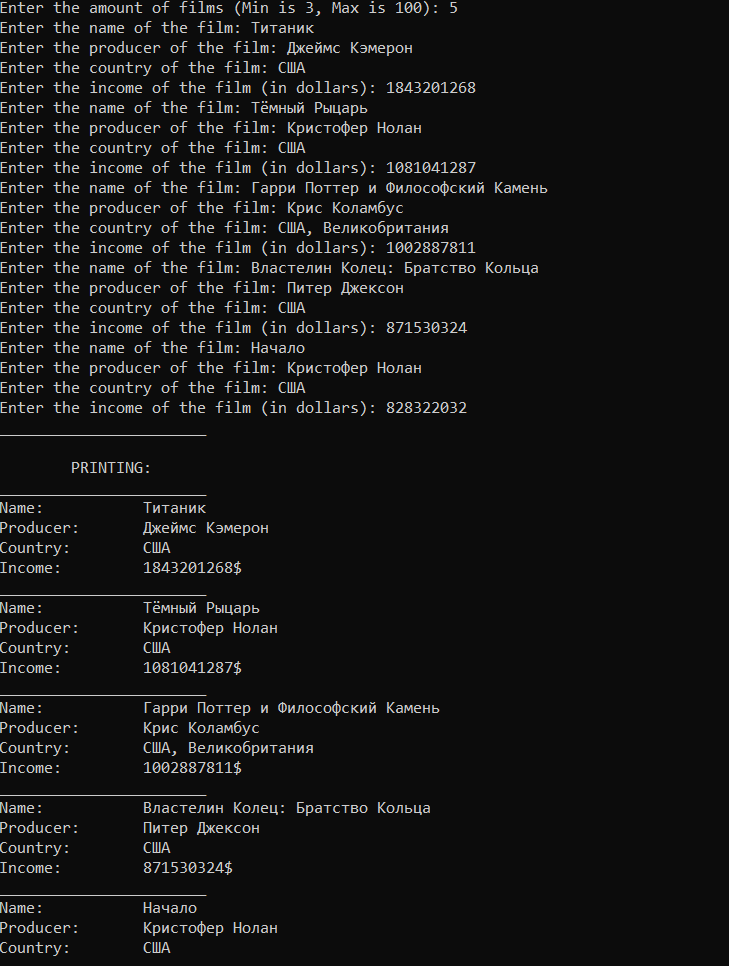
cout << "File \"Result.txt\" created.\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

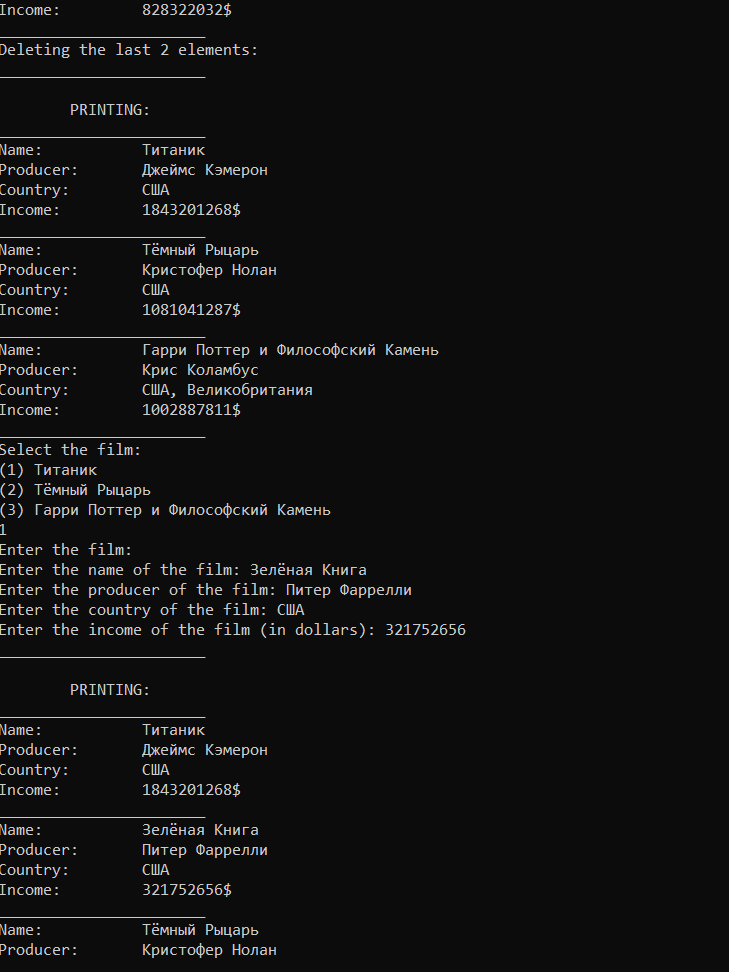
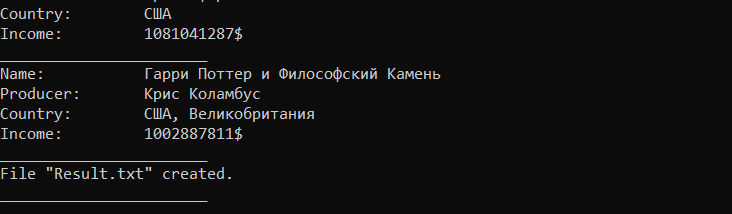
exit.close();

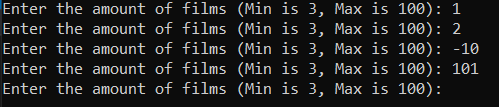
return 0;

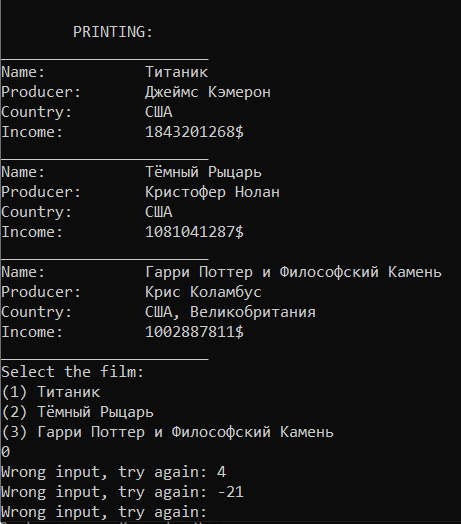
}

**Скриншоты работы программы**

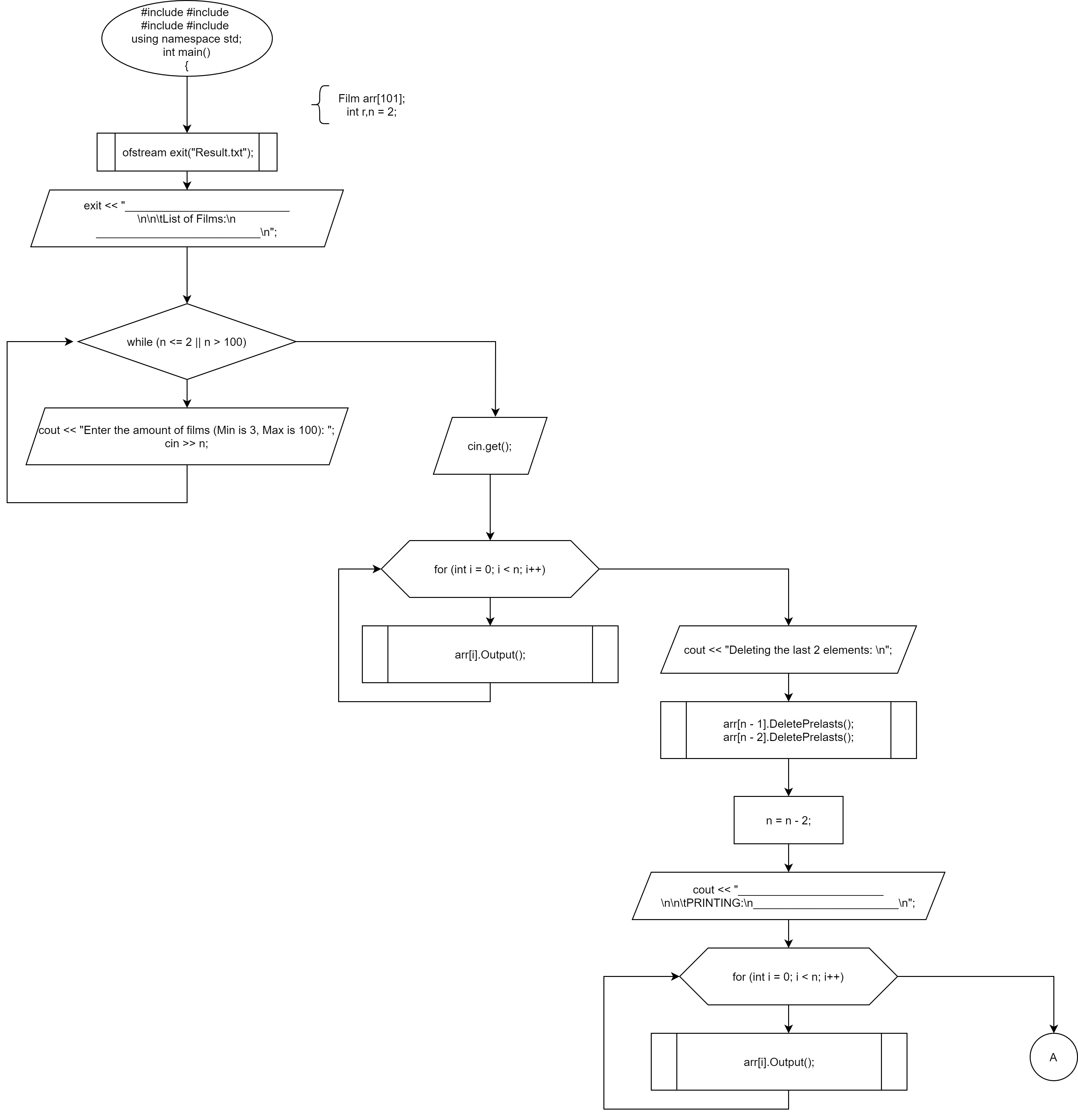


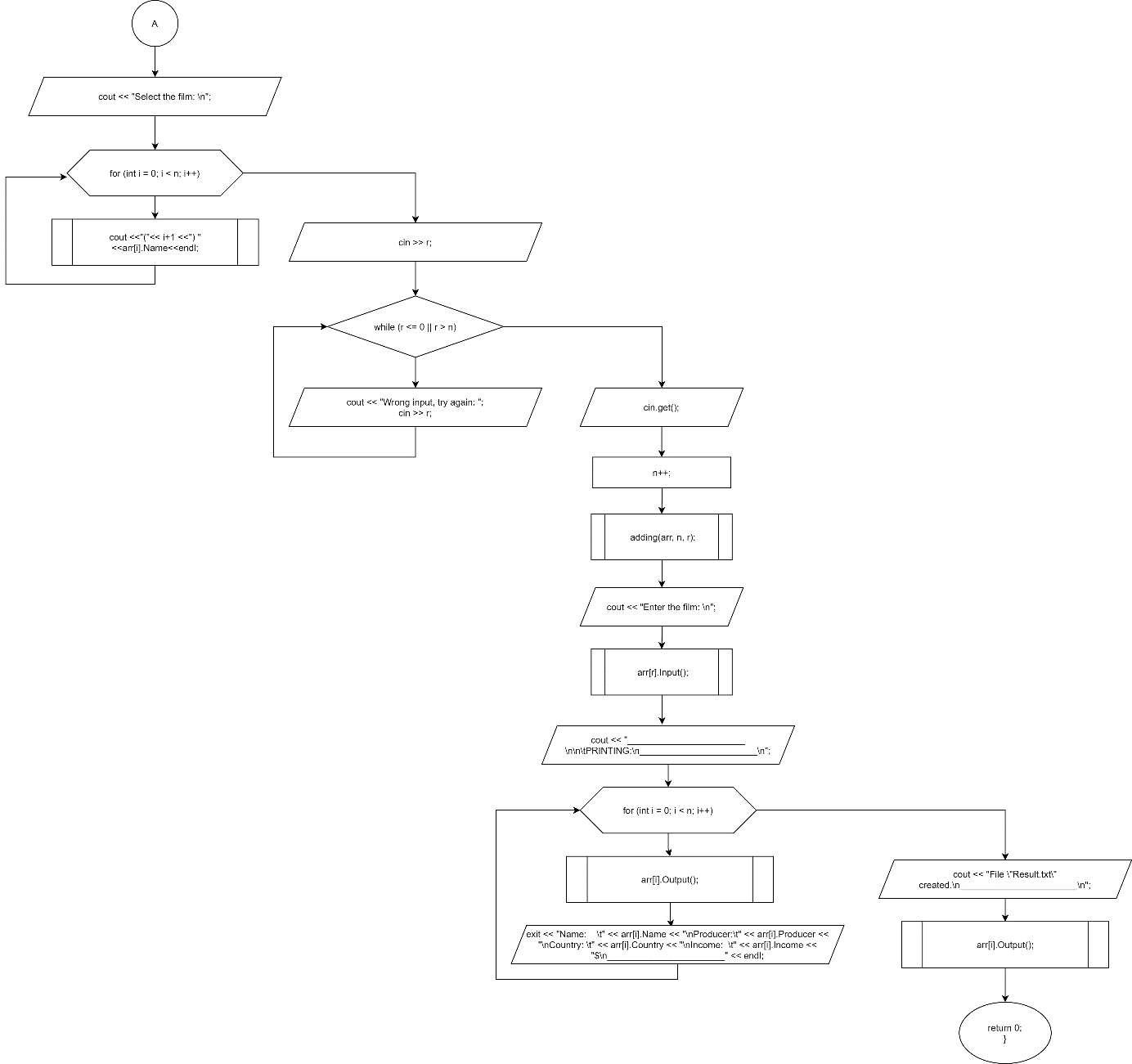


Проверка правильности ввода



**Блок-схема**



****

