Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе №12**

Дисциплина: «Основы теории алгоритмов и структуры данных»

Тема: Алгоритмы поиска. Работа со списками. Метод интерполяционного поиска

Вариант 19

Выполнил:

Студент группы РИС-20-1б

Ремянников Александр Владимирович

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2021**

**Цель работы**

Получить опыт работы с методами поиска.

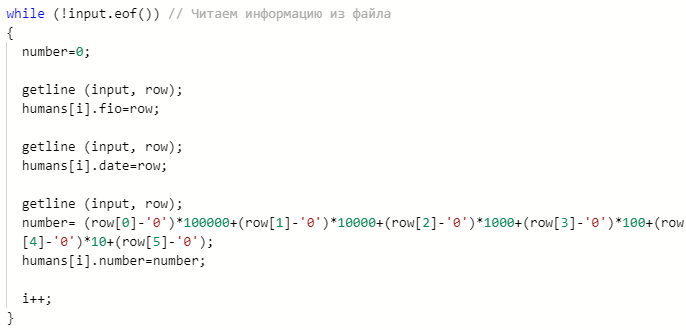
**Постановка задачи**

(Вариант 19) Выполнить поиск элемента в массиве по ключу в соответствии с вариантом. Для поиска использовать метод линейного поиска, метод Прямого поиска подстроки в строке и интерполяционный метод.

**Анализ задачи**

**1.** Для решения задачи необходимо:

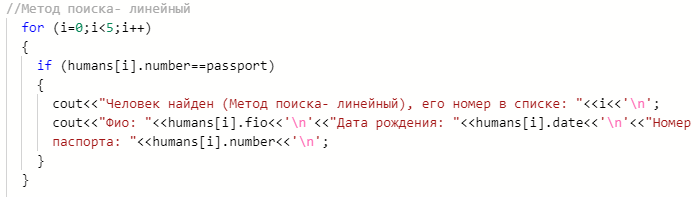
**1.1.** Считать информацию из файла и занести её в массив структур.



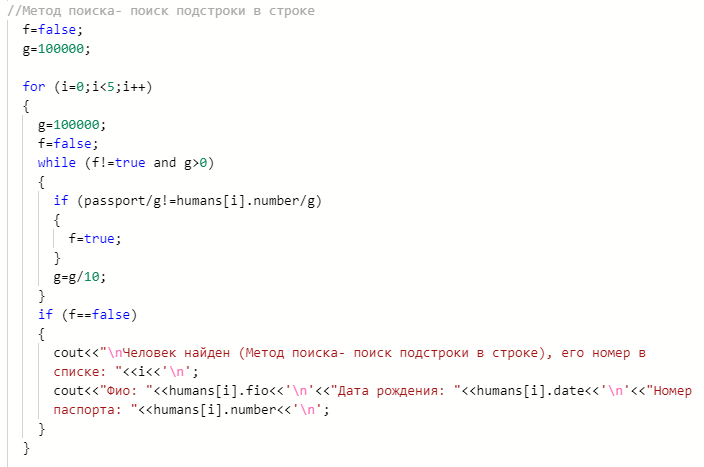
**1.2.** Предоставить пользователю возможность ввести номер паспорта человека, информацию о котором необходимо вывести.



**1.3.** Выполнить поиск методом линейного поиска и вывести результат в консоль.



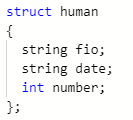
**1.4.** Выполнить поиск методом поиска подстроки в строке и вывести результат в консоль.



**1.5.** Выполнить поиск интерполяционным методом и вывести результат в консоль.

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

**2.1.** Для хранения информации о людях использовался массив из структур human.



**2.2.** Для получения информации из файла использовался класс ifstream.



**2.3.** Для извлечения из файла информации построчно использовался тип string.

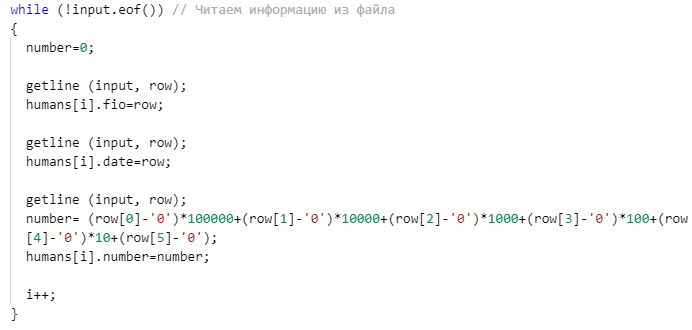


**2.4.** Для переменных-индексов использовался тип int.



**3.** Для решения задачи данные были представлены в следующем виде:

**3.1.** Для удобства поиска данные о людях были записаны в массив из структур.



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

**4.1.** Для вывода использовался оператор cout.



**4.2.** Для ввода информации из файла использовалась функция getline.

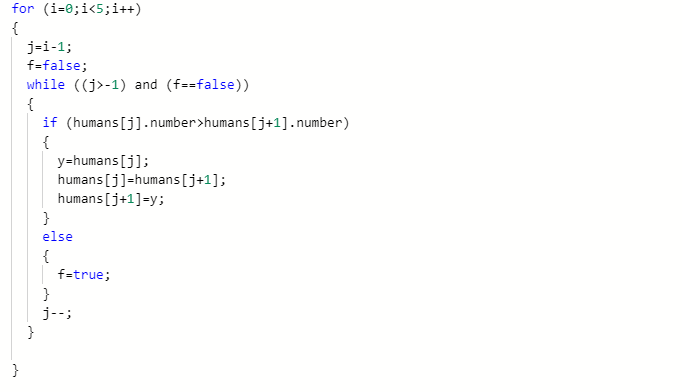


**4.3.** Для ввода информации с клавиатуры использовался оператор cin.

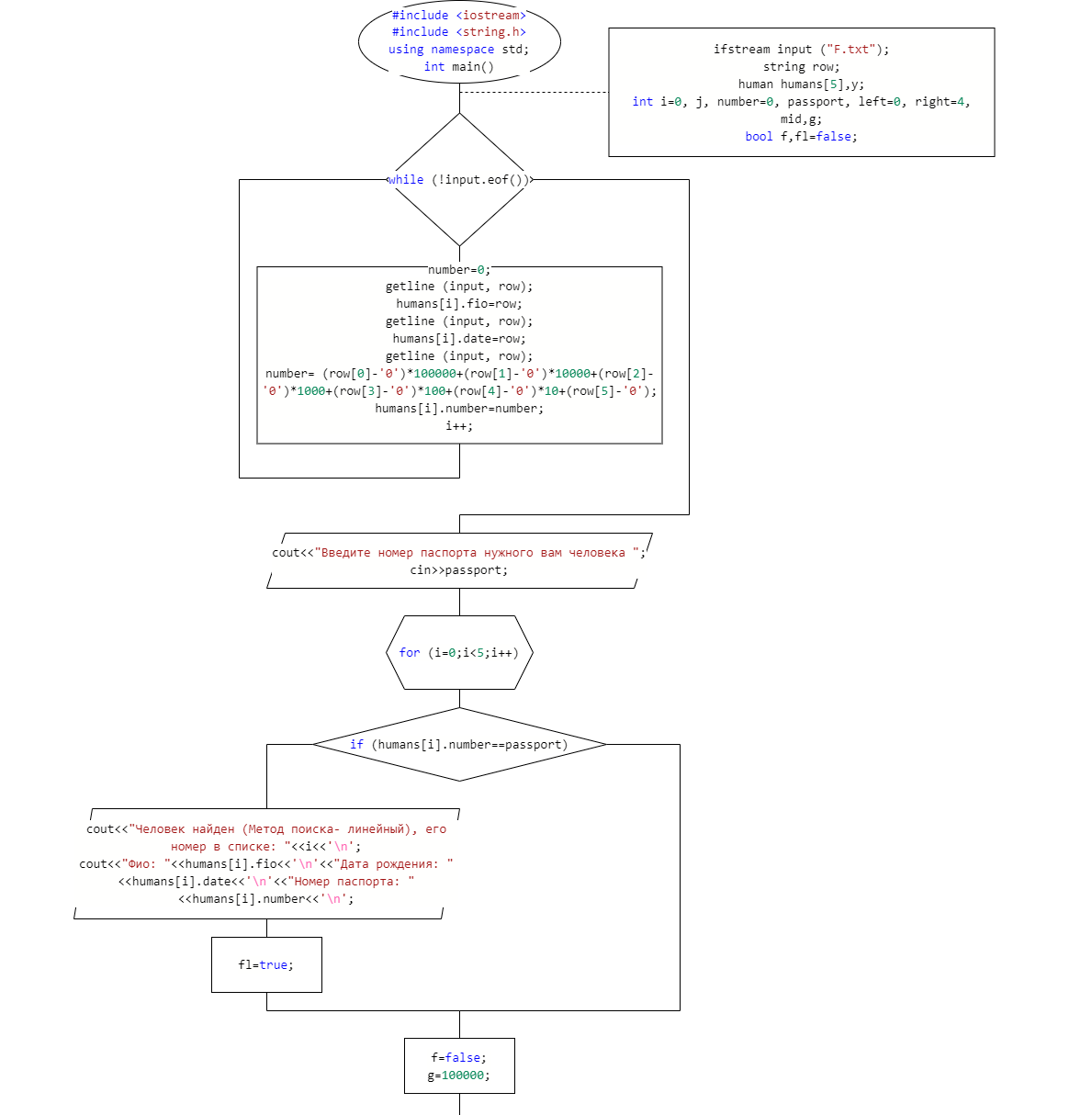


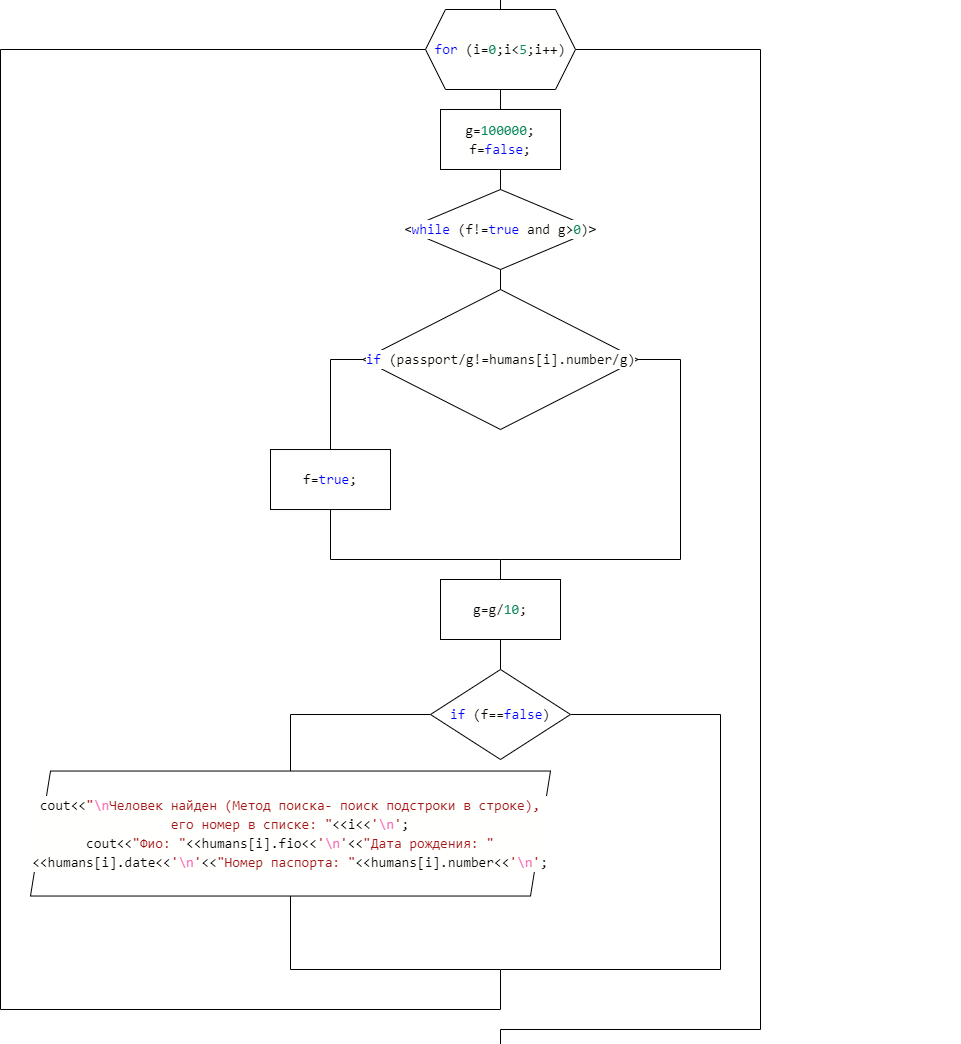
**5.** Поставленные задачи будут решены следующими действиями:

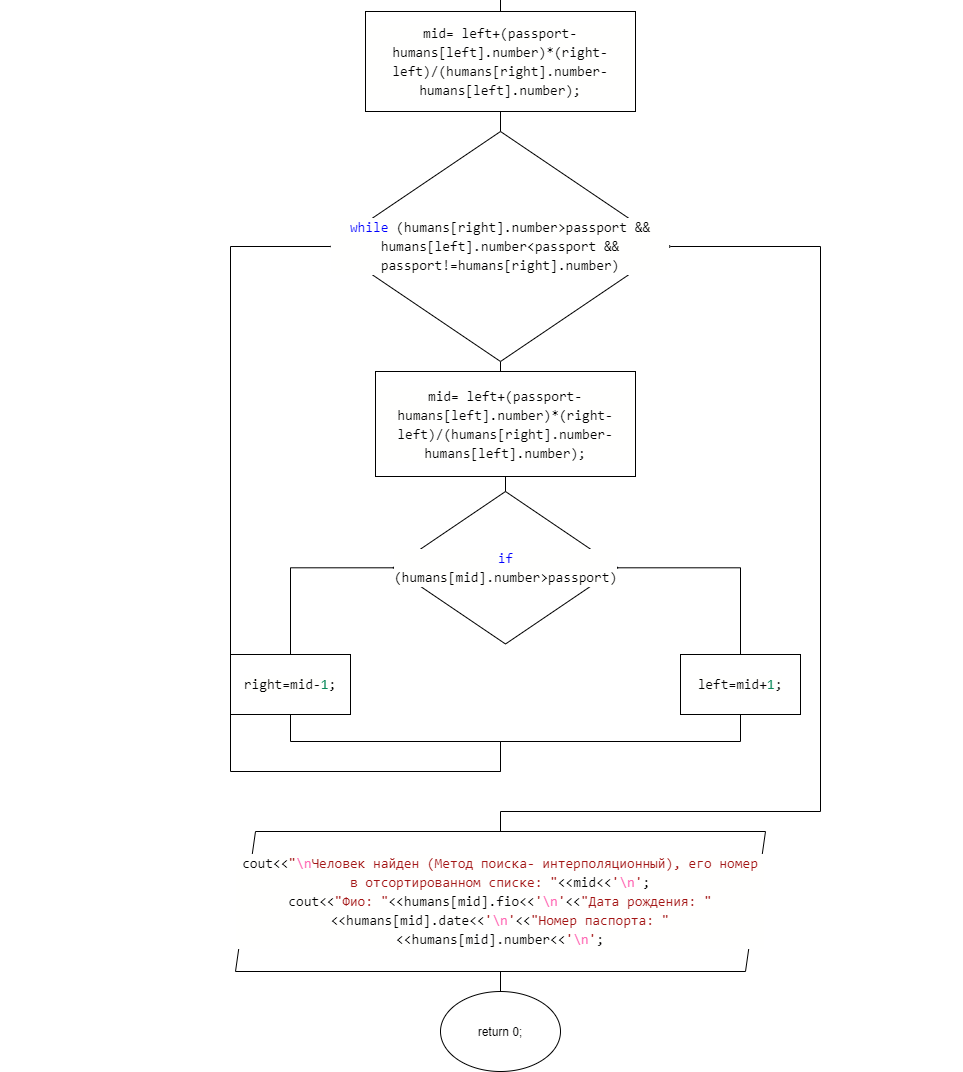
**5.1.** Для успешного применения метода интерполяции, необходимо отсортировать массив по номеру паспорта в порядке возрастания, сортировка проводится методом обмена.

****

**Блок-схема программы**



****

**Решение**

#include <iostream>

#include <string>

#include <fstream>

using namespace std;

struct human

{

string fio;

string date;

int number;

};

int main()

{

// Описание переменных

ifstream input ("F.txt");

string row;

human humans[100], humans2[100], y;

int i=0, j, number=0, passport, left=0, right=4, mid, g, option1=0, option2, length, position;

bool f,fl=false;

//

// Читаем информацию из файла

while (!input.eof())

{

number=0;

getline (input, row);

humans[i].fio=row;

getline (input, row);

humans[i].date=row;

getline (input, row);

number= (row[0]-'0')\*100000+(row[1]-'0')\*10000+(row[2]-'0')\*1000+(row[3]-'0')\*100+(row[4]-'0')\*10+(row[5]-'0');

humans[i].number=number;

i++;

}

input.close();

length=i;

ofstream output ("F.txt");

//

// Выводим массив в консоль

cout<<"Ваш массив:\n";

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

{

cout<<"Фио: "<<humans[i].fio<<'\n'<<"Дата рождения: "<<humans[i].date<<'\n'<<"Номер паспорта: "<<humans[i].number<<'\n'<<'\n';

}

//

while (option1!=5)

{

// Вывод пунктов меню и ввод номера операции

cout<< "Какую операцию вы хотите выполнить?\n";

cout<<"1 - Найти человека по номеру паспорта\n";

cout<<"2 - Добавить информацию о новом человеке\n";

cout<<"3 - Удалить информацию о человеке\n";

cout<<"4 - Занести текущий массив данных в файл\n";

cout<<"5 - Выйти из программы \n";

cout<<"> ";

cin>>option1;

//

// Выполнение операции

switch (option1)

{

case 1:

// Поиск по номеру паспорта

cout<<"Введите номер паспорта нужного вам человека ";

cin>>passport;

//Метод поиска- линейный

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

{

if (humans[i].number==passport)

{

cout<<"Человек найден (Метод поиска- линейный), его номер в списке: "<<i<<'\n';

cout<<"Фио: "<<humans[i].fio<<'\n'<<"Дата рождения: "<<humans[i].date<<'\n'<<"Номер паспорта: "<<humans[i].number<<'\n';

fl=true;

}

}

//

if (fl)

{

//Метод поиска- поиск подстроки в строке

f=false;

g=100000;

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

{

g=100000;

f=false;

while (f!=true and g>0)

{

if (passport/g!=humans[i].number/g)

{

f=true;

}

g=g/10;

}

if (f==false)

{

cout<<"\nЧеловек найден (Метод поиска- поиск подстроки в строке), его номер в списке: "<<i<<'\n';

cout<<"Фио: "<<humans[i].fio<<'\n'<<"Дата рождения: "<<humans[i].date<<'\n'<<"Номер паспорта: "<<humans[i].number<<'\n';

}

}

//

// Метод поиска- интерполяционный

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

{

humans2[i].fio=humans[i].fio;

humans2[i].date=humans[i].date;

humans2[i].number=humans[i].number;

}

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

{

j=i-1;

f=false;

while ((j>-1) and (f==false))

{

if (humans2[j].number>humans2[j+1].number)

{

y=humans2[j];

humans2[j]=humans2[j+1];

humans2[j+1]=y;

}

else

{

f=true;

}

j--;

}

}

mid= left+(passport-humans2[left].number)\*(right-left)/(humans2[right].number-humans2[left].number);

while (humans2[right].number>passport && humans2[left].number<passport && passport!=humans2[right].number)

{

mid= left+(passport-humans2[left].number)\*(right-left)/(humans2[right].number-humans2[left].number);

if (humans2[mid].number>passport)

{

right=mid-1;

}

else

{

left=mid+1;

}

}

cout<<"\nЧеловек найден (Метод поиска- интерполяционный), его номер в отсортированном списке: "<<mid<<'\n';

cout<<"Фио: "<<humans2[mid].fio<<'\n'<<"Дата рождения: "<<humans2[mid].date<<'\n'<<"Номер паспорта: "<<humans2[mid].number<<'\n'<<'\n';

//

}

else

{

cout<<"Человек с этим номером паспорта не найден";

}

break;

//

case 2:

// Добавление элемента

// Вывод пунктов меню и ввод номера операции

cout<<" 1 - Добавление информации в начало списка\n";

cout<<" 2 - Добавление информации в конец списка\n";

cout<<" 3 - Добавление информации на заданную позицию\n";

cout<<" > ";

cin>>option2;

cout<<'\n';

//

// Варианты добавления нового элемента

switch (option2)

{

case 1:

// В начало

for (i=length; i>-1; i--)

{

humans[i].fio=humans[i-1].fio;

humans[i].date=humans[i-1].date;

humans[i].number=humans[i-1].number;

}

cout<<"Введите фио нового человека ";

cin>>row;

humans[0].fio=row;

cout<<"Введите дату рождения нового человека ";

cin>>row;

humans[0].date=row;

cout<<"Введите шестизначный номер паспорта нового человека (не может начинаться на 0) ";

cin>>row;

humans[0].number=(row[0]-'0')\*100000+(row[1]-'0')\*10000+(row[2]-'0')\*1000+(row[3]-'0')\*100+(row[4]-'0')\*10+(row[5]-'0');

length++;

break;

//

case 2:

// В конец

cout<<" Введите фио нового человека ";

cin>>row;

humans[length].fio=row;

cout<<" Введите дату рождения нового человека ";

cin>>row;

humans[length].date=row;

cout<<" Введите шестизначный номер паспорта нового человека (не может начинаться на 0) ";

cin>>row;

humans[length].number=(row[0]-'0')\*100000+(row[1]-'0')\*10000+(row[2]-'0')\*1000+(row[3]-'0')\*100+(row[4]-'0')\*10+(row[5]-'0');

length++;

break;

//

case 3:

// На заданную позицию

cout<<" Введите позицию, на которую вы хотите поставить новый элемент ";

cin>>position;

if (position-1<length+1)

{

for (i=length; i>position-1; i--)

{

humans[i].fio=humans[i-1].fio;

humans[i].date=humans[i-1].date;

humans[i].number=humans[i-1].number;

}

cout<<" Введите фио нового человека ";

cin>>row;

humans[position-1].fio=row;

cout<<" Введите дату рождения нового человека ";

cin>>row;

humans[position-1].date=row;

cout<<" Введите шестизначный номер паспорта нового человека (не может начинаться на 0) ";

cin>>row;

humans[position-1].number=(row[0]-'0')\*100000+(row[1]-'0')\*10000+(row[2]-'0')\*1000+(row[3]-'0')\*100+(row[4]-'0')\*10+(row[5]-'0');

length++;

}

else

{

cout<<"Неверная позиция";

}

break;

//

default:

// Неверный номер

cout<<"Выбран несуществующий вариант";

break;

//

}

//

// Выводим массив в консоль

cout<<"Ваш массив:\n";

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

{

cout<<"Фио: "<<humans[i].fio<<'\n'<<"Дата рождения: "<<humans[i].date<<'\n'<<"Номер паспорта: "<<humans[i].number<<'\n'<<'\n';

}

//

break;

//

case 3:

// Удаление элемента

// Вывод пунктов меню и ввод номера операции

cout<<" 1 - Удаление по фио\n";

cout<<" 2 - Удаление по дате рождения\n";

cout<<" 3 - Удаление по номеру паспорта\n";

cout<<" 4 - Удаление с указанной позиции\n";

cout<<" > ";

cin>>option2;

//

// Варианты удаления элемента

switch (option2)

{

case 1:

// По фио

cout<<"Введите фио человека, информацию о котором вы хотите удалить ";

cin>>row;

fl=false;

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

{

if (humans[i].fio==row)

{

fl=true;

break;

}

}

if (fl)

{

cout<< "Вы собиратесь удалить элемент с индексом "<<i<<'\n';

cout<<"Фио: "<<humans[i].fio<<'\n'<<"Дата рождения: "<<humans[i].date<<'\n'<<"Номер паспорта: "<<humans[i].number<<'\n'<<'\n';

for (;i<length;i++)

{

humans[i].fio=humans[i+1].fio;

humans[i].date=humans[i+1].date;

humans[i].number=humans[i+1].number;

}

humans[i].fio="";

humans[i].date="";

humans[i].number=0;

length--;

}

else

{

cout<<"В списке нет человека с заданным фио\n";

}

break;

//

case 2:

// По дате рождения

cout<<"Введите дату рождения человека, информацию о котором вы хотите удалить ";

cin>>row;

fl=false;

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

{

if (humans[i].date==row)

{

fl=true;

break;

}

}

if (fl)

{

cout<< "Вы собиратесь удалить элемент с индексом "<<i<<'\n';

cout<<"Фио: "<<humans[i].fio<<'\n'<<"Дата рождения: "<<humans[i].date<<'\n'<<"Номер паспорта: "<<humans[i].number<<'\n'<<'\n';

for (;i<length;i++)

{

humans[i].fio=humans[i+1].fio;

humans[i].date=humans[i+1].date;

humans[i].number=humans[i+1].number;

}

humans[i].fio="";

humans[i].date="";

humans[i].number=0;

length--;

}

else

{

cout<<"В списке нет человека с заданной датой рождения\n";

}

break;

//

case 3:

// По номеру паспорта

cout<<"Введите номер паспорта человека, информацию о котором вы хотите удалить ";

cin>>row;

fl=false;

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

{

if (humans[i].number==(row[0]-'0')\*100000+(row[1]-'0')\*10000+(row[2]-'0')\*1000+(row[3]-'0')\*100+(row[4]-'0')\*10+(row[5]-'0'))

{

fl=true;

break;

}

}

if (fl)

{

cout<< "Вы собиратесь удалить элемент с индексом "<<i<<'\n';

cout<<"Фио: "<<humans[i].fio<<'\n'<<"Дата рождения: "<<humans[i].date<<'\n'<<"Номер паспорта: "<<humans[i].number<<'\n'<<'\n';

for (;i<length;i++)

{

humans[i].fio=humans[i+1].fio;

humans[i].date=humans[i+1].date;

humans[i].number=humans[i+1].number;

}

humans[i].fio="";

humans[i].date="";

humans[i].number=0;

length--;

}

else

{

cout<<"В списке нет человека с заданным номером паспорта\n";

}

break;

//

case 4:

// По заданной позиции

cin>>position;

if (position<length)

{

cout<< "Вы собиратесь удалить элемент с индексом "<<i<<'\n';

cout<<"Фио: "<<humans[i].fio<<'\n'<<"Дата рождения: "<<humans[i].date<<'\n'<<"Номер паспорта: "<<humans[i].number<<'\n'<<'\n';

for (i=position;i<length;i++)

{

humans[i].fio=humans[i+1].fio;

humans[i].date=humans[i+1].date;

humans[i].number=humans[i+1].number;

}

humans[i].fio="";

humans[i].date="";

humans[i].number=0;

length--;

}

else

{

cout<<"В списке нет элемента с заданным номером";

}

//

default:

// Неверный номер

cout<<"Выбран несуществующий вариант";

break;

//

}

//

// Выводим массив в консоль

cout<<"Ваш массив:\n";

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

{

cout<<"Фио: "<<humans[i].fio<<'\n'<<"Дата рождения: "<<humans[i].date<<'\n'<<"Номер паспорта: "<<humans[i].number<<'\n'<<'\n';

}

//

//

break;

//

case 4:

// Запись массива в файл

cout<<"Проводим запись в файл...\n";

for (i=0;i<length-1;i++)

{

output<<humans[i].fio;

output<<endl;

output<<humans[i].date;

output<<endl;

output<<humans[i].number;

output<<endl;

}

output<<humans[i].fio;

output<<endl;

output<<humans[i].date;

output<<endl;

output<<humans[i].number;

output.close();

break;

//

case 5:

// Выход из программы

for (i=0;i<length-1;i++)

{

output<<humans[i].fio;

output<<endl;

output<<humans[i].date;

output<<endl;

output<<humans[i].number;

output<<endl;

}

output<<humans[i].fio;

output<<endl;

output<<humans[i].date;

output<<endl;

output<<humans[i].number;

output.close();

break;

//

default:

// Неверный номер

cout<<"Выбран несуществующий вариант";

break;

//

}

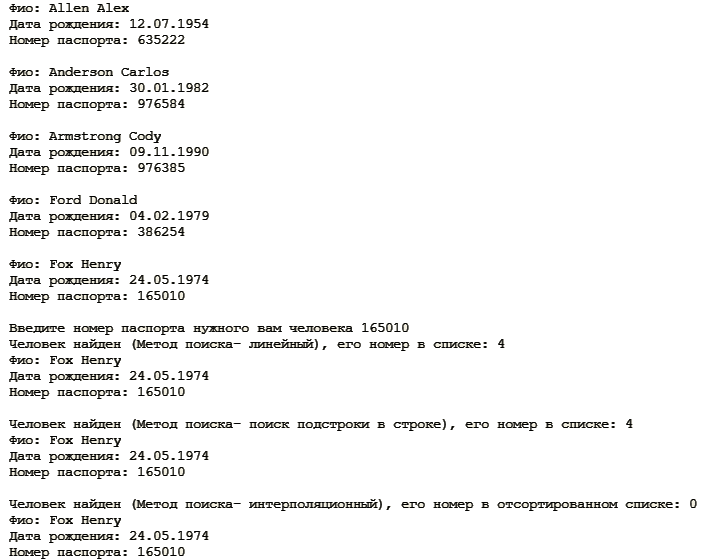
//

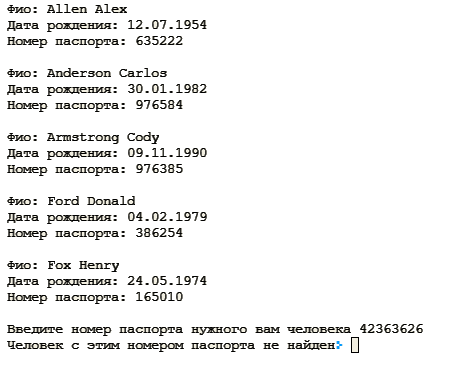
}

return 0;

}

**Скриншоты результатов работы программы**

****

****

**GitHub**

<https://github.com/sugarrrfqs/pnrpuLab12/tree/main>