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 №14**

Дисциплина: «Основы теории алгоритмов и структуры данных»

Тема: Методы поиска: метод Кнутта-Мориса Пратта, метод Бойера-Мур

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.

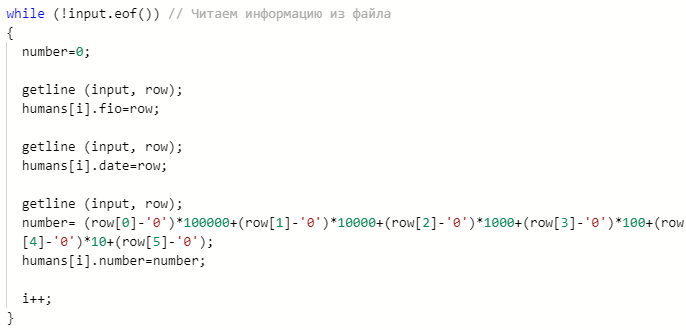
**Постановка задачи**

(Вариант 19) Выполнить поиск элемента в массиве по ключу в соответствии с вариантом. Для поиска использовать метод Кнута-Морриса-Пратта и метод Бойера-Мура.

**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 ФИО человека, информацию о котором необходимо вывести.



**1.3.** Выполнить поиск методом Кнута-Морриса-Пратта и вывести результат в консоль.

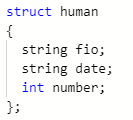


**1.4.** Выполнить поиск методом Бойера-Мура и вывести результат в консоль.



**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.

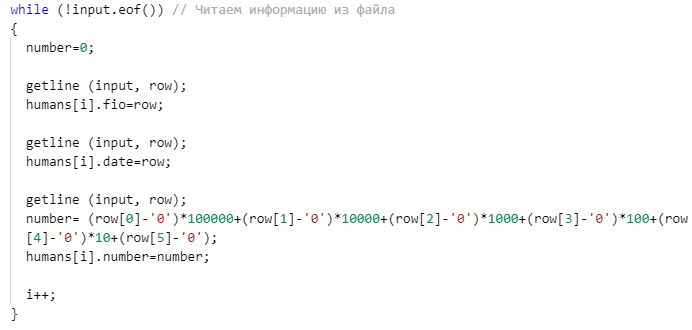


**2.4.** Для переменных-индексов использовался тип int.



**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.

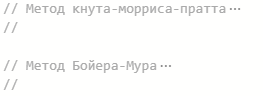


**4.2.** Для ввода информации из файла и с клавиатуры использовалась функция getline.

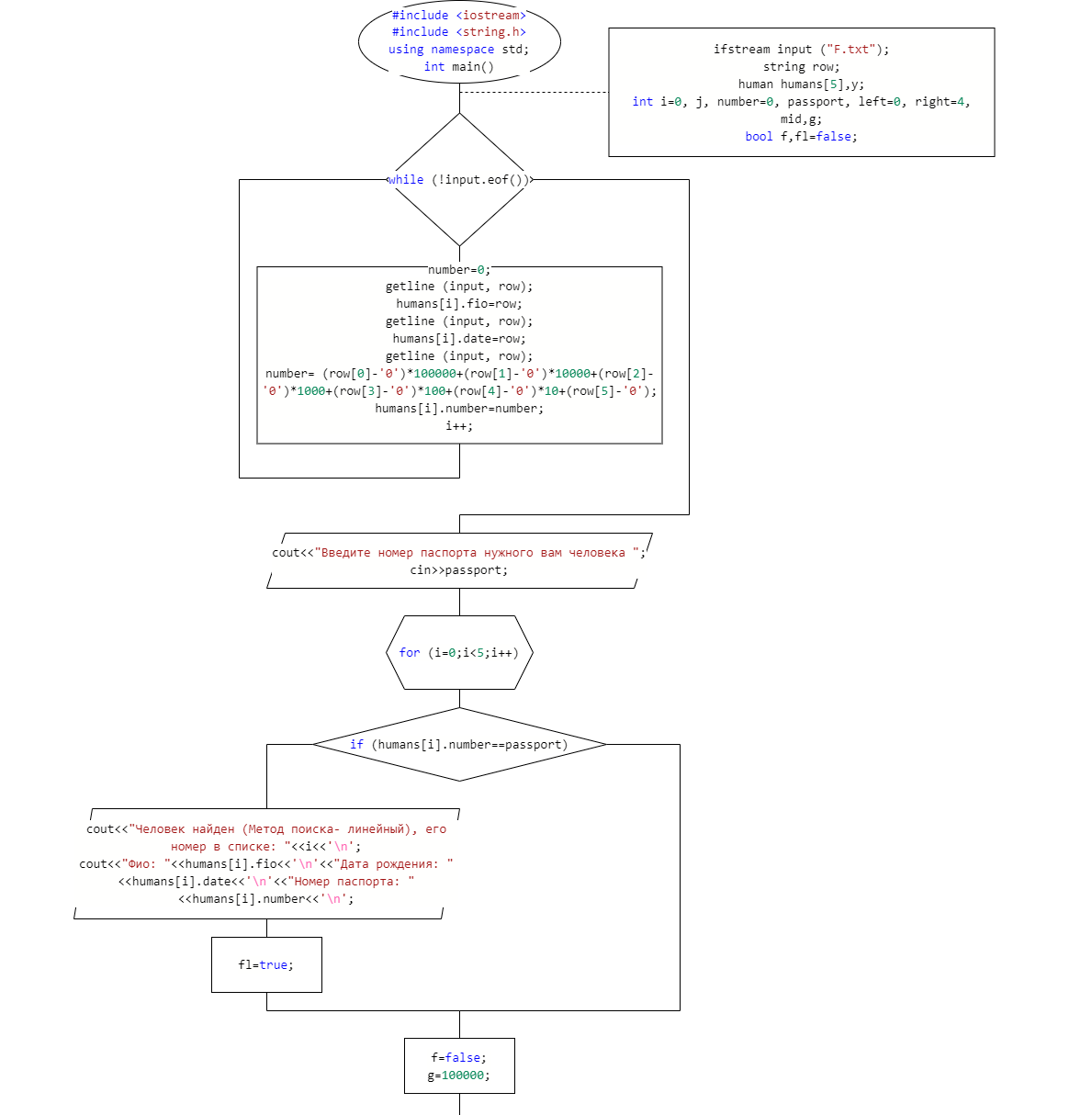


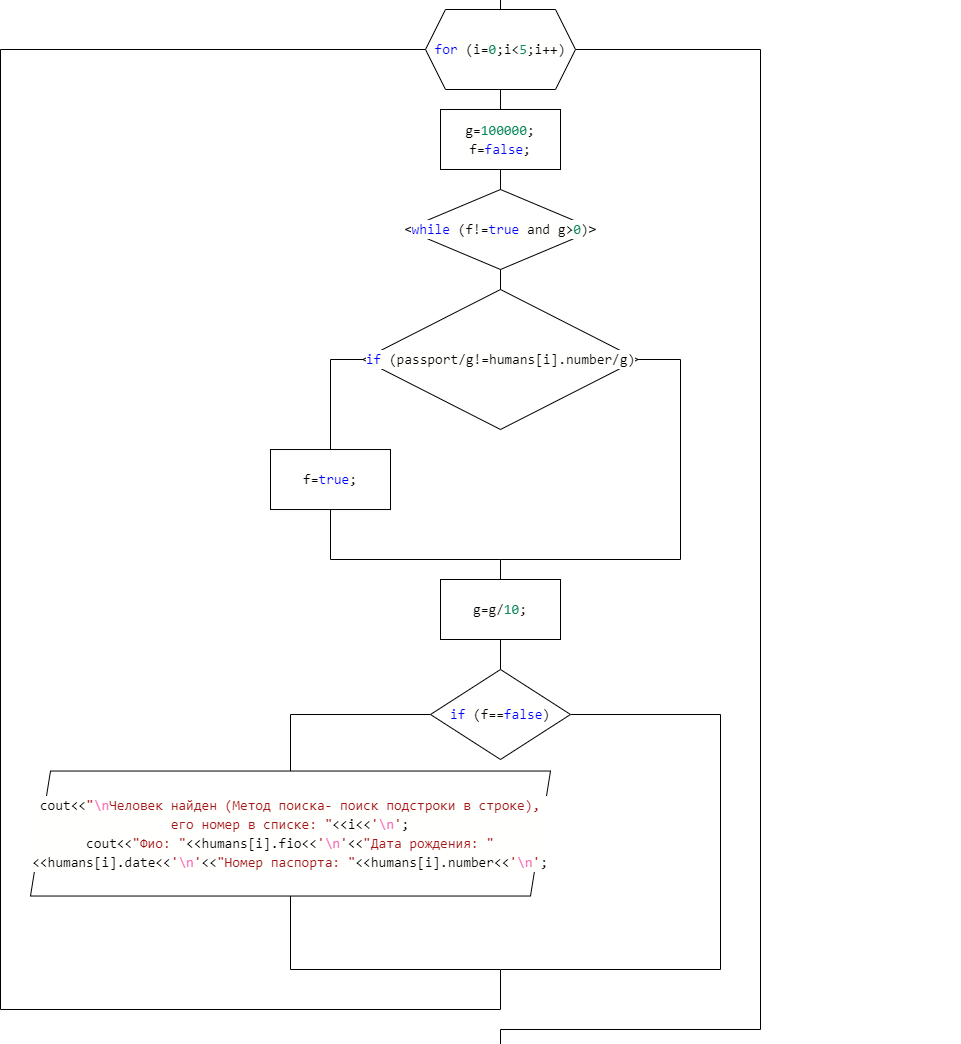
**5.** Поставленные задачи будут решены следующими действиями:

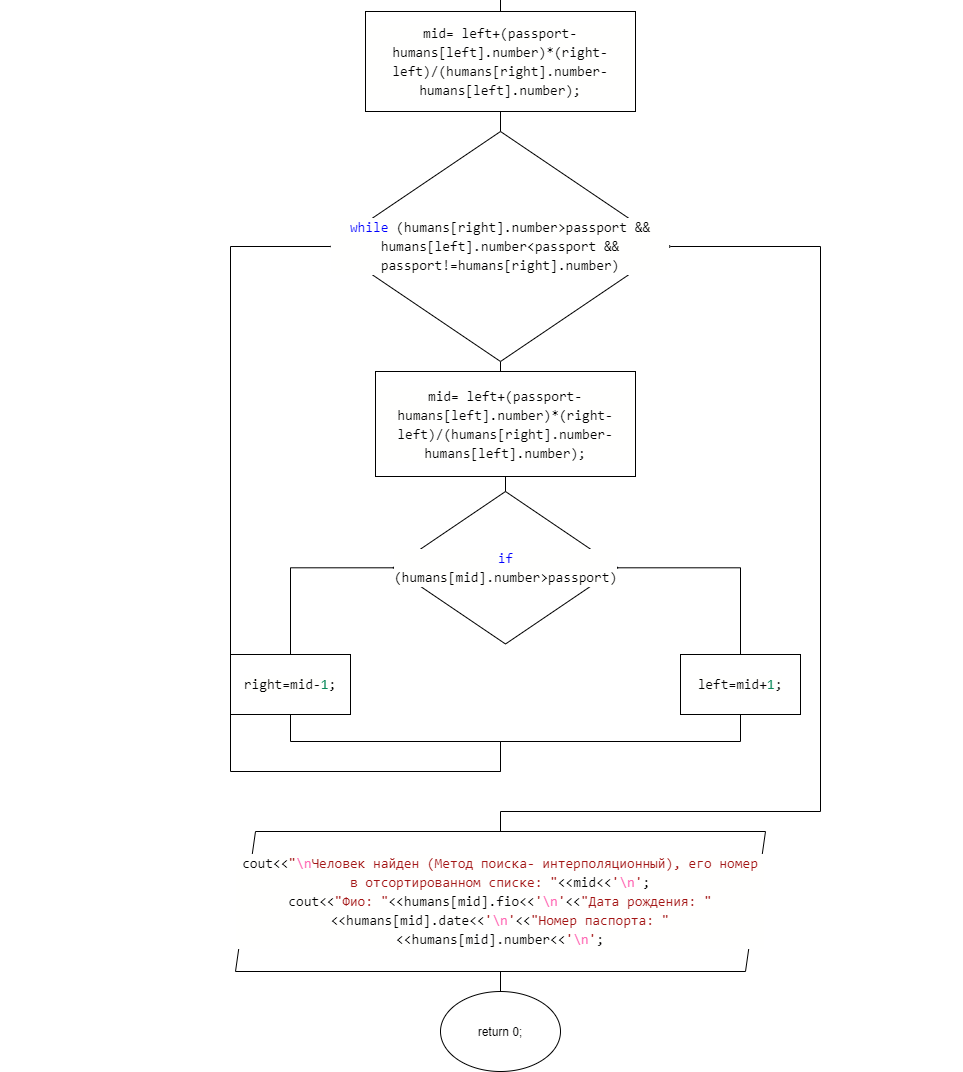
**5.1.** Будут применены коды для 2х методов поиска- метода Кнута-Морриса-Пратта и метода Бойера-Мура.



**Блок-схема программы**



****

****

**Решение**

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

struct human

{

string fio;

string date;

int number;

};

int main()

{

// Описание переменных

ifstream input ("F.txt");

string suff, preff, row, fio, fioForPrefFun;

human humans[100];

int i=0, j, k, length, fioLength, searchFioLength, skip, n, prefFun[100], m, o, moveTable[100], l, z, x;

bool f, fl, fla, flag=false;

//

// Читаем информацию из файла

while (!input.eof())

{

getline (input, row);

humans[i].fio=row;

getline (input, row);

humans[i].date=row;

getline (input, row);

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');

i++;

}

input.close();

length=i;

//

// 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';

}

//

// Метод кнута-морриса-пратта

cout<<"Введите фио человека, которого вы ищете ";

getline (cin, fioForPrefFun);

searchFioLength=fioForPrefFun.length();

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

{

f=false;

fio=fioForPrefFun+'@'+humans[m].fio;

fioLength=fio.length();

cout<<"то самое "<<fioLength<<'\n';

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

{

cout<<"i= "<<i<<'\n';

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

{

preff="";

suff="";

for (k=0; k<=j; k++)

{

preff+=fio[k];

suff+=fio[i-j+k];

}

cout<<"preff= "<<preff<<" suff= "<<suff;

if (preff==suff)

{

skip=j+1;

cout<<" Совпало, skip= "<<skip;

}

cout<<'\n';

}

cout<<'\n';

if (i==0) prefFun[i]=0;

else prefFun[i]=skip;

skip=0;

}

cout<<"Ваша префикс функция ";

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

{

cout<<prefFun[i]<<' ';

}

cout<<'\n';

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

{

if (prefFun[i]==searchFioLength)

{

f=true;

o=m;

}

}

if (f)

{

cout<<"Строка содержит подстроку, индекс человека в массиве- "<<m<<'\n';

fl=true;

}

else

{

cout<<"Строка не содержит подстроку\n";

}

}

cout<<'\n';

if (fl)

{

cout<<"Человек найден (Метод кнута-морриса-пратта), его позиция в списке- "<<o+1<<'\n';

}

else

{

cout<<"Человека с введенным фио нет в списке (Метод кнута-морриса-пратта)";

}

//

// Метод Бойера-Мура

fio=fioForPrefFun+'\*';

fioLength=fio.length();

cout<<fioLength<<'\n';

k=1;

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

{

f=false;

if (fio[i]=='\*' or i==fioLength-2)

{

moveTable[i]=fioLength-1;

f=true;

}

else

{

for (j=fioLength-1; j>i; j--)

{

if (fio[i]==fio[j])

{

f=true;

if (j==fioLength-2 && !flag)

{

moveTable[i]=k;

moveTable[j]=moveTable[i];

flag=true;

}

else

{

moveTable[i]=moveTable[j];

}

}

}

if (!f) moveTable[i]=k;

k++;

}

}

cout<<'\n'<<'\n'<<"Таблица сдвигов:\n";

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

{

cout<<fio[i]<<' ';

}

cout<<'\n';

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

{

cout<<moveTable[i]<<' ';

}

cout<<'\n';

i=0;

fla=true;

while (i<length && fla)

{

m=fioLength-2;

j=fioLength-2;

cout<<"i= "<<i<<'\n';

while (m<=humans[i].fio.length()-1 && fla && j<fioLength-1)

{

f=false;

cout<<m<<' '<<humans[i].fio[m]<<' '<<j<<' '<<fio[j]<<'\n';

if (humans[i].fio[m]!=fio[j])

{

for (k=0; k<fioLength-1; k++)

{

if (humans[i].fio[fioLength]==fio[k] && !f)

{

m+=moveTable[k];

f=true;

j=fioLength-1;

}

}

if (!f)

{

m+=moveTable[fioLength-1];

j=fioLength-1;

}

}

else

{

m--;

j--;

}

if (j==-1)

{

fla=false;

l=i;

}

}

i++;

}

if (!fla)

{

cout<<"Человек найден (Метод Бойера- Мура), его позиция - "<<l<<'\n';

}

else

{

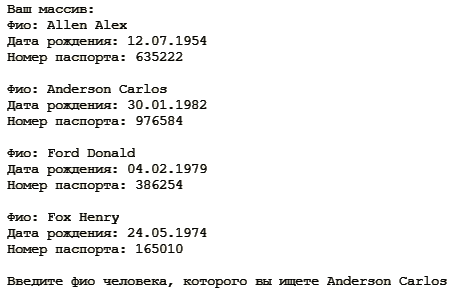
cout<<"Человека с введенным фио нет в списке (Метод Бойера- Мура)";

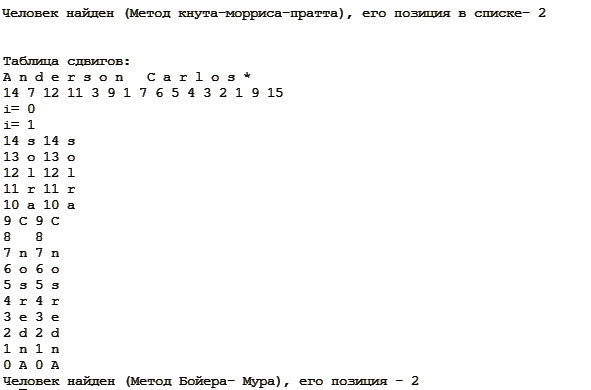
}

//

}

**Скриншоты результатов работы программы**

****

****

**GitHub**

<https://github.com/sugarrrfqs/Lab14/tree/main>