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 №17**

Дисциплина: «Основы теории алгоритмов и структуры данных»

Тема: Методы быстрого поиска. Хеширование

Вариант 5

Выполнил работу

студент группы РИС-20-1б

Зверев А.Д.

Проверила

Доцент кафедры ИТАС

Полякова О.А.

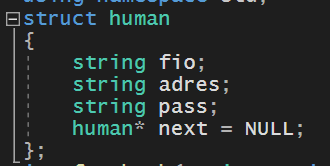
Пермь, 2021

**Постановка задачи**

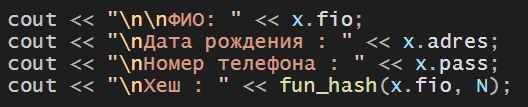
В качестве данных используются ФИО, номер паспорта, и адрес человека. С помощью ключа поиска, которым в данном варианте является ФИО, и с помощью хэш—функции H(k)=k mod M, а также в качестве метода рехеширования используя метод цепочек, написать программу, которая будет выполнять поиск по заданным изначально данным (данные находятся в текстовых документах).

**Анализ задачи**

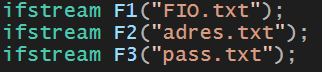
1. Для решения задачи необходимо…
   1. Создать структуру human для хранения данных различных типов в программе для удобства обращения к ним;



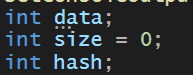
* 1. В цикле for, который будет читать данные и перебирать их в хэш—таблице сделать отдельным пунктом вывод данных;



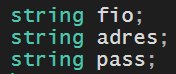
* 1. С помощью переменных ifstream открываем текстовые файлы с данными о людях, данные изначально;



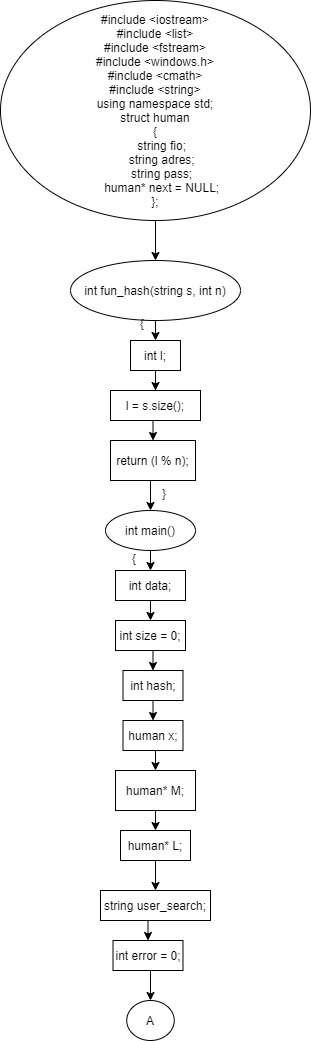
1. В программе были использованы следующие типы данных:
   1. Тип данных int для хранения различных промежуточных данных в программе;

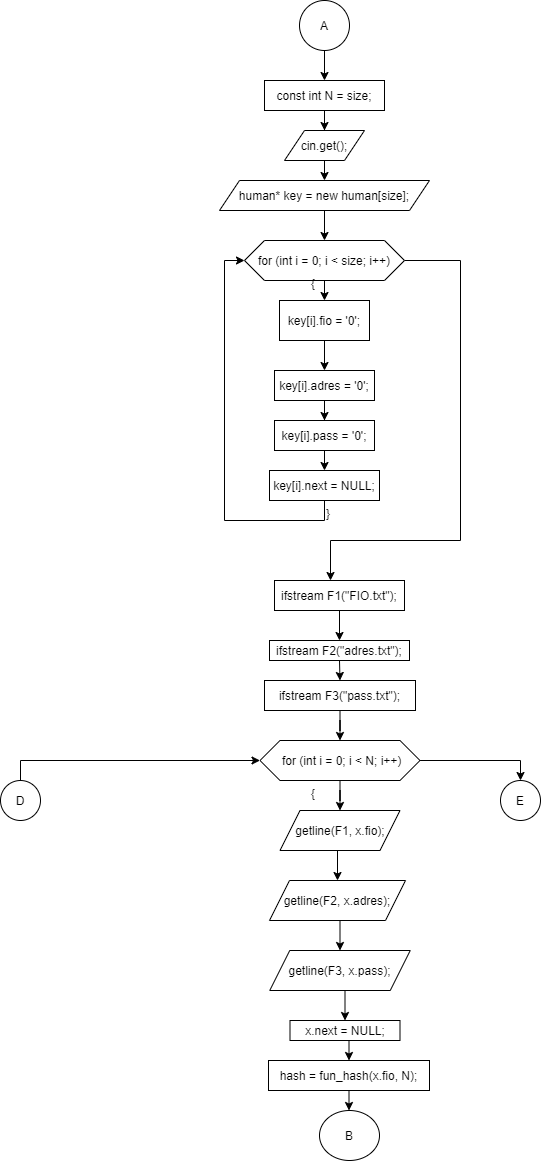


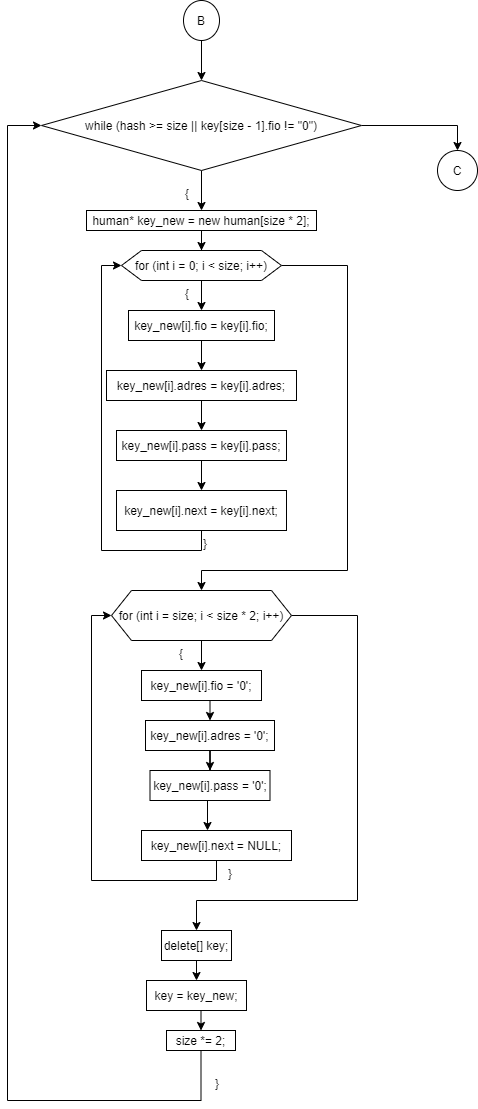
* 1. Тип данных string для хранения различной информации о человеке, например, его ФИО;

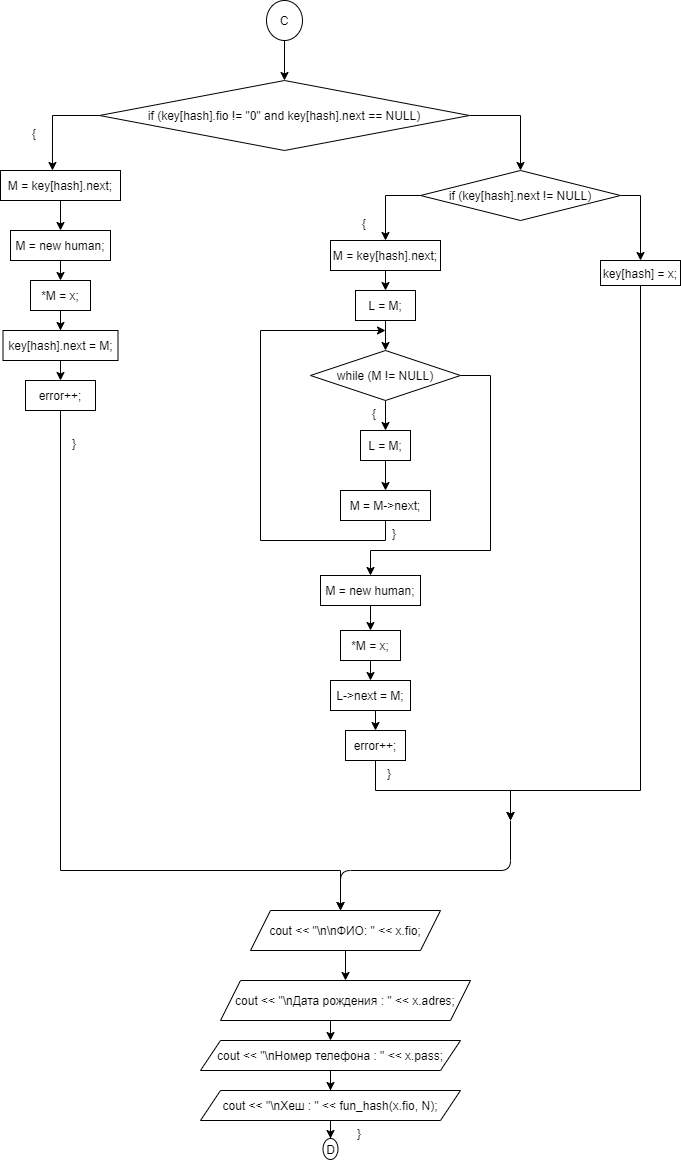


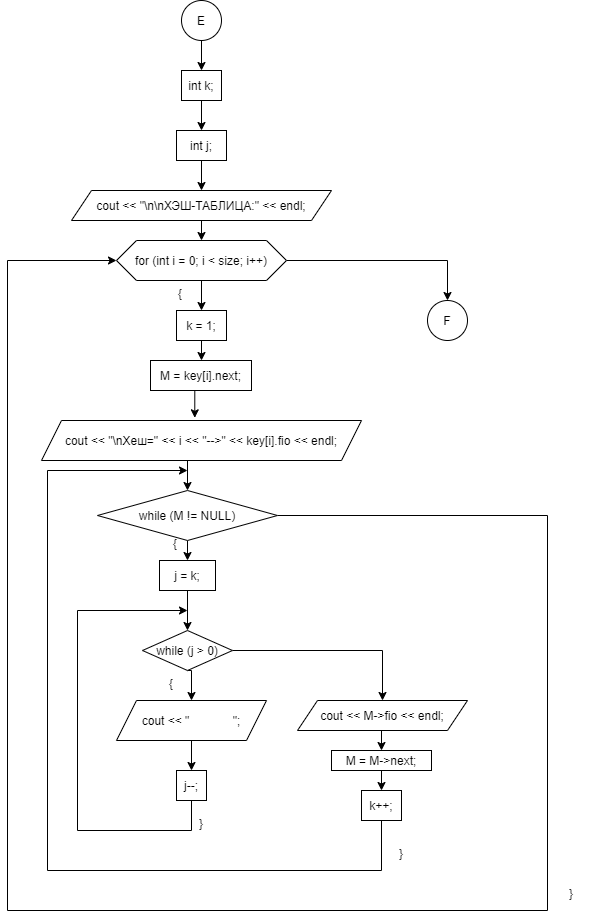
**Блок—схема программы**

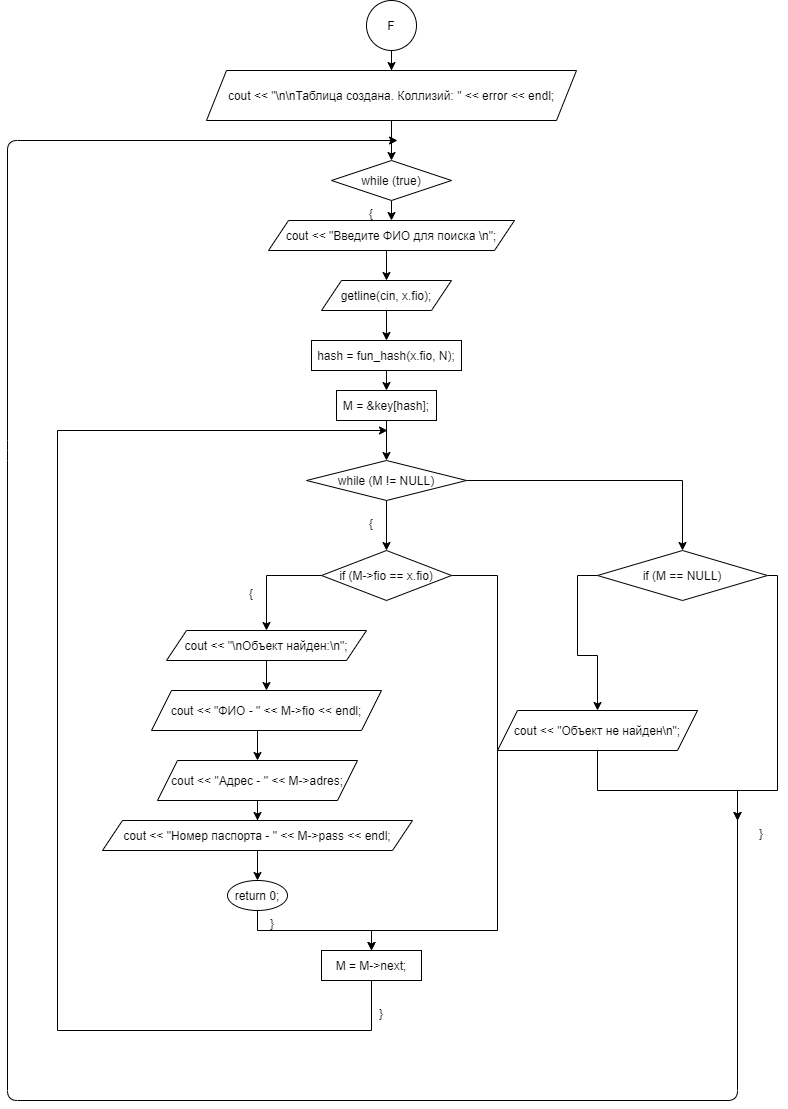












**Решение**

#include <iostream>

#include <list>

#include <fstream>

#include <windows.h>

#include <cmath>

#include <string>

using namespace std;

struct human

{

string fio;

string adres;

string pass;

human\* next = NULL;

};

int fun\_hash(string s, int n)

{

int l;

l = s.size();

return (l % n);

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int data;

int size = 0;

int hash;

human x;

human\* M;

human\* L;

string user\_search;

int error = 0;

while (size < 2 || size>100)

{

cout << "Введите размер таблицы\n";

cin >> size;

}

const int N = size;

cin.get();

human\* key = new human[size];

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

{

key[i].fio = '0';

key[i].adres = '0';

key[i].pass = '0';

key[i].next = NULL;

}

ifstream F1("FIO.txt");

ifstream F2("adres.txt");

ifstream F3("pass.txt");

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

{

getline(F1, x.fio);

getline(F2, x.adres);

getline(F3, x.pass);

x.next = NULL;

hash = fun\_hash(x.fio, N);

while (hash >= size || key[size - 1].fio != "0")

{

human\* key\_new = new human[size \* 2];

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

{

key\_new[i].fio = key[i].fio;

key\_new[i].adres = key[i].adres;

key\_new[i].pass = key[i].pass;

key\_new[i].next = key[i].next;

}

for (int i = size; i < size \* 2; i++)

{

key\_new[i].fio = '0';

key\_new[i].adres = '0';

key\_new[i].pass = '0';

key\_new[i].next = NULL;

}

delete[] key;

key = key\_new;

size \*= 2;

}

if (key[hash].fio != "0" and key[hash].next == NULL)

{

M = key[hash].next;

M = new human;

\*M = x;

key[hash].next = M;

error++;

}

else

if (key[hash].next != NULL)

{

M = key[hash].next;

L = M;

while (M != NULL)

{

L = M;

M = M->next;

}

M = new human;

\*M = x;

L->next = M;

error++;

}

else

{

key[hash] = x;

}

cout << "\n\nФИО: " << x.fio;

cout << "\nДата рождения : " << x.adres;

cout << "\nНомер телефона : " << x.pass;

cout << "\nХеш : " << fun\_hash(x.fio, N);

}

int k;

int j;

cout << "\n\nХЭШ-ТАБЛИЦА:" << endl;

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

{

k = 1;

M = key[i].next;

cout << "\nХеш=" << i << "-->" << key[i].fio << endl;

while (M != NULL)

{

j = k;

while (j > 0) { cout << " "; j--; } cout << M->fio << endl;

M = M->next;

k++;

}

}

cout << "\n\nТаблица создана. Коллизий: " << error << endl;

while (true)

{

cout << "Введите ФИО для поиска \n";

getline(cin, x.fio);

hash = fun\_hash(x.fio, N);

M = &key[hash];

while (M != NULL)

{

if (M->fio == x.fio)

{

cout << "\nОбъект найден:\n";

cout << "ФИО - " << M->fio << endl;

cout << "Адрес - " << M->adres;

cout << "Номер паспорта - " << M->pass << endl;

return 0;

}

M = M->next;

}

if (M == NULL) cout << "Объект не найден\n";

}

}

**Скриншоты**

