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 14

Выполнил:

студент группы РИС-20-2б

Вичугов Алексей Дмитриевич

Проверила:

доцент кафедры ИТАС

Полякова О. А.

Пермь, 2021

**Цель работы**

1. Изучить построение функции хеширования и алгоритмов хеширования данных.

2. Научиться разрабатывать алгоритмы открытого и закрытого хеширования при решении задач на языке С++.

**Постановка задачи**

1. Создать динамический массив из записей (в соответствии с вариантом), содержащий не менее 100 элементов. Для заполнения элементов массива использовать ДСЧ.

2. Выполнить поиск элемента в массиве по ключу в соответствии с вариантом. Для поиска использовать хеш-таблицу.

3. Подсчитать количество коллизий при размере хеш-таблицы 40, 75 и 90 элементов.

**Анализ задачи**

1. Для решения задачи необходимо:

1.1. Сформировать массив структурных элементов:

struct Dan{

string fio;

string data;

string adres;

};

Dan \*m;

void Create(){

cout << "Размерность массива: ";

cin >> r;

while (r<2||cin.fail()){

cin.ignore(30,'\n');

cin.clear();

cout << "Размерность массива должна иметь значение не менее 2! Повторите ввод: ";

cin >> r;

}

m = new Dan[r];

h = new string[r\*2];

Zapolnenie();

HTab();

}

1.2. Создать хеш-функцию:

int Hesher(string s){

int ind;

float keysa=0;

for (int i=0;i<s.length();i++) keysa+=s[i];

if (keysa<0) keysa=-keysa;

int ky=keysa\*0.72;

keysa\*=0.72;

keysa-=ky;

ind = keysa \* r;

return ind;

}

1.3. Сформировать хеш-таблицу:  
void HTab(){

int in;

int col=0;

int raz=r;

for (int i=0;i<r;i++){

in=Hesher(m[i].adres);

if (h[in].length()>0){

col++;

cout << "Обнаружена коллизия на " << in << "-м элементе. " << endl;

while (in<raz){

if (h[in].length()>0)in++;

else break;

}

}

if (in>=raz){

raz\*=2;

string \*nh= new string [raz];

for (int e=0;e<r;e++) nh[e]=h[e];

delete[] h;

string \*h= new string [raz];

for (int e=0;e<r;e++) h[e]=nh[e];

delete[] nh;

}

h[in]=m[i].adres;

cout << "h[" << in << "] " << h[in] << endl; //!\_!

}

cout << "Хеш-таблица заполнена. " << endl << "Обнаружено " << col << " коллизий. " << endl;

}

2. В ходе работы были использованы следующие типы данных:

2.1. Для хранения значений элементов массива и хранения нужных индексов элементов используется тип int:

int r;

2.2. Для формирования результата хеш-функции используется тип float:  
float keysa=0;

3. Для хранения данных элементов массива используется структура:

struct Dan{

string fio;

string data;

string adres;

};

4. Для операций ввода и вывода использовались следующие операторы:

4.1. Для ввода данных использовался оператор cin:

cin >> r;

4.2. Для вывода результата работы программы на экран используется оператор cout:

cout << "Элемент с введённым адресом найден. " << endl;

5. Для решения задач использовались следующие циклы:

5.1. Цикл while для проверки правильности ввода:

while (cin.fail()||r<2){

cout << "Размерность массива должна иметь значение не менее 2! Повторите ввод: ";

cin.ignore(30,'\n');

cin.clear();

cin >> r;

}

5.2.Для работы хеш-функции используется цикл for:

for (int i=0;i<s.length();i++) keysa+=s[i];

**Полный код программы**

#include <iostream>

#include <string>

#include <cstdlib>

#include <ctime>

using namespace std;

struct Dan{

string fio;

string data;

string adres;

};

Dan \*m;

string \*h;

int r;

int Hesher(string);

void HTab(){

int in;

int col=0;

int raz=r;

for (int i=0;i<r;i++){

in=Hesher(m[i].adres);

if (h[in].length()>0){

col++;

cout << "Обнаружена коллизия на " << in << "-м элементе. " << endl;

while (in<raz){

if (h[in].length()>0)in++;

else break;

}

}

if (in>=raz){

raz\*=2;

string \*nh= new string [raz];

for (int e=0;e<r;e++) nh[e]=h[e];

delete[] h;

string \*h= new string [raz];

for (int e=0;e<r;e++) h[e]=nh[e];

delete[] nh;

}

h[in]=m[i].adres;

cout << "h[" << in << "] " << h[in] << endl; //!\_!

}

cout << "Хеш-таблица заполнена. " << endl << "Обнаружено " << col << " коллизий. " << endl;

}

void Zapolnenie();

void Create(){

cout << "Размерность массива: ";

cin >> r;

while (r<2||cin.fail()){

cin.ignore(30,'\n');

cin.clear();

cout << "Размерность массива должна иметь значение не менее 2! Повторите ввод: ";

cin >> r;

}

m = new Dan[r];

h = new string[r\*2];

Zapolnenie();

HTab();

}

void Zapolnenie(){

int ran,n;

srand(time(0));

for (int i=0;i<r;i++){

n=rand()%10;

ran=rand()%10;

switch (n){

case 0: m[i].fio="Анастасия "; break;

case 1: m[i].fio="Мария "; break;

case 2: m[i].fio="Дарья "; break;

case 3: m[i].fio="Анна "; break;

case 4: m[i].fio="Елизавета "; break;

case 5: m[i].fio="Александр "; break;

case 6: m[i].fio="Максим "; break;

case 7: m[i].fio="Иван "; break;

case 8: m[i].fio="Артём "; break;

case 9: m[i].fio="Дмиртий "; break;

}

switch (ran){

case 0: if (n<5) m[i].fio+="Владимировна "; else m[i].fio+="Владимирович "; break;

case 1: if (n<5) m[i].fio+="Ивановна "; else m[i].fio+="Иванович "; break;

case 2: if (n<5) m[i].fio+="Александровна "; else m[i].fio+="Александрович "; break;

case 3: if (n<5) m[i].fio+="Сергеевна "; else m[i].fio+="Сергеевич "; break;

case 4: if (n<5) m[i].fio+="Антоновна "; else m[i].fio+="Антонович "; break;

case 5: if (n<5) m[i].fio+="Петровна "; else m[i].fio+="Петрович "; break;

case 6: if (n<5) m[i].fio+="Дмитриевна "; else m[i].fio+="Дмитриевич "; break;

case 7: if (n<5) m[i].fio+="Анатольевна "; else m[i].fio+="Анатольевич "; break;

case 8: if (n<5) m[i].fio+="Артёмовна "; else m[i].fio+="Артёмович "; break;

case 9: if (n<5) m[i].fio+="Семёновна "; else m[i].fio+="Семёнович "; break;

}

ran=rand()%10;

switch (ran){

case 0: if (n<5) m[i].fio+="Иванова"; else m[i].fio+="Иванов"; break;

case 1: if (n<5) m[i].fio+="Петрова"; else m[i].fio+="Петров"; break;

case 2: if (n<5) m[i].fio+="Сидорова"; else m[i].fio+="Сидоров"; break;

case 3: if (n<5) m[i].fio+="Кузнецова"; else m[i].fio+="Кузнецов"; break;

case 4: if (n<5) m[i].fio+="Попова"; else m[i].fio+="Попов"; break;

case 5: if (n<5) m[i].fio+="Смирнова"; else m[i].fio+="Смирнов"; break;

case 6: if (n<5) m[i].fio+="Васильева"; else m[i].fio+="Васильев"; break;

case 7: if (n<5) m[i].fio+="Соколова"; else m[i].fio+="Соколов"; break;

case 8: if (n<5) m[i].fio+="Новикова"; else m[i].fio+="Новиков"; break;

case 9: if (n<5) m[i].fio+="Волкова"; else m[i].fio+="Волков"; break;

}

ran=rand()%10;

switch(ran){

case 0: m[i].data+="12.05.1930"; break;

case 1: m[i].data+="03.07.1928"; break;

case 2: m[i].data+="17.12.1992"; break;

case 3: m[i].data+="13.08.2000"; break;

case 4: m[i].data+="14.06.1974"; break;

case 5: m[i].data+="01.01.2001"; break;

case 6: m[i].data+="28.02.2020"; break;

case 7: m[i].data+="31.03.2007"; break;

case 8: m[i].data+="16.01.1960"; break;

case 9: m[i].data+="15.05.1943"; break;

}

ran=rand()%10;

switch (ran){

case 0: m[i].adres="Пермь, "; break;

case 1: m[i].adres="Москва, "; break;

case 2: m[i].adres="Санкт-Петербург, "; break;

case 3: m[i].adres="Екатеринбург, "; break;

case 4: m[i].adres="Новосибирск, "; break;

case 5: m[i].adres="Новокузнецк, "; break;

case 6: m[i].adres="Новороссийск, "; break;

case 7: m[i].adres="Омск, "; break;

case 8: m[i].adres="Архангельск, "; break;

case 9: m[i].adres="Курган, "; break;

}

ran=rand()%10;

switch(ran){

case 0: m[i].adres+="ул. Екатерининская, "; break;

case 1: m[i].adres+="ул. Мостовая, "; break;

case 2: m[i].adres+="ул. Центральная, "; break;

case 3: m[i].adres+="ул. Королёва, "; break;

case 4: m[i].adres+="ул. Мира, "; break;

case 5: m[i].adres+="ул. Луговая, "; break;

case 6: m[i].adres+="ул. Лаврова, "; break;

case 7: m[i].adres+="ул. Институтская, "; break;

case 8: m[i].adres+="ул. Арбат, "; break;

case 9: m[i].adres+="ул. Революции, "; break;

}

ran=rand()%10;

switch(ran){

case 0: m[i].adres+="д. 17"; break;

case 1: m[i].adres+="д. 14"; break;

case 2: m[i].adres+="д. 46"; break;

case 3: m[i].adres+="д. 8"; break;

case 4: m[i].adres+="д. 5-А"; break;

case 5: m[i].adres+="д. 91"; break;

case 6: m[i].adres+="д. 194"; break;

case 7: m[i].adres+="д. 18"; break;

case 8: m[i].adres+="д. 7"; break;

case 9: m[i].adres+="д. 81"; break;

}

cout << i+1 << ". " << m[i].fio << "; " << m[i].data << "; " << m[i].adres << "; " << endl;

}

}

int Hesher(string s){

int ind;

float keysa=0;

for (int i=0;i<s.length();i++) keysa+=s[i];

if (keysa<0) keysa=-keysa;

int ky=keysa\*0.72;

keysa\*=0.72;

keysa-=ky;

ind = keysa \* r;

return ind;

}

void Poisk(){

bool p,b=0;

string s;

int i;

cout << "Адрес искомого элемента: ";

cin.ignore(30,'\n');

getline (cin, s, '\n');

i=Hesher(s);

cout << "index " << i << endl;//!\_!

if (h[i]==s) {

cout << "Элемент с введённым адресом найден. " << endl;

b=1;

}

else while (i<r&&!b){

i++;

if (h[i]==s){

cout << "Элемент с введённым адресом найден. " << endl;

b=1;

}

}

if (!b) cout << "Элемент не найден. " << endl;

cout << "Продолжить поиск? (1/0). " << endl;

cin >> p;

if (p) Poisk();

}

void Set(){

// setlocale(LC\_ALL, "");

const int fon = system("Color F0");

system("chcp 1251>nul");

}

int main(){

Set();

Create();

Poisk();

}

**Результат работы программы**