Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Кафедра **«**Информационные технологии и автоматизированные системы**»**

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ**

Дисциплина: «Основы алгоритмизации и программирования»

Семестр 2

Тема: Хэш-таблицы

Выполнил работу

Студент группы РИС-22-1Б

Лихачев Д.А.

Проверил

Доцент кафедры ИТАС

Полякова О.А.

г. Пермь-2023

**Введение**

Для выполнения лабораторной работы требуется сделать отчет программы. Создание программ – отличный способ практики программирования.

**Постановка задачи**

Требуется реализовать алгоритмы нахождения элементов по ключу используя хэш-таблицы



**Код программы**

**hash tab.cpp**

#include <iostream>

#include <string>

#include <fstream>

#include <vector>

using namespace std;

int random(int max)

{

return (rand() % (max + 1));

}

int hash\_func(string tel\_num, int tab\_size);

string NAMES[10] = { "Андрей", "Антон", "Иван", "Сергей", "Владимир", "Николай", "Егор", "Дмитрий", "Данил", "Александр" };

string SURNAMES[10] = { "Соколов", "Кузнецов", "Попов", "Беляев", "Орлов", "Баранов", "Давыдов", "Павлов", "Романов", "Морозов" };

string PATRONYMIC[10] = { "Александрович", "Кирилович", "Олегович", "Иванович", "Алексеевич", "Михайлович", "Константинович", "Григорьевич", "Романович", "Николаевич" };

string DATES[10] = { "27.01.1988","28.02.1967","29.04.1999","30.05.1867","31.12.1999","01.06.1976","02.07.1980","03.08.1982","04.09.1955","05.10.1948" };

struct Human

{

string name;

string date;

string tel\_num;

Human()

{

name = "NULL";

date = "NULL";

tel\_num = "NULL";

}

void print()

{

cout << name << " " << date << " " << tel\_num;

}

};

struct hash\_tab

{

Human\* humans;

int t\_size;

int colls;

hash\_tab(int size)

{

humans = new Human[size];

t\_size = size;

colls = 0;

}

~hash\_tab()

{

delete[]humans;

}

void print()

{

for (int i = 0; i < t\_size; i++)

{

humans[i].print();

cout << endl;

}

}

void add(Human human, int size)

{

int idx = hash\_func(human.tel\_num, size);

int hash = idx;

if (humans[idx].name == "NULL")

{

humans[idx] = human;

return;

}

else

{

while (idx < size)

{

if (humans[idx].name == "NULL")

{

humans[idx] = human;

return;

}

idx++;

colls++;

}

if (idx >= size)

{

idx = 0;

colls++;

while (idx < hash)

{

if (humans[idx].name == "NULL")

{

humans[idx] = human;

return;

}

idx++;

colls++;

}

}

}

}

void find(string tel, int size)

{

int hash = hash\_func(tel, size);

int idx = hash;

while (humans[idx].tel\_num != tel && idx < size)

{

idx++;

}

if (idx >= size)

{

idx = 0;

while (humans[idx].tel\_num != tel && idx < hash)

{

idx++;

}

if (idx >= hash)

{

cout << "Не найден" << endl;

}

else

{

cout << "Index: " << idx << endl;

}

}

else

{

cout << "Index: " << idx << endl;

}

}

};

int hash\_func(string tel\_num, int tab\_size)

{

string tel\_s;

for (int i = 2; i < 10; i++)

{

tel\_s += tel\_num[i];

}

//cout << "tel\_s = " << tel\_s;

int tel = stoi(tel\_s);

//cout << " tel = " << tel;

double ka = tel \* 0.467;

//cout << " ka = " << ka;

double t;

double kamod1 = modf(ka, &t);

//cout << " kamod1 = " << kamod1 << " -> ";

double H = tab\_size \* kamod1;;

return int(H);

}

Human create\_human()

{

Human temp;

temp.name = SURNAMES[random(9)] + " " + NAMES[random(9)] + " " + PATRONYMIC[random(9)];

temp.date = DATES[random(9)];

string temp\_telnum = "89";

for (int i = 0; i < 10; i++)

{

temp\_telnum += to\_string(random(9));

}

temp.tel\_num = temp\_telnum;

return temp;

}

void fill\_human\_arr(Human\* arr, int size)

{

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

{

arr[i] = create\_human();

}

}

void print\_human\_arr(Human\* arr, int size)

{

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

{

arr[i].print();

cout << endl;

}

}

int main()

{

setlocale(LC\_ALL, "RUS");

srand(time(NULL));

int size\_m;

cout << "Размер массива: ";

cin >> size\_m;

cout << endl;

Human\* humans = new Human[size\_m];

fill\_human\_arr(humans, size\_m);

cout << "Исходный массив:\n\n";

print\_human\_arr(humans, size\_m);

cout << "\nХеш таблица:\n\n";

hash\_tab HASH\_TABLE(size\_m);

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

{

HASH\_TABLE.add(humans[i], size\_m);

}

HASH\_TABLE.print();

cout << "\nПоиск по номеру: ";

string str\_f;

cin >> str\_f;

cout << endl;

HASH\_TABLE.find(str\_f, size\_m);

cout << "\nКоллизий: " << HASH\_TABLE.colls;

cout << "\n\n\n";

}

**Вывод программы**



***Рисунок 1 – вывод программы***

**Вывод**

Программа выполняет свою задачу.

**Заключение**

Для решения задачи потребовались знания языка программирования, а конкретнее C++. Программа выполняет те условия, что были указаны в постановке задачи и работает без проблем.