# Учреждение образования

# «Белорусский государственный технологический университет»

Кафедра информационных систем и технологий

Отчёт по лабораторной работе №15

Вариант 2

Выполнил:

студент 1 курса 2 группы Самцевич Алексей

Минск 2020

Задание №1: **Список электронных адресов.** Создать хеш-таблицу со следующими полями: адрес, фамилия абонента, год создания. Ключ – год создания

Код:

Главный файл:

//

// main.cpp

// laba\_15.1

//

// Created by Alex Samtsevich on 08.06.2020.

// Copyright © 2020 Alex Samtsevich. All rights reserved.

//

#include <iostream>

#include<string>

#include<ctime>

using namespace std;

struct Obj {

string surname = "";

string adress = "";

int year = 0;

int getKey() {

return year;

}

};

struct node {// узел списка

Obj\* inf;

node\* next = nullptr;

};

void add(node\*& root, Obj\* obj) {

if (root == nullptr) {

root = new node;

root->inf = obj;

}

else {

node\* t = root;

while (t->next != nullptr)

t = t->next;

t->next = new node;

t->next->inf = obj;

}

}

bool search(node\* root, Obj\* obj) {

node\* t = root;

while (t->next != nullptr) {

if (t->inf->adress == obj->adress && t->inf->surname == obj->surname && t->inf->year == obj->year)

return true;

t = t->next;

}

return false;

}

void print(node\* root) {

for (node\* t = root; t != nullptr; t = t->next)

cout << "(" << t->inf->surname << " " << t->inf->adress << " " << t->inf->year << "), ";

}

struct hashmas {

int cur\_size = 0; // текущий размер

int size;

node\*\* mas; // массив указателей на начало спискof

hashmas(int s) {

size = s;

mas = new node \* [size];

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

mas[i] = nullptr;

}

~hashmas() {// конструктор/ деструктур

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

delete mas[i];

delete[]mas;

}

};

int Hash(int key, int size, int p) {

int a = 7;

int b = 3571;

return ((a \* key + p) % b) % size;

}

bool add(hashmas& mas, Obj\* obj, int& a) {// добавление в хэштаблицу

int p = 0;

int key1 = Hash(obj->getKey(), mas.size, p++);

add(mas.mas[key1], obj);

return true;

}

int search(hashmas& mas, Obj\* obj) {

int p = 0;

int key1 = Hash(obj->getKey(), mas.size, p++);

search(mas.mas[key1], obj);

return key1;

}

void print(hashmas& mas) {

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

if (mas.mas[i] != nullptr) {

cout << i << " ";

print(mas.mas[i]);

cout << endl;

}

}

int main() {

int n = 10;

int col = 0;

hashmas mas1(n);

for (int i = 0; i < 20; i++) {

string str = "Alex" + to\_string(i);

Obj\* obj = new Obj;

obj->surname = str;

str += "@gmail.com";

obj->adress = str;

obj->year = rand() % 70 + 1950;

add(mas1, obj, col);

}

string str = "Tramp";

Obj\* obj = new Obj;

obj->surname = str;

str += ".official@gmail.com";

obj->adress = str;

obj->year = 2015;

add(mas1, obj, col);

clock\_t b = clock();

int a = search(mas1, obj);

clock\_t e = clock();

print(mas1);

//cout << "Colisions: " << col << endl;

cout << a << " " << ((double)(e - b) / CLOCKS\_PER\_SEC) << endl;

}

Скриншот:

