Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики

и радиоэлектроники»

Специальность «Программная инженерия»

Кафедра инженерной психологии и эргономики

Учебная дисциплина «Основы алгоритмизации и программирования»

Отчет

по лабораторной работе №15

«Хеширование»

Вариант 26

Подготовила: Сёмчена Д. В.

Проверил: Усенко Ф. В.

Минск 2025

**Цель работы**: сформировать знания и умения по работе с подпрограммами, приобрести навыки написания программ с использованием хеш-функций.

Задание: Составить хеш-функцию в соответствии с заданным вариантом и проанализировать ее. При необходимости доработать хеш-функцию. Используя полученную хеш-функцию разработать на языке программирования С++ программу, которая должна выполнять следующие функции: создавать хеш-таблицу; добавлять элементы в хеш-таблицу; просматривать хеш-таблицу; искать элементы в хеш-таблице; удалять элементы из хеш-таблицы.

Листинг кода:

#include<iostream>

#include<string>

#include<windows.h>

using namespace std;

const int table\_size = 2000;

struct hashtable {

int key;

string value;

hashtable() : key(), value("") {}

};

class hashTable {

hashtable table[table\_size];

public:

int HashFunction(const string& x) {

int sqr = 0;

int\* el{ new int[x.length()] };

for (int i = 0; i < x.length(); i++) { el[i] = (int)(x[i]); sqr = sqr + el[i] \* (el[i] / (3 \* (i + 1))); }

sqr = pow(sqr, 2);

string sqrs = to\_string(sqr);

sqrs = sqrs.substr(2, sqrs.length() - 4);

sqr = stoi(sqrs);

while (sqr > 2000) sqr = sqr % 2000;

delete[]el; el = nullptr;

return sqr;

}

void insert(const string& x)

{

int key = HashFunction(x);

table[key].key = key;

table[key].value = x;

}

void show() {

bool emp = true;

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

if (table[i].key != 0) {

cout << "Ключ: " << table[i].key << " Значение: " << table[i].value << endl;

emp = false;

}

}

if (emp) cout << "Хэш-таблица пуста" << endl;

}

void search(const string& x) {

int k = HashFunction(x);

if (table[k].value == "") { cout << "Такого элемента в таблице нет." << endl; return; }

cout<<"Элемент находится на позиции "<<HashFunction(x)<<endl;

return;

}

void remove(const string& x) {

int key = HashFunction(x);

table[key].value = "";

table[key].key = 0;

}

};

int main() {

setlocale(LC\_ALL, "Russian");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

hashTable ht;

string x; int a; bool d = true;

while (d == true) {

system("cls");

cout << "Выберите действие:\n1 - Добавить элемент в хэш-таблицу \n2 - Просмотреть хэш-таблицу \n3 - Найти элемент в хэш-таблице\n4 - Удалить элемент из хэш-таблицы\n5 - Выйти из программы\n";

cin >> a;

cin.ignore();

switch (a)

{

case 1:

{

string key;

cout << "Введите элемент: ";

getline(cin, key);

ht.insert(key);

cout << "Элемент добавлен" << endl;

system("pause");

break;

}

case 2:

{

ht.show();

system("pause");

break;

}

case 3:

{

string key;

cout << "Введите элемент для поиска: ";

getline(cin, key);

ht.search(key);

system("pause");

break;

}

case 4:

{

string key;

cout << "Введите элемент для удаления: ";

getline(cin, key);

ht.remove(key);

cout << "Элемент удален" << endl;

system("pause");

break;

}

case 5:

{

d = false;

break;

}

default:

{

cout << "Неверно выбрано действие. Пожалуйста, повторите ввод." << endl;

system("pause");

break;

}

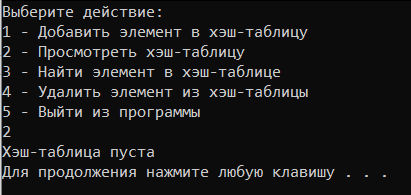
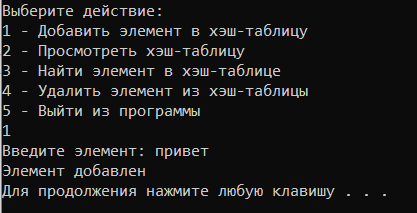
}

}

return 0;

}

Результат работы программы представлен на рисунках 1-4.

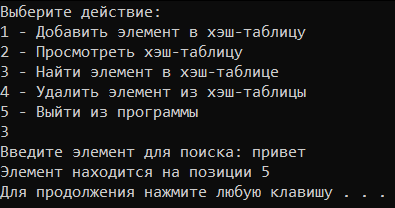
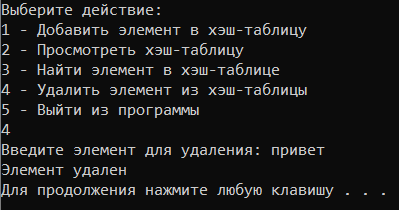
 

Рисунок 1-4 – Результат работы программы

Блок-схема работы программы представлена на рисунках 5-7.

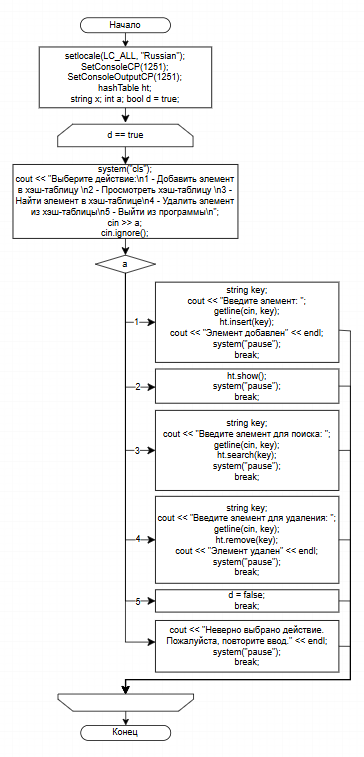


Рисунок 5

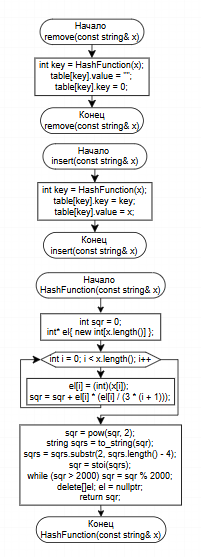
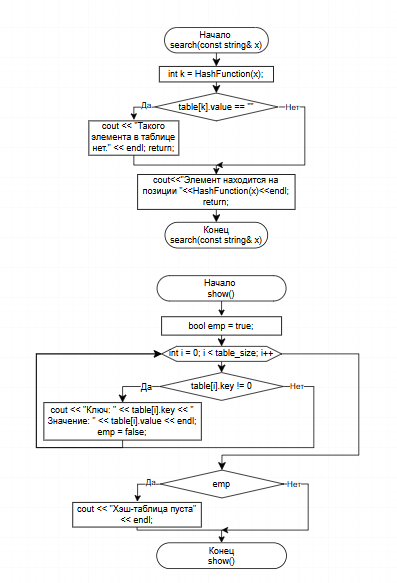
Рисунок 6

Рисунок 7

**Вывод:** В ходе выполнения работы была достигнута цель данной лабораторной работы: сформировать знания и умения по работе с подпрограммами, приобрести навыки написания программ с использованием хеш-функций.