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 14

Подготовил: Зайцев Е. А.

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.

Код программы приведен ниже:

#include <iostream>

#include <string>

#include <Windows.h>

using namespace std;

const int TABLE\_SIZE = 2000;

string\* hashTable[TABLE\_SIZE] = { nullptr }; // массив указателей на строк

int customHash(const string& key) {

long long total = 0;

for (int i = 0; i < key.length(); ++i) {

int code = key[i];

int modifier = code / (3 \* (i + 1)); // порядковый номер начинается с 1

total += code \* modifier;

}

// Возводим сумму в квадрат

long long squared = total \* total;

// Преобразуем в строку, чтобы вырезать цифры

string squaredStr = to\_string(squared);

// Удаляем первые и последние два символа, если это возможно

string middle;

if (squaredStr.length() > 4) {

middle = squaredStr.substr(2, squaredStr.length() - 4);

}

else {

middle = squaredStr;

}

// Преобразуем обратно в число

int middleNumber = stoi(middle);

// Приводим к диапазону от 0 до 1999

int address = middleNumber % 2000;

while (address > 1999) {

address %= 2000;

}

return address;

}

void find\_elem() {

system("cls");

string key;

cout << "Введите строку для поиска: ";

cin.ignore();

getline(cin, key);

int hashValue = customHash(key);

if (hashTable[hashValue] != nullptr && \*hashTable[hashValue] == key) {

cout << "Строка найдена по адресу " << hashValue << ": " << \*hashTable[hashValue] << endl;

}

else {

cout << "Строка не найдена в таблице.\n";

}

system("pause");

system("cls");

}

void delete\_elem() {

system("cls");

string key;

cout << "Введите строку для удаления: ";

cin.ignore();

getline(cin, key);

int hashValue = customHash(key);

if (hashTable[hashValue] != nullptr && \*hashTable[hashValue] == key) {

delete hashTable[hashValue];

hashTable[hashValue] = nullptr;

cout << "Строка удалена из таблицы.\n";

}

else {

cout << "Такой строки нет в таблице.\n";

}

system("pause");

system("cls");

}

void display\_hash() {

system("cls");

cout << "Содержимое хэш-таблицы:\n";

bool empty = true;

for (int i = 0; i < TABLE\_SIZE; ++i) {

if (hashTable[i] != nullptr) {

cout << i << ": " << \*hashTable[i] << endl;

empty = false;

}

}

if (empty) {

cout << "Хэш-таблица пуста.\n\n\n";

}

system("pause");

system("cls");

}

void add\_elem() {

system("cls");

string key;

cout << "\nВведите строку для хэширования (или пустую строку для выхода): ";

cin.ignore();

getline(cin, key);

while(key.empty()){

cout << "Некорректный запрос\n";

cin.ignore();

getline(cin, key);

}

int hashValue = customHash(key);

if (hashTable[hashValue] != nullptr) {

cout << "Коллизия! Ячейка " << hashValue << " уже содержит: \"" << \*hashTable[hashValue] << "\"\n";

}

else {

hashTable[hashValue] = new string(key);

cout << "Строка сохранена по адресу " << hashValue << endl;

}

}

int main() {

setlocale(LC\_ALL, "Russian");

int choise;

SetConsoleCP(1251);

system("cls");

while (true) {

cout << "1 - Добавить элемент в хэш-таблицу\n"

<< "2 - Просмотреть хэш-таблицу\n"

<< "3 - Искать элемент в хэш-таблице\n"

<< "4 - Удалить элемент из хэш-таблицы\n"

<< "5 - Выход\n";

cin >> choise;

while (choise < 1 && choise > 5) {

cout << "\nНекорректный запрос\n";

cin >> choise;

}

switch (choise) {

case 1:

add\_elem();

break;

case 2:

display\_hash();

break;

case 3:

find\_elem();

break;

case 4:

delete\_elem();

break;

case 5:

return 0;

break;

}

}

return 0;

}

Результат работающей программы представлен на рисунке 1:

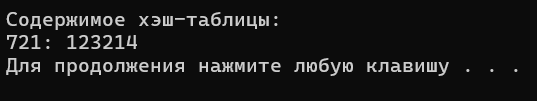


Рисунок 1 – Результат работы программы

Блок-схема работы программы представлена на рисунке 2-7.

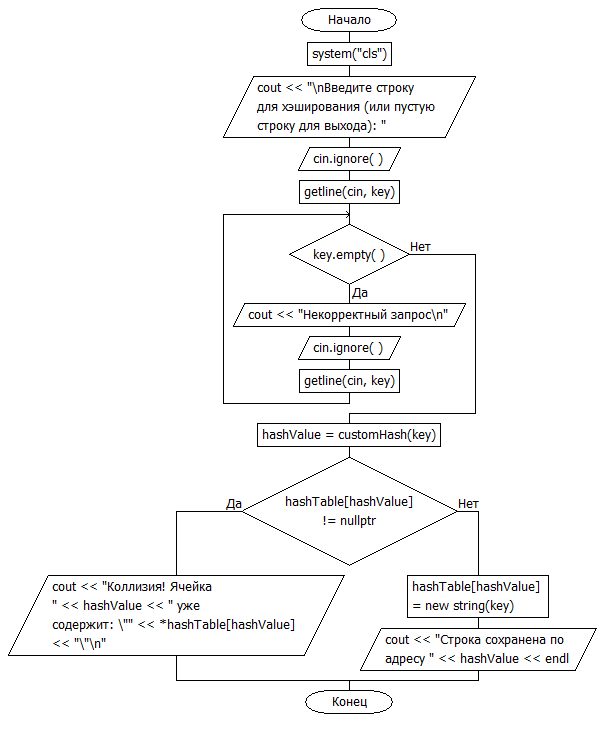


Рисунок 2 – add\_elem()

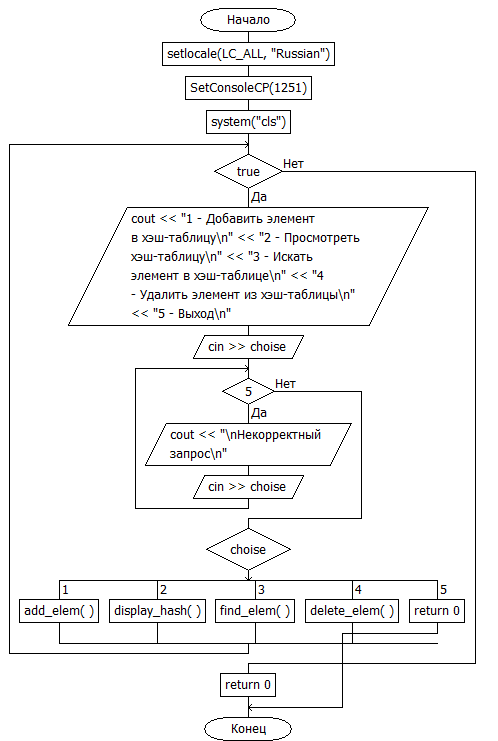


Рисунок 3 - main()

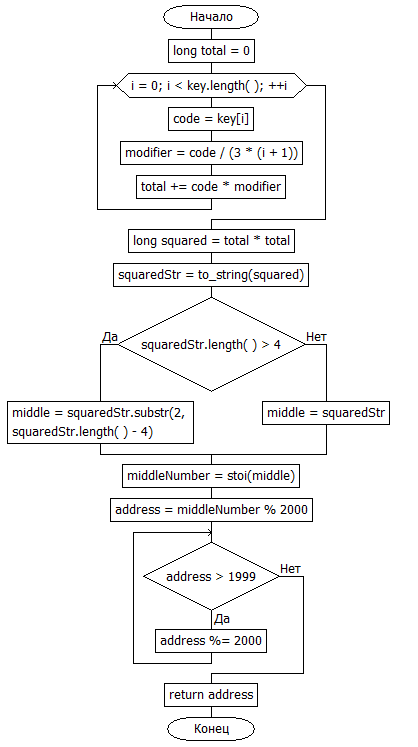


Рисунок 4 - customHash()

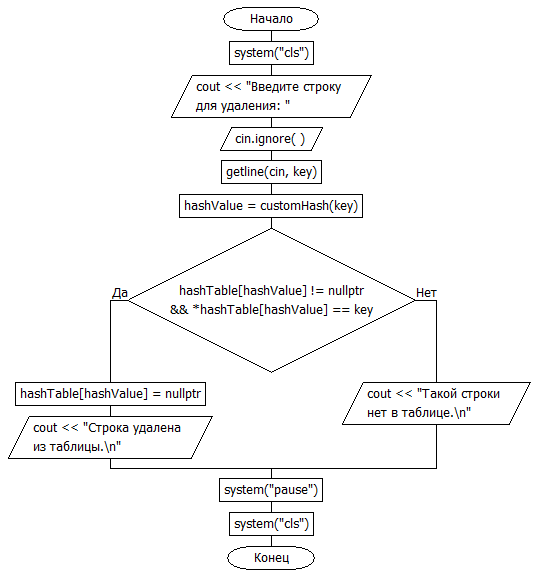


Рисунок 5 – delete\_elem()

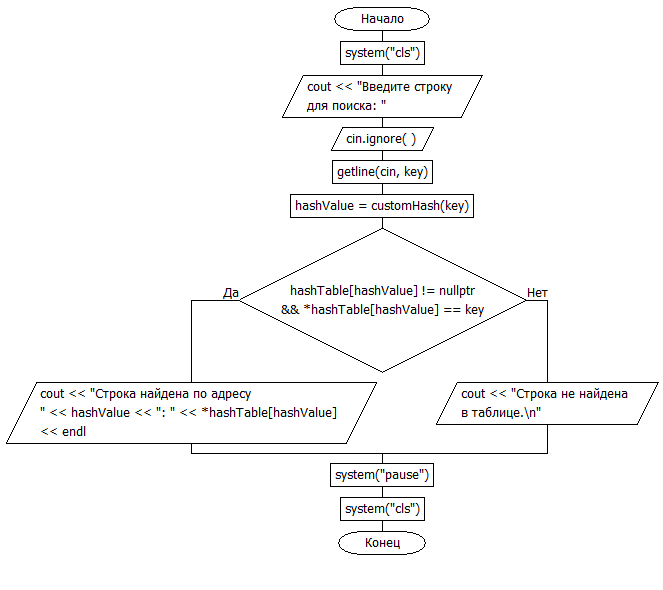


Рисунок 6 – find\_elem()

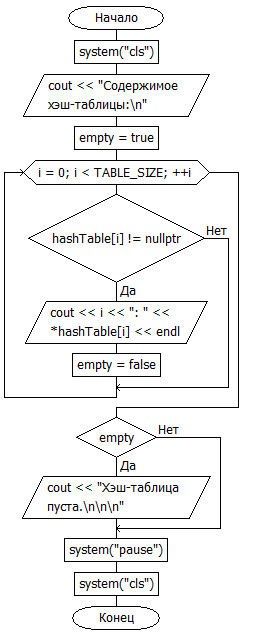


Рисунок 7 – display\_hash()

**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.