Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра автоматизированных систем управления (АСУ)

**“Хэш-таблицы”**

**Отчет по лабораторной работе №5**

**По дисциплине**

**«Структуры и алгоритмы обработки данных в ЭВМ»**

Студент гр. 431-3

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.П. Андреев

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Проверил: профессор кафедры АСУ, д.т.н.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Н. Горитов

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Томск 2023

# 1.Задание на лабораторную работу

Написать программу, которая реализует метод закрытого хеширования с линейной последовательностью проб и хеш-функцией, основанной на методе деления с остатком. Данные, хранящиеся в файле занести в хеш-таблицу. Файл должен содержать не менее 15 целых чисел. Вывести построенную хеш-таблицу на экран (вместе с количеством выполненных проб). Организовать поиск и удаление данных в хеш-таблице. Результаты поиска данных вывести на экран. Также вывести количество проб, которые были затрачены при поиске.

# 2.Алгоритм решения задачи

Первый шаг — это написание структуры хэш-таблицы с полями: индекс, ключ и попытки. Далее прописываем операции инициализации, хэш функцию, добавления, вывод на экран, поиск и удаление. В основной части создаем хэш-таблицу. Заполняем её из файла, после чего выводим её на экран. Далее производим поиск рандомного числа и выводим результат на экран.

# 3.Листинг программы

#include <iostream>

#include <fstream>

using namespace std;

const int hash\_size = 20;

struct Hash

{

int data; //ключ

int index; //индекс

int tries; //попытки

};

Hash hash\_array[hash\_size];

void init()//Инициализация

{

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

{

hash\_array[i].tries = 0;

hash\_array[i].index = i;

}

}

int getHash(int number)

{

return (number % hash\_size);

}

void Add(int Index, int number)//Добавление

{

bool found = false;

int i = 0;

while (i != hash\_size)

{

Index = (number + i) % hash\_size;

if (hash\_array[Index].data == 0 || hash\_array[Index].data == -1)

{

hash\_array[Index].data = number;

hash\_array[Index].tries = i;

found = true;

break;

}

else

{

i++;

}

}

if (found != true)

{

cout << "Хеш-таблица заполнена" << endl;

}

}

void Print()//Вывод

{

cout << "Хэш-Таблица:" << endl;

cout << "Индекс " << " Ключ " << "Попытки:" << endl;

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

{

cout << "---------------------" << endl;

cout << " " << hash\_array[i].index << " " << hash\_array[i].data << " " << hash\_array[i].tries << endl;

}

}

void Search(int number)//Поиск

{

int i = 0;

bool found = false;

while (i != hash\_size)

{

int index = (number + i) % hash\_size;

if (hash\_array[index].data == number)

{

cout << "Число " << number << " найдено" << endl;

cout << "Кол-во попыток при поиске "<<i << endl;

found = true;

break;

}

else if (hash\_array[index].data == 0)

{

break;

}

i++;

}

if (found != true)

{

cout << "Число " << number << " не найдено" << endl;

}

}

void Delete(int number)//Удаление

{

int i = 0;

bool found = false;

while (i != hash\_size)

{

int index = (number + i) % hash\_size;

if (hash\_array[index].data == number)

{

hash\_array[index].data = -1;

hash\_array[index].tries = -1;

cout << "Число "<< number << " удалено" << endl;

found = true;

break;

}

i++;

}

if (found != true)

{

cout << "Число " << number << " не найдено" << endl;

}

}

int main()

{

setlocale(LC\_ALL, "RUS");

int number, Index;

init();

ifstream File("IN.txt");

if (File.is\_open())

{

while (File.eof() == 0)

{

File >> number;

Index = getHash(number);

Add(Index, number);

}

File.close();

Print();

}

else

{

cout << "Файл не открыт";

}

cout << endl;

Search(1+rand()%100);

}

# 4.Пример решения

Входные данные можно увидеть на рисунке 4.1.

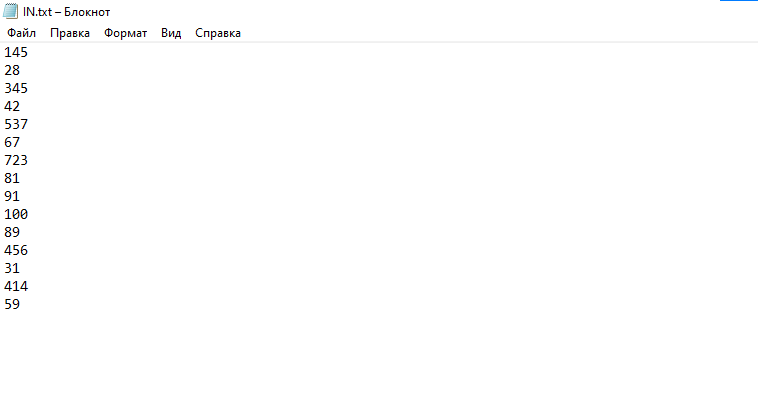


Рисунок 4.1— Входные данные из файла IN.txt

Результат программы можно увидеть на рисунке 4.2.

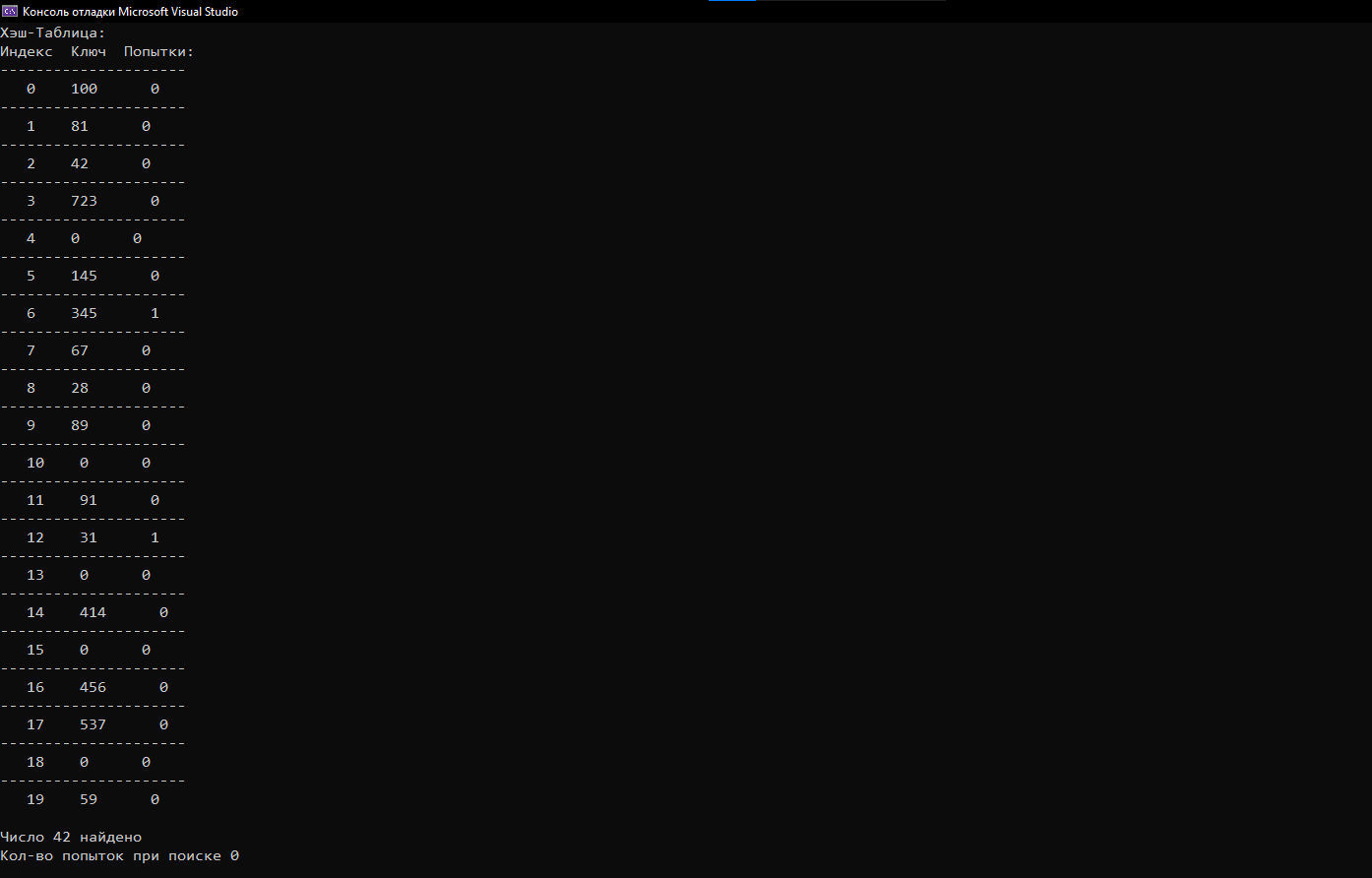


Рисунок 4.2— Выходные данные после запуска программы

# 5.Вывод

В результате лабораторной работы были изучены хэш-таблицы, методы хэширования, последовательность проб, методы хэш-функций и операции для взаимодействия с этой структурой данных.