Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

кафедра «Вычислительных методов и программирования»

**ОТЧЁТ**

По лабораторной работе №10

«ХЕШИРОВАНИЕ»

Выполнил:

Студент ФИТиУ

гр. 020601

Гудков А. С.

Вариант №7

Проверил:

ассистент кафедры ВМиП

Беспалов С. А.

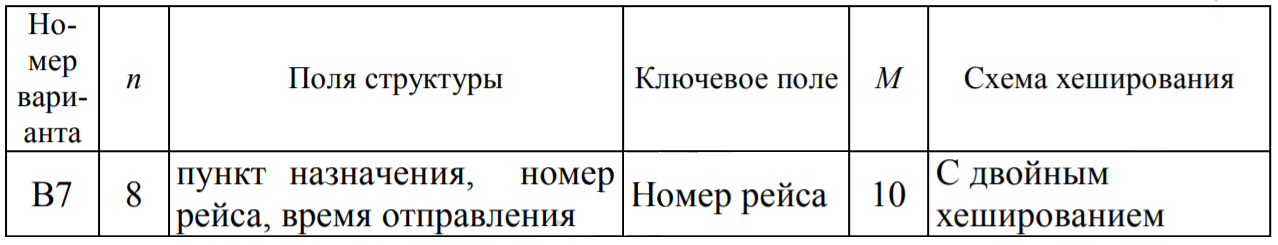
Минск 2021

***Цель работы:***

Научиться реализовывать алгоритмы хеширования, создавать хеш-таблицы, добавлять в них данные и вести поиск по ключу. Написать и отладить программу для работы хеш-таблицой.

***Индивидуальное задание:***

Объявить и ввести массив структур из n элементов. Создать хеш-таблицу из М элементов. Осуществить поиск элемента по ключу в хеш-таблице. Вывести на экран исходный массив, хеш-таблицу и все поля найденной структуры.



***Текст программы:***

#include <iostream>

#include <stdio.h>

using namespace std;

struct **Flight\_Info**

{

char dest[20];

int number;

char time[10];

} \*flights;

int n = 0;

struct **Hash\_Table\_Double**

{

Flight\_Info\*\* table = nullptr; // array of pointers to struct

int size = 0;

void **create**(int m)

{

table = new Flight\_Info\*[m];

for (int i = 0; i < m; i++) table[i] = nullptr;

size = m;

}

void **add**(Flight\_Info\* flight)

{

int i = flight->number % size;

if (table[i])

{

int j = (size-1) - (flight->number % (size-1));

while (table[i])

{

i -= j;

if (i < 0) i += size;

}

}

table[i] = flight;

}

Flight\_Info\* **search**(int number)

{

int i = number % size;

if (table[i] && table[i]->number == number) return table[i];

int j = (size-1) - (number % (size-1));

while (table[i])

{

if (table[i]->number == number) return table[i];

i -= j;

if (i < 0) i += size;

}

return nullptr;

}

void **show**()

{

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

{

if (table[i]) cout << "[" << i << "] " << table[i]->number << endl;

else cout << "[" << i << "]" << endl;

}

}

void **clear**()

{

if (table)

{

delete[] table;

table = nullptr;

}

}

} H;

void read\_data\_from\_file();

void print\_data();

void create\_hash\_table();

void print\_hash\_table();

void search\_in\_hash\_table();

int **main**()

{

while (true)

{

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

cout << "| 1 - Read data from file |" << endl;

cout << "| 2 - Print data |" << endl;

cout << "| 3 - Create hash table |" << endl;

cout << "| 4 - Print hash table |" << endl;

cout << "| 5 - Search in hash table |" << endl;

cout << "| 7 - Exit |" << endl;

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

int menu; cin >> menu;

cout << endl;

switch (menu)

{

case 1: read\_data\_from\_file(); break;

case 2: print\_data(); break;

case 3: create\_hash\_table(); break;

case 4: print\_hash\_table(); break;

case 5: search\_in\_hash\_table(); break;

default: H.clear(); return 0;

}

cout << endl;

system("pause");

system("cls");

}

}

void **read\_data\_from\_file**()

{

if (n > 0)

{

cout << "Data already exists." << endl;

return;

}

char fl\_name[20];

cout << "Enter file name: "; cin >> fl\_name;

FILE\* fl;

fopen\_s(&fl, fl\_name, "r");

if (fl)

{

fscanf\_s(fl, "%d", &n);

flights = new Flight\_Info[n];

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

{

fscanf\_s(fl, "%s", flights[i].dest, 20);

fscanf\_s(fl, "%d", &flights[i].number);

fscanf\_s(fl, "%s", flights[i].time, 10);

}

cout << "Data readed." << endl;

}

else cout << "File open error." << endl;

}

void **print\_data**()

{

if (n > 0)

{

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

{

cout << "Number: " << flights[i].number << ", Dest: " << flights[i].dest << ", Time: " << flights[i].time << endl;

}

}

else cout << "No data." << endl;

}

void **create\_hash\_table**()

{

if (H.size > 0)

{

cout << "Hash table already exists" << endl;

return;

}

int m;

cout << "Enter table size: "; cin >> m;

if (m > 0 && m >= n)

{

H.create(m);

for (int i = 0; i < n; i++) H.add(&flights[i]);

cout << "Hash table created." << endl;

}

else cout << "Create eror." << endl;

}

void **print\_hash\_table**()

{

if (H.size == 0)

{

cout << "Hash table not exist." << endl;

return;

}

cout << "Hash table:" << endl;

H.show();

}

void **search\_in\_hash\_table**()

{

if (H.size == 0)

{

cout << "Hash table not exist." << endl;

return;

}

int number;

cout << "Enter flight number: "; cin >> number;

Flight\_Info\* ptr = H.search(number);

if (ptr) cout << "Number: " << ptr->number < ", Dest: "

<< ptr->dest << ", Time: " << ptr->time << endl;

else cout << "Flight not found." << endl;

}

***Результат работы программы:***

