БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет информационных технологий и управления

Кафедра вычислительных методов и программирования

ОТЧЕТ

по лабораторной работе

**«Программирование с использованием хеширования»**

|  |  |  |
| --- | --- | --- |
| Выполнила |  | Семенова В. С. |
| Проверила |  | Шатилова О. О. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Минск 2023

**Цель**: изучить алгоритмы обработки данных с использованием

хеш-таблиц.

**Общая постановка задачи**

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

**Вариант 11**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *n* | *Поля структуры* | *Ключевое поле* | *m* | *Схема хеширования* |
| 7 | Фамилия, кол-во очков, занятое место | Занятое место | 15 | С двойным хешированием |

**Листинг**

#include <iostream>

#include <string>

#include <string.h>

#include <windows.h>

#include <math.h>

using namespace std;

struct Uchastnik{

string FIO; int score, place;

};

Uchastnik\* dobaw\_m(Uchastnik\*, int\*);

void dobaw\_H(Uchastnik\*, Uchastnik, int );

bool proverka(string D);

void dobaw\_H(Uchastnik\* H, Uchastnik e, int m) {

int c, j;

j = e.place % m;

if (H[j].place == -1) { H[j] = e; }

else {

while (H[j].place != -1) {

c = 1 + (e.place % (m - 2));

j -= c;

if (j < 0) j = j + m;

}

H[j] = e;

}

}

Uchastnik\* dobaw\_m(Uchastnik\* a, int \*n) {

Uchastnik\* tmp; string k;

(\*n)++;

tmp = new Uchastnik[\*n];

for (int i = 0; i < \*n - 1; i++) {

tmp[i].FIO = a[i].FIO;

tmp[i].score = a[i].score;

tmp[i].place = a[i].place;

}

delete[]a;

cout << "\nФИО: "; cin >> tmp[\*n-1].FIO;

do {

cout << "Счет: ";

cin >> k;

} while (!proverka(k));

tmp[\*n-1].score = stod(k);

do {

cout << "Место: ";

cin >> k;

} while (!proverka(k));

tmp[\*n-1].place = stod(k);

return tmp;

}

int main() {

setlocale(LC\_ALL, "Russian");

SetConsoleCP(1251);

int i, j, n, m, c, k\_; Uchastnik\* uch, \* H; string k; char p;

do {

cout << "Введите кол-во элементов массива: ";

cin >> k;

} while (!proverka(k)); n = stod(k);

m = 2 \* n;

uch = new Uchastnik[n];

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

cout << "\nФИО: "; cin >> uch[i].FIO;

do {

cout << "Счет: ";

cin >> k;

} while (!proverka(k));

uch[i].score = stod(k);

do {

cout << "Место: ";

cin >> k;

} while (!proverka(k));

uch[i].place = stod(k);

}

H = new Uchastnik[m];

for (i = 0; i < m; i++) H[i].place = -1;

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

j = uch[i].place % m;

if (H[j].place == -1) { H[j] = uch[i]; }

else {

while (H[j].place != -1) {

c = 1 + (uch[i].place % (m - 2));

j -= c;

if (j < 0) j = j + m;

}

H[j] = uch[i];

}

}

j = 0;

while (true) {

cout << "\n 1 - Добавить участника;\n 2 - Удалить участника;\n 3 - Вывести хеш-таблицу;\n 4 - Вывести исходный массив;\n 5 - Поиск участника по занятому месту;\n 0 - Выход;\n" << endl;

cin >> p; k = "";

switch (p) {

case'1':

if (n < m) {

uch = dobaw\_m(uch, &n); dobaw\_H(H, uch[n - 1], m);

}

else cout << "Хеш-таблица переполнена! Сначала удалите элемент" << endl; break;

case '2': if (n == 0) {

cout << "Таблица пуста!" << endl; break;

}

do {

cout << "Введите место участника, которого хотите удалить: "; cin >> k;

} while (!proverka(k)); k\_ = stod(k);

for (i = 0; i < n; i++) if (uch[i].place == k\_) j = i;

if (j == n) {

cout << "Такого участника нет!" << endl; break;

}

uch[j] = uch[n - 1]; n--;

H[k\_ % m].place = -1; break; cout << "Участник удален" << endl;

case '3':

cout << "\nХэш-таблица----------------------------------------------------" << endl;

for (i = 0; i < m; i++) {

if (H[i].place == -1) cout << " - - - " << endl;

else cout << H[i].FIO << " " << H[i].score << " " << H[i].place << endl;

}

break;

case'4':

cout << "\nИсходный массив----------------------------------------------------" << endl;

for (i = 0; i < n; i++) cout << uch[i].FIO << " " << uch[i].score << " " << uch[i].place << endl;

break;

case'5':

do {

cout << "\nВведите место, которое занял искомый участник: "; cin >> k;

} while (!proverka(k)); k\_ = stod(k);

if (H[k\_ % m].place == k\_) cout << "\nФИО: " << H[k\_ % m].FIO << " Очки: " << H[k\_ % m].score << " Место: " << H[k\_ % m].place;

else {

if (H[k\_ % m].place == -1) cout << "Такого элемента нет!" << endl;

else {

while (H[k\_ % m].place != k\_ && j <= m) {

c = 1 + (k\_ % (m - 2));

j -= c;

if (j < 0) j = j + m;

}

cout << "\n\nИскомый участник: \nФИО: " << H[k\_ % m].FIO << " Очки: " << H[k\_ % m].score << " Место: " << H[k\_ % m].place;

}

}

break;

case'0': delete[]H; delete[]uch; return 0;

default:cout << "Некорректный ввод!" << endl;

}

}

}

bool proverka(string D) {

int p = 0, u = 0, i = 1;

if (D[0] == '-') {

p++;

}

else {

if (D[0] == '.') {

p++; u++;

}

else {

for (char ii = '0'; ii <= '9'; ii++) {

if (D[0] == ii) {

p++;

}

}

}

}

if (p == 1) {

while (D[i] != '\0') {

if (D[i] == ' ') {

return false;

}

for (char ii = '0'; ii <= '9'; ii++) {

if (D[i] == ii) {

p++;

}

}

if (u == 0) {

if (D[i] == '.') {

p++; u++;

}

}

i++;

}

}

else {

return false;

}

if (p != i) {

return false;

}

if (i + 1 > 1 && D[0] == '0' && D[1] != '.') {

return false;

}

if (i + 1 > 1 && D[0] == '-' && D[1] == '0' && D[2] != '.') {

return false;

}

return true;

}