

Федеральное государственное автономное образовательное учреждение высшего образования
«Санкт-Петербургский политехнический университет Петра Великого»
Институт компьютерных наук и технологий
Высшая школа киберфизических систем и управления

УДК 004.421

УТВЕРЖДАЮ

« ____ » _____ г.

ОТЧЕТ
по дисциплине «Теория и технологии программирования»

Лабораторная работа №8

Выполнил:

студент гр. 3532703/00001

_____ Д.П. Кимельфельд
подпись, дата

Санкт-Петербург 2021 г.

Оглавление

Введение	3
1 Текст программы	4
2 Пример работы	7
Выводы	9

Введение

1. Цель работы

Цель задания - овладеть методами хэширования.

2. Задание

Номер варианта №13.

Составить программу для поиска по хэшам данных. Хэширование проводить в соответствии с индивидуальными заданиями. В модуле поиска предусмотреть реализацию обработки случая, при котором хэш-коды различных данных совпадают.

Вариант	Индивидуальный вариант задания
13	Дана таблица текстовой базы данных записями: фамилия; имя; отчество. Произвести хэширование вместе трех полей и поиск по запросу «Фамилия Имя Отчество» и «Фамилия И.О.»

1 Текст программы

```
using namespace std;
#include <iostream>
#include <string>
#include <algorithm>
#include <fstream>
#include <vector>
#include <sstream>
#include <Windows.h>
#include <iomanip>
// Сделать хэширование
struct stringTable {
    string firstName; // Фамилия
    string middleName; // Имя
    string lastName; // Отчество
    long hash;
    int number;
};
long hashFunction(stringTable personality) {
    string *arr[3] = { &personality.firstName, &personality.middleName, &personality.lastName };
};
    long hash = 0;
    for (int i = 0; i < 3; i++) {
        string a = *arr[i];
        for (int j = 0; (j < a.size()) && (i == 0) ; j++) hash += (-1)*int(a[j]) ;
        if (i != 0 ) hash += (-1)*a[0] ;
    }
    return hash % 100000 ;
};
void seperation(const string &text, stringTable &personality){
    istringstream instr(text);
    char I, O;
    instr >> personality.firstName >> personality.middleName >> personality.lastName;
    if (personality.middleName.size() == 4 && personality.lastName == "") {
        I = personality.middleName[0];
        O = personality.middleName[2];
        personality.middleName = I;
        personality.lastName = O;
    }
};
bool checkInput(stringTable &key) {
    for (int i = 0; i < key.middleName.size(); i++) if (key.middleName[i] == '.')
key.middleName.erase(i);
    for (int i = 0; i < key.lastName.size(); i++) if (key.lastName[i] == '.')
key.lastName.erase(i);
    return 0;
}
void sortArray(vector <stringTable> &array, int n) {
    for (int i = 0; i < n - 1; i++) {
        int smallestIndex = i;
        for (int j = i + 1; j < n; j++) if (array[smallestIndex].hash > array[j].hash)
smallestIndex = j;
        swap(array[smallestIndex], array[i]);
    }
}
void binarySearch(vector <stringTable> &array, stringTable key, int &sumBinary, int n , int
index[]) {
    sortArray(array, n);
    int sum = 0;
    int left = 0;
    int right = size(array) - 1;
    int mid = 0;
    while (1) {
```

```

        mid = (left + right) / 2;
        if (key.hash < array[mid].hash) right = mid - 1;
        else if (key.hash > array[mid].hash) left = mid + 1;
        else { // ключ найден
            int save = mid + 1;
            while (array[mid].hash == key.hash && mid > 0) {
                index[sum] = mid;
                sum++;
                mid--;
            }
            if (array[0].hash == key.hash) {
                index[sum] = 0;
                sum++;
            }
            while (array[save].hash == key.hash && save < size(array) - 1) {
                index[sum] = save;
                sum++;
                save++;
            }
            if (array[size(array) - 1].hash == key.hash) {
                index[sum] = array.size() - 1;
                sum++;
            }
            sumBinary = sum;
            break;
        }
        if (left > right) break;
    }
}

void checkCollision(int arrInd[], vector <stringTable> p, stringTable key, int &n, int &C) {
    bool IO = false;
    if (key.middleName.size() == 1 && key.lastName.size() == 1) IO = true;
    for (int i = 0; i < n && !IO; i++) {
        if (p[arrInd[i]].firstName == key.firstName && p[arrInd[i]].middleName ==
key.middleName && p[arrInd[i]].lastName == key.lastName) {
            continue;
        }
        else {
            arrInd[i] = -1;
            C++;
        }
    }
    for (int i = 0; i < n && IO; i++) {
        if (p[arrInd[i]].firstName == key.firstName && p[arrInd[i]].middleName[0] ==
key.middleName[0] && p[arrInd[i]].lastName[0] == key.lastName[0]) {
            continue;
        }
        else {
            arrInd[i] = -1;
            C++;
        }
    }
}

int main() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    ifstream data("база данных.txt");
    char end = '\0';
    string field;
    int n = 0;
    vector < stringTable > personality;
    cout << setw(13) << "\tФамилия" << setw(17) << "\tИмя " << setw(20) << " \tОтчество" <<
setw(10) << "\tХэш" << endl;
    do {

```

```

        getline(data, field);
        if (field.size() > 0) {
            stringTable p0;
            p0.number = n + 1;
            seperation(field, p0);
            p0.hash = hashFunction(p0);
            personality.push_back(p0);
        }
        end = data.get();
        data.putback(end);
        cout << setw(20) << personality[n].firstName << setw(20) <<
personality[n].middleName << setw(20) << personality[n].lastName << setw(10) <<
personality[n].hash << endl;
        n++;
        field = "";
    } while (end != EOF);
    //Ввод на поиск по массиву, сверка коллизий, а затем вывод всей работы ( разница с
колизией и найденные по запросу)
    stringTable key;
    string input;
    do {
        cout << "Введите полное ФИО или полную фамилию и инициалы (Фамилия И. О.) для
осуществления поиска по исходной таблице:" << endl;
        getline(cin, input);
        seperation(input, key);
    } while (checkInput(key));
    key.hash = hashFunction(key);
    int sumBinary = 0;
    int collisionIndex[100];
    int C = 0;
    binarySearch(personality, key, sumBinary, n, collisionIndex);
    // Всего совпадений по хэшу
    cout << "Количество совпадений по хэшу в бинарном поиске: " << sumBinary << endl;
    // Индексы возможного ответа
    if (!sumBinary) {
        cout << "увы, такого человека здесь нет....";
        return 0;
    }
    cout << "Номера строк этих совпадений в исходной таблице: ";
    for (int i = 0; i < sumBinary; i++) {
        if (i == sumBinary - 1) {
            cout << personality[collisionIndex[i]].number;
            break;
        }
        cout << personality[collisionIndex[i]].number << ", ";
    }
    // Исключений коллизий
    checkCollision(collisionIndex, personality, key, sumBinary, C);
    // sumBinary - совпадения, C - кол-во коллизии
    cout << endl << "Количество коллизии в совпадениях: " << C;
    cout << endl << "Итоговый ответ (номер(а) строки в исходной таблице): ";
    for (int i = 0; i < sumBinary; i++)
    {
        int sum = sumBinary - C;
        if (collisionIndex[i] < 0) continue;
        sum--;
        if (i == sumBinary - 1 || sum == 0) {
            cout << personality[collisionIndex[i]].number;
            break;
        }
        cout << personality[collisionIndex[i]].number << ", ";
    }
    system("pause");
    return 0;}

```

2 Пример работы

C:\Users\HUAWEI\source\repos\ТПП_8\х64\Debug\ТПП_8.exe

Фамилия	Имя	Отчество	Хэш
Кимельфельд	Дмитрий	Петрович	351
Юдин	Роман	Олегович	203
Жмурин	Роман	Андреевич	262
Шленцов	Дмитрий	Андреевич	289
Колесов	Иван	Евгеньевич	298
Поляков	Николай	Николаевич	261
Шленцов	Данил	Антонович	289
Калюга	Кирилл	Евгеньевич	283
Климентов	Андрей	Романович	339
Гребнев	Николай	Владимирович	324
Гребнева	Наталья	Владимировна	356
Хлопин	Сергей	Владимирович	251
Хлопин	Сергей	Вальтерович	251
Кимельфельд	Денис	Петрович	351
Кимельфельд	Дмитрий	Пинхасович	351
Юдин	Родион	Осипович	203
Шленцов	Даниэль	Абрамович	289
Путин	Владимир	Владимирович	243
Путин	Всеволод	Владимирович	243
Иванов	Иван	Иванович	297
Иванова	Елизавета	Владимировна	338
Иванова	Екатерина	Венедиктовна	338
Иванов	Илья	Иванович	297
Ленин	Владимир	Ильич	260
Ульянов	Владимир	Ильич	256
Ульянов	Дмитрий	Ильич	254
Ульянов	Владимир	Иванович	256
Жмурин	Рекс	Антонович	262
Малащинский	Максим	Витальевич	385
Малащинский	Марат	Владимирович	385
Нирумж	Роман	Андреевич	262
Нирумж	Намор	Андреевич	265

Введите полное ФИО или полную фамилию и инициалы (Фамилия И. О.) для осуществления поиска по исходной таблице:
Кимельфельд Дмитрий Петрович
Количество совпадений по хэшу в бинарном поиске: 3
Номера строк этих совпадений в исходной таблице: 14, 1, 15
Количество коллизии в совпадениях: 2
Итоговый ответ (номер(а) строки в исходной таблице): 1Для продолжения нажмите любую клавишу . . .

Рисунок 1 – пример №1 выполнения программы

```

C:\Users\HUAWEI\source\repos\ТПП_8\Debug\ТПП_8.exe
    Фамилия      Имя      Отчество      Хэш
    Кимельфельд  Дмитрий  Петрович      351
    Юдин          Роман    Олегович      203
    Жмурин        Роман    Андреевич      262
    Шленцов       Дмитрий  Андреевич      289
    Колесов       Иван     Евгеньевич     298
    Поляков       Николай  Николаевич     261
    Шленцов       Данил    Антонович      289
    Калюга        Кирилл   Евгеньевич     283
    Климентов     Андрей    Романович      339
    Гребнев       Николай  Владимирович   324
    Гребнева      Наталья  Владимировна   356
    Хлопин        Сергей   Владимирович   251
    Хлопин        Сергей   Вальтерович    251
    Кимельфельд  Денис    Петрович      351
    Кимельфельд  Дмитрий  Пинхасович     351
    Юдин          Родион   Осипович      203
    Шленцов       Даниэль  Абрамович      289
    Путин         Владимир Владимирович   243
    Путин         Всеволод Владимирович   243
    Иванов        Иван     Иванович        297
    Иванова       Елизавета Владимировна   338
    Иванова       Екатерина Венедиктовна   338
    Иванов        Илья     Иванович        297
    Ленин         Владимир Ильич      260
    Ульянов       Владимир Ильич      256
    Ульянов       Дмитрий  Ильич          254
    Ульянов       Владимир Иванович    256
    Жмурин        Рекс     Антонович      262
    Малащинский   Максим   Витальевич     385
    Малащинский   Марат    Владимирович   385
    Нирумж        Роман    Андреевич      262
    Нирумж        Намор    Андреевич      265
Введите полное ФИО или полную фамилию и инициалы (Фамилия И. О.) для осуществления поиска по исходной таблице:
Кимельфельд Д.П.
Количество совпадений по хэшу в бинарном поиске: 3
Номера строк этих совпадений в исходной таблице: 14, 1, 15
Количество коллизии в совпадениях: 0
Итоговый ответ (номер(а) строки в исходной таблице): 14, 1, 15Для продолжения нажмите любую клавишу . . .

```

Рисунок 2 - пример №2 выполнения программы

Выводы

Лабораторная работа №8 была выполнена успешно. В ходе данной работы я узнал о том, что такое хэширование и реализациях алгоритмов хэширования. Также применил эти знания на практике. Затем столкнувшись с неправильным поиском позиций, я понял, что мой способ хэширования делает осечки, а именно имеет коллизии. Я также узнал о такой вещи и исключил их из своей программы, путем точной сверки позиций в таблице.