|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования FPMI_ngtu_neti_rgb_polya«Новосибирский государственный технический университет» | | | |
|  | | | |
| Кафедра прикладной математики | | | |
| Практическое задание № 4+ | | | |
| по дисциплине «Структуры данных и алгоритмы» | | | |
| **«Управление таблицами»** | | | |
|  | | | |
|  | Группа | ПМ-21 | |
| Бригада 7 | Егупов иван | |
|
|
|  | |
|  | |
| Преподаватель | ХИЦЕНКО ВАЛЕНТИНА ПАВЛОВНА | |
| Дата | 15.05. | |
| Новосибирск,2023 | | | |

**1.Условие задачи**

В текстовом файле содержится список телефонных разговоров. Элемент списка включает номер телефона – 10 символьный код, ФИО абонента не более 30 символов, количество минут разговора, стоимость минуты разговора. Добавить эту информацию в упорядоченную таблицу «Оплата телефонных разговоров» размером не более 500. Элемент таблицы включает номер телефона – 10 символьный код (и это ключ таблицы), ФИО абонента не более 30 символов, количество минут разговора, стоимость минуты разговора и сумму к оплате. Сумма к оплате вычисляется умножением количества минут на стоимость минуты. Поиск места элемента в таблице организовать используя бинарный поиск

**2. Анализ задачи**

Входные данные:

Table = {elemi = (numberi, fioi, timei, pricei, sumi)| numberi ∈ G, fioi ∈ M, time, pricei, sumi ∈ N,i<=V}V = 500

G = {gi|g i∈ {’1’,’2’…’0’}, i = 1,10}

M = {mi|m i∈ {‘a’,’b’… ‘z’}, i = 1,30}

P = {eli = (numberi, fioi, timei, pricei)| numberi ∈ G, fioi ∈ M, time, pricei ∈ N,i>=1}

Выходные данные:

k ∈ {0;1}

Table = {elemi = (numberi, fioi, timei, pricei, sumi)| numberi ∈ G, fioi ∈ M, time, pricei, sumi ∈ N,i<=500} или ∅

Решение:

**при**  el = первый элемент последовательности P, k=общее количество элементов Table, p = 1;

**повторять** i = индекс в таблице Table, куда нужно поместить el, вычисленный с помощью бинарного поиска. Если k==N И Table[i]number ≠ elnumber, то p = 0

Иначе

*Если* el уже присутсвует в Table, то Table[i]sum = Table[i]sum + elsum; Table[i]time = Table[i]time + eltime

*Иначе* **при** j=k

**повторять** Table[j] = Table[j-1]; j = j-1;

**пока** j>i

Table[i] = el;k = k+1

**пока** ∃el И p = 1

Подзадачи:

*Binary\_search – поиск места для элемента в таблице*

Входные данные: ключ очередного элемента из P, общее количество элементов в Table, Table

Выходные данные: целочисленное значение местоположения элемента в таблице

Add\_in\_table – добавление очередного элемента из P в Table

Входные данные: Информация о файле, в котором содержится P, Table, текущее общее количество элементов в Table

Выходные данные: 0 или 1, если количество элементов в Table увеличилось

*Print\_table –* печать элементов таблицы в файл

Входные данные: Table, информация о файле, в который будет записан результат

**3.Структура программы**

int binary\_search(char num[], int k, Elem table[]) Назначение: поиск позиции для очередного элемента P в таблице Table

char num[] – номер элемента из P, по которому нужно определить позицию

int k – текущее количество элементов в таблице

Elem table[] – упорядоченная таблица, в которой ищем место для элемента из P

Возвращаемое значение – индекс массива table, в который нужно поместить элемент из P

int add\_in\_table(FILE\* add, Elem table[], int k) Назначение: добавление элемента в упорядоченную таблицу

Elem table[] – упорядоченная таблица, в которой ищем место для элемента из P

int k – текущее количество элементов в таблице

FILE \*add – указатель на файл, содержащий элемент из P

void Print\_table(Elem table[], FILE\* output) Назначение: печать элементов таблицы в файл

Elem table[] – таблица, содержащая итоговый ответ

FILE \*output – указатель на файл, куда будет произведена печать

**4.Текст программы**

Заголовочный файл “Table.h”

#pragma once

#include <string.h>

const int N = 500;

struct FIO

{

char fname[15];

char sename[15];

char suname[15];

};

struct Elem

{

char number[11];

struct FIO fio;

int time;

int price;

int sum;

};

struct Table

{

struct Elem elem[N]{};

int n;

};

int binary\_search(char num[], int k, Elem table[])

{

int l = 0, r = k - 1, m = 0;

for (;l < r;)

{

m = (l + r) / 2;

if (strcmp(num, table[m].number) == 1) l = m + 1;

else r = m;

}

if (strcmp(num, table[m].number) == 0) return m;

return strcmp(num, table[m].number) == -1 ? m : m + 1;

}

int add\_in\_table(FILE\* add, Elem table[], int k)

{

int flag = 0;

struct Elem el;

fscanf\_s(add, "%s%s%s%s%d%d", &el.number, 11, &el.fio.fname, 15, &el.fio.sename, 15, &el.fio.suname, 15, &el.time, &el.price);

el.sum = el.time \* el.price;

int i = binary\_search(el.number, k, table); //поиск позиции для элемента

if (k==N && strcmp(el.number, table[i].number)) return 2;

else {

if (!strcmp(el.number, table[i].number)) //если уже присутсвует, то изменить значение sum и time

{

table[i].sum += el.sum;

table[i].time += el.time;

}

else //сдвинуть массив на 1 элемент вправо и в свободное место поло-жить новый элемент

{

for (int j = k; j > i; j--)

{

table[j] = table[j - 1];

}

table[i] = el;

flag = 1;

}

}

return flag;

}

void Print\_table(Elem table[], FILE\* output)

{

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

if (strlen(table[i].number) == 10)

fprintf\_s(output, "%s\t %s %s %s\t %d\t %d\t %d\n", table[i].number, table[i].fio.fname, table[i].fio.sename, table[i].fio.suname, table[i].time, table[i].price, table[i].sum);

}

Основная программа Source.cpp

#include <stdio.h>

#include "Table.h"

int main()

{

Table T;

T.n = 0;

FILE\* tab = NULL;

FILE\* add = NULL;

fopen\_s(&tab, "Table.txt", "r");

for (; !feof(tab);) //считывание упорядоченной таблицы из файла

{

struct Elem el;

fscanf\_s(tab, "%s%s%s%s%d%d%d", &el.number, 11, &el.fio.fname,15,&el.fio.sename,15,&el.fio.suname,15, &el.time, &el.price, &el.sum);

T.elem[T.n] = el;

T.n += 1;

}

fclose(tab);

fopen\_s(&add, "add.txt", "r");

int flag = 0;

char k = 1;

while ((flag != 2) && !feof(add)) //считывание элементов, которые необходимо добавить

{

flag = add\_in\_table(add,T);

switch (flag)

{

case 2: {printf\_s("Table is full"); k = 0; break; }

case 1: {T.n += flag; break; }

default:

break;

}

}

fclose(add);

printf\_s("k = %d\n", k);

fopen\_s(&tab, "Table.txt", "w");

Print\_table(T.elem, tab);

fclose(tab);

return 1;

}

**5.Набор тестов**

|  |  |  |
| --- | --- | --- |
| Входные данные | Выходные данные | Назначение |
| Table.txt:  add.txt:  7775152344 Egupov Ivan Nikolaevich 10 2 | Table.txt:  7775152344 Egupov Ivan Nikolaevich 10 2 20 | Добавление нового элемента в пустую таблицу |
| Table.txt:  7775152344 Egupov Ivan Nikolaevich 10 2 20  add.txt:  7775152344 Egupov Ivan Nikolaevich 10 2 | Table.txt:  7775152344 Egupov Ivan Nikolaevich 20 2 40 | В add.txt содержится элемент, который уже присутсвует в Table.txt |
| Table.txt: 7775152344 Egupov Ivan Nikolaevich 20 2 40  add.txt:  9628127368 Martov Dmitry Nikolaevich 30 7  9059042338 Tashlikov Danil Andreevich 78 4  9224177441 Tadzibaev Zavik Abdukarimo 90 1 | Table.txt:  7775152344 Egupov Ivan Nikolaevich 20 2 40  9059042338 Tashlikov Danil Andreevich 78 4 312  9224177441 Tadzibaev Zavik Abdukarimo 90 1 90  9628127368 Martov Dmitry Nikolaevich 30 7 210 | Добавление в Таблицу трёх новых элементов |
| Table.txt:  7775152344 Egupov Ivan Nikolaevich 20 2 40  9059042338 Tashlikov Danil Andreevich 78 4 312  9224177441 Tadzibaev Zavik Abdukarimo 90 1 90  9628127368 Martov Dmitry Nikolaevich 30 7 210  add.txt:  9628127368 Martov Dmitry Nikolaevich 30 7  9059042338 Tashlikov Danil Andreevich 78 4  9224177441 Tadzibaev Zavik Abdukarimo 90 1  1523232323 Sanina Anastasia xxxxxxx 63 78 | Table.txt:  1523232323 Sanina Anastasia xxxxxxx 63 78 4914  7775152344 Egupov Ivan Nikolaevich 20 2 40  9059042338 Tashlikov Danil Andreevich 156 4 624  9224177441 Tadzibaev Zavik Abdukarimo 180 1 180  9628127368 Martov Dmitry Nikolaevich 60 7 420 | Добавление в таблицу нового элемента с наименьшим номером, а также трёх, которые уже присутствовали. |
| N = 5  Table.txt:  1523232323 Sanina Anastasia xxxxxxx 252 78 19656  7775152344 Egupov Ivan Nikolaevich 20 2 40  9059042338 Tashlikov Danil Andreevich 390 4 1560  9224177441 Tadzibaev Zavik Abdukarimo 450 1 450  9628127368 Martov Dmitry Nikolaevich 150 7 1050  add.txt:  9628127368 Martov Dmitry Nikolaevich 30 7  9059042338 Tashlikov Danil Andreevich 78 4  9224177441 Tadzibaev Zavik Abdukarimo 90 1  1523232323 Sanina Anastasia xxxxxxx 63 78 | Выходные данные:  Table.txt:  1523232323 Sanina Anastasia xxxxxxx 315 78 24570  7775152344 Egupov Ivan Nikolaevich 20 2 40  9059042338 Tashlikov Danil Andreevich 468 4 1872  9224177441 Tadzibaev Zavik Abdukarimo 540 1 540  9628127368 Martov Dmitry Nikolaevich 180 7 1260 | Добавление в полную таблицу существующих элементов |
| N = 5  Table.txt:  1523232323 Sanina Anastasia xxxxxxx 315 78 24570  7775152344 Egupov Ivan Nikolaevich 20 2 40  9059042338 Tashlikov Danil Andreevich 468 4 1872  9224177441 Tadzibaev Zavik Abdukarimo 540 1 540  9628127368 Martov Dmitry Nikolaevich 180 7 1260  add.txt:  8628127368 Trakimus Yuriy Viktorovich 30 7 | Выходные данные:  Table.txt:  1523232323 Sanina Anastasia xxxxxxx 315 78 24570  7775152344 Egupov Ivan Nikolaevich 20 2 40  9059042338 Tashlikov Danil Andreevich 468 4 1872  9224177441 Tadzibaev Zavik Abdukarimo 540 1 540  9628127368 Martov Dmitry Nikolaevich 180 7 1260  Table is full | Попытка добавить новый элемент в полную таблицу. |