Министерство высшего образования и науки Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

(ФГАОУ ВО «ЮФУ»)

Институт компьютерных технологий и информационной безопасности

**ОТЧЁТ**

по индивидуальному заданию

по дисциплине «Основы алгоритмизации и программирования»

Вариант №11

Выполнил

студент группы КТсо1-6 В. А. Овсянникова

Принял

ассистент

кафедры ИАСБ  С. А. Барковский  
  
  
  
  
Таганрог 2020

**Вариант №11.**

1) Задание заключается в том, чтобы перечислить услуги, предоставленные указанному клиенту во 2-м квартале нынешнего года. Отчет записывается в текстовый файл. ФИО и диапазон времени записан в параметр.

2) Листинг:   
#define \_CRT\_SECURE\_NO\_WARNINGS

#define STR\_LEN 32

#include <iostream>

#include <string>

#include <fstream>

#include <vector>

#include <map>

using namespace std;

// Возвращает true, если дата находится во 2 квартале 2020 года

bool is\_in\_second\_part(char date[]) {

int month, year;

// Преобразуем числа формата string в int, отнимая 30 в шестнадцатиричной. (Например, 2 в string = 0x32 в int)

// Обращаемся к конкретным индексам, что бы получить нужные цифры, затем делаем из них единое число

month = (date[3] - 0x30) \* 10 + (date[4] - 0x30);

year = (date[6] - 0x30) \* 1000 + (date[7] - 0x30) \* 100 + (date[8] - 0x30) \* 10 + (date[9] - 0x30);

// Проверяем, находится ли дата во 2 квартале 2020 года

return month >= 4 && month <= 6 && year == 2020;

}

struct Customer {

char fio[STR\_LEN \* 3], date\_start[STR\_LEN], number[STR\_LEN];

vector<string> services; // ID-шники услуг, которые использует клиент

};

int main()

{

// Список(вектор) клиентов

vector<Customer> customers;

// Блок обработки файла с клиентами

{

// Открываем clients.txt в режиме чтения

FILE\* clients\_in = fopen("clients.txt", "r");

// Строка на 128 символов +1 засчет \n и +1 засчет \0

char current\_line[130];

// Считывая очередную строку, сразу проверяем, не NULL-овая ли она

while (fgets(current\_line, 130, clients\_in) != NULL) {

// Имя, фамилия, отчество, даты начала договора и его конца.

char fio[STR\_LEN \* 3], date\_start[STR\_LEN], date\_end[STR\_LEN];

// Телефонный номер клиента

char number[STR\_LEN];

// Зануляем все байты в строках

memset(fio, 0, STR\_LEN \* 3);

memset(date\_start, 0, STR\_LEN);

memset(number, 0, STR\_LEN);

// Задолженность и допустимый кредит не обрабатываем, т.к. задача этого не требует

// Указатели на очередную строку и на поля с вводимыми данными

char\* ptr = current\_line, \* fio\_ptr = fio, \* date\_start\_ptr = date\_start, \* date\_end\_ptr = date\_end, \* num\_ptr = number;

// Блок для обработки очередной строки

{

// Пока не дошли до 1 запятой, считываем ФИО

while (\*ptr != ',') \*fio\_ptr++ = \*ptr++;

while (\*ptr == ',' || \*ptr == ' ') ptr++; // Чтобы пропустить запятую или пробел

// Пока не дошли до 2 запятой, считываем номер

while (\*ptr != ',') \*num\_ptr++ = \*ptr++;

while (\*ptr == ',' || \*ptr == ' ') ptr++; // Чтобы пропустить запятую или пробел

// Пока не дошли до 3 запятой, считываем дату начала

while (\*ptr != ',') \*date\_start\_ptr++ = \*ptr++;

while (\*ptr == ',' || \*ptr == ' ') ptr++; // Чтобы пропустить запятую или пробел

// Пока не дошли до 4 запятой, считываем дату конца

while (\*ptr != ',') \*date\_end\_ptr++ = \*ptr++;

while (\*ptr == ',' || \*ptr == ' ') ptr++; // Чтобы пропустить запятую или пробел

}

// Создание скелета нового клиента

Customer new\_customer;

// Заполнение данных о нём

strcpy(new\_customer.fio, fio);

strcpy(new\_customer.number, number);

strcpy(new\_customer.date\_start, date\_start);

// Добавление очередного клиента в список, если он заключил договор во 2 квартале 2020

if (is\_in\_second\_part(date\_start))

customers.push\_back(new\_customer);

}

fclose(clients\_in);

}

// Блок обработки файла с привязкой номера к услугам

{

// Открываем sptc.txt в режиме чтения

FILE\* clients\_in = fopen("sptc.txt", "r");

// Строка на 128 символов +1 засчет \n и +1 засчет \0

char current\_line[130];

// Считывая очередную строку, сразу проверяем, не NULL-овая ли она

while (fgets(current\_line, 130, clients\_in) != NULL) {

// Телефонный номер клиента и id услуги

char number[STR\_LEN], id[STR\_LEN];

// Зануляем все байты в строках

memset(number, 0, STR\_LEN);

memset(id, 0, STR\_LEN);

char\* num\_ptr = number, \* id\_ptr = id, \* ptr = current\_line;

// Блок для обработки очередной строки

{

// Пока не дошли до 1 запятой, считываем ФИО

while (\*ptr != ',') \*num\_ptr++ = \*ptr++;

while (\*ptr == ',' || \*ptr == ' ') ptr++; // Чтобы пропустить запятую или пробел

// Пока не дошли до 2 запятой, считываем номер

while (\*ptr != ',') \*id\_ptr++ = \*ptr++;

}

// Сопоставление клиентов и использованных услуг

for (int i = 0; i < customers.size(); i++) {

// Сравниваем номера телефонов

if (strcmp(customers[i].number, number) == 0) {

// Добавляем в список услуг клиента ещё услугу

customers[i].services.push\_back(id);

}

}

}

fclose(clients\_in);

}

// Создаем map, где id - ключ, а название услуги значение

map<string, string> services;

// Блок обработки файла с названиями услуг

{

// Открываем services.txt в режиме чтения

FILE\* clients\_in = fopen("services.txt", "r");

// Строка на 128 символов +1 засчет \n и +1 засчет \0

char current\_line[130];

// Считывая очередную строку, сразу проверяем, не NULL-овая ли она

while (fgets(current\_line, 130, clients\_in) != NULL) {

// Название услуги и её id

char name[STR\_LEN \* 3], id[STR\_LEN];

// Зануляем все байты в строках

memset(name, 0, STR\_LEN \* 3);

memset(id, 0, STR\_LEN);

char\* name\_ptr = name, \* id\_ptr = id, \* ptr = current\_line;

// Блок для обработки очередной строки

{

// Пока не дошли до 1 запятой, считываем ФИО

while (\*ptr != ',') \*name\_ptr++ = \*ptr++;

while (\*ptr == ',' || \*ptr == ' ') ptr++; // Чтобы пропустить запятую или пробел

// Пока не дошли до 2 запятой, считываем номер

while (\*ptr != ',') \*id\_ptr++ = \*ptr++;

}

// Добавляем пару "Ключ - значение" в map

services.insert({ id, name });

}

fclose(clients\_in);

}

// Создаем поток для вывода

ofstream out("Report.txt");

// Если нет данных, так и пишем

if (customers.empty()) {

// Нет данных

out << "No data" << endl;

}

// Если есть, выводим их

else {

// Проходим по списку клиентов

for (int i = 0; i < customers.size(); i++) {

out << customers[i].number << endl;

out << "Services:" << endl;

cout << customers[i].number << endl;

cout << "Services:" << endl;

// Проходим по списку услуг клиента

for (int j = 0; j < customers[i].services.size(); j++) {

// Обращаемся к map, чтобы получить название услуги по id

out << "\t" << services[customers[i].services[j]] << endl;

cout << "\t" << services[customers[i].services[j]] << endl;

}

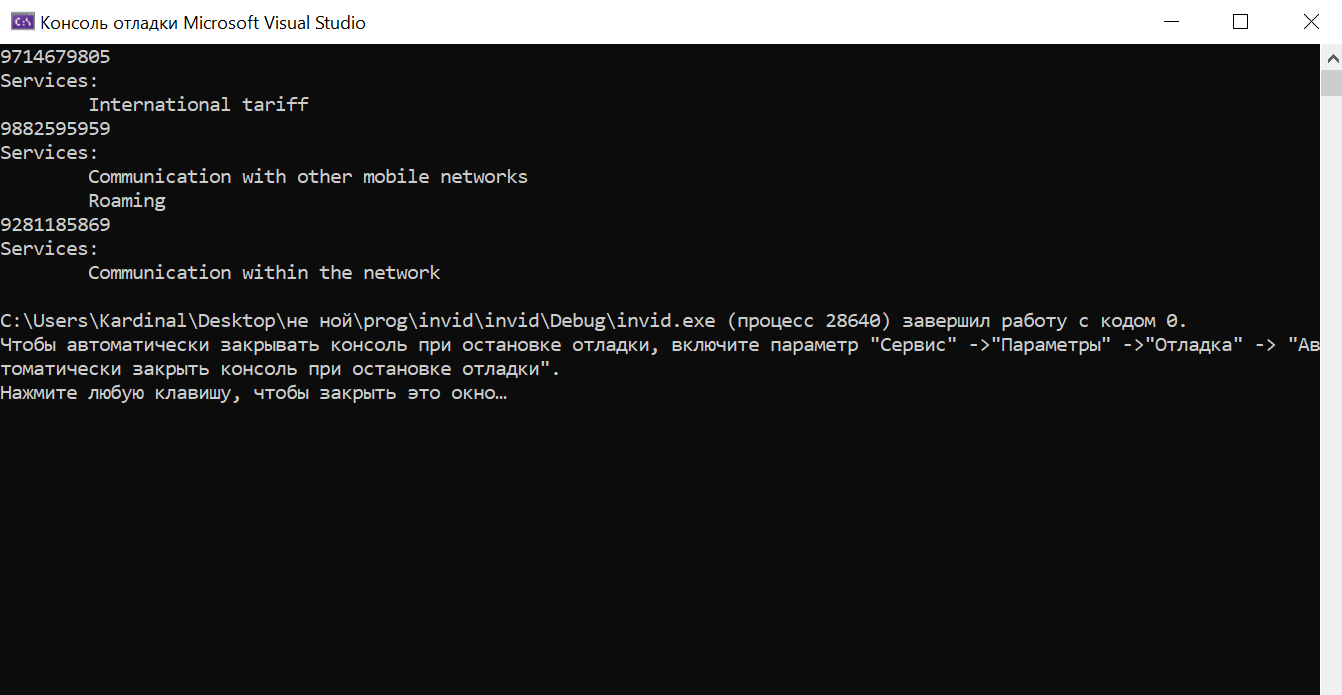
}

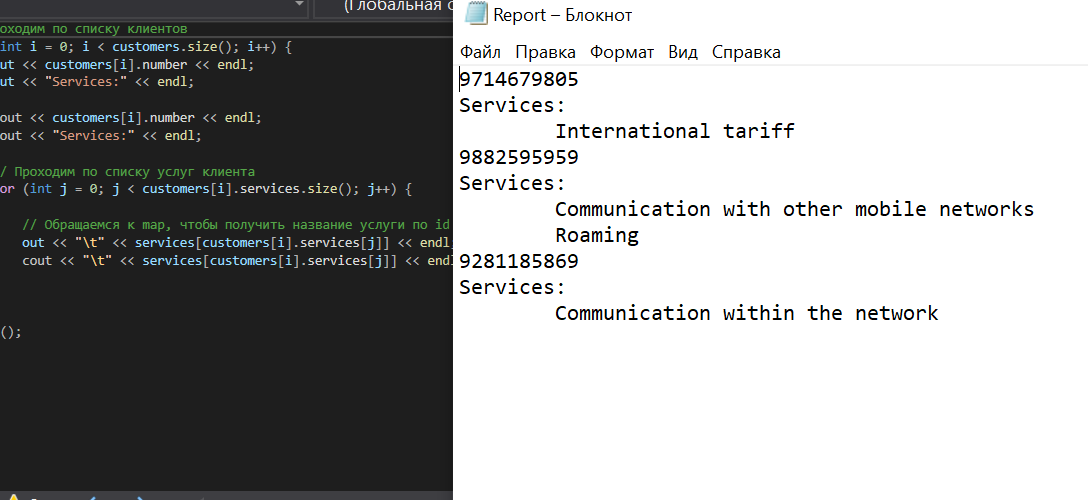
}

out.close();

}  
  
3) Тексты файлов данных:

clients.txt:  
Vasiliev Vasily Vasilievich, 9773672365, 12.10.2020, 12.10.2014,0,0  
Petrov Petr Petrovich, 9734672311, 22.01.2020, 01.11.2020,210,200  
Vasiliev Ilia Vasilievich, 9714679805, 05.06.2020, 01.12.2012,0,2000  
Petrov Petr Petrovich, 1111111111, 01.02.2020, 02.02.2020,0,0  
Bobryshev Svyatoslav Aleksandrovich, 9381173929, 10.12.2001, 11.12.2019,0,0  
Starodubtsev Vitaly Sergeevich, 9882595959, 13.04.2020, 12.12.2019,0,2000  
Grigoryan Marat Karenovich, 9896215028, 19.12.2009, 13.10.2010,0,1000  
Bobrysheva Tatiana Petrovna, 9281185869, 01.05.2020, 01.01.2019,0,3000  
Bobryshev Aleksandr Aleksandrovich, 9876543210, 03.08.2010, 08.03.2019,0,2010  
Valenok Evgeny Aleksandrovich, 1234567890, 01.09.2020, 02.09.2019,1000,0  
  
  
services.txt:  
Communication within the network, 1, 0.30, мин  
Messages, 2, 0.15, #  
Communication with other mobile networks, 3, 0.50, мин  
Roaming, 4, 10, мин  
International tariff, 5, 50, мин  
Home Internet, 6, 600, мес  
Internet outside the home, 7, 8000, день  
Help in setting up the home, 8, 300, #  
Services to remote psychiatrist, 9, 123, #  
Consultation with specialists, 10, 200, #  
  
  
sptc.txt:  
9773672365, 1, 13.02.2008 13:01:55, 300  
9734672311, 2, 28.11.2012 01:32:30, #  
9714679805, 5, 23.09.2013 19:12:00, 54  
1111111111, 7, 03.01.2001 18:18:00, 3600  
9381173929, 4, 18.08.2013 17:39:44, 56  
9882595959, 3, 14.12.2018 15:15:32, 48  
9882595959, 4, 14.12.2018 15:15:32, 48  
9896215028, 1, 14.09.2010 06:12:31, 45  
9281185869, 1, 15.10.2017 18:34:32, 28  
9876543210, 2, 16.09.2018 12:14:28, #  
1234567890, 8, 01.09.2019 12:12:12, #  
  
  
param:  
Messages  
Communication within the network  
13.02.2008  
23.09.2013  
  
  
4)Контрольный пример в виде входных и выходного файла:

Результат на выходной строке и в файле “Report”:  




5) Блок-схема:

