Выполнила

студентка группы КТбо1-2 Н. А. Свидич

Принял

доцент кафедры САиТ А. С. Свиридов

ИНДИВИДУАЛЬНАЯ РАБОТА

по дисциплине «Алгоритмизация и программирование»

**«ИНДИВИДУАЛЬНАЯ РАБОТА №1»**

Вариант 22

Таганрог 2023

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

(ФГАОУ ВО «ЮФУ»)

Институт компьютерных технологий и информационной безопасности

Кафедра системного анализа и телекоммуникаций

**ОГЛАВЛЕНИЕ**

Техническое задание 3

Цель задания 3

Задача 3

Ход работы 4

Алоритм решения 4

Блок-схемы 5

Написание программы 7

Пример работы 11

Вывод 13

Приложение А – Листинг 14

main.cpp 14

# Техническое задание

## Цель задания

Цель данной индивидуальной работы проверить навыки студента работы с файлами, работы со строковыми переменными, работы с функциями.

## Задача

Задание состоит в разработке программы, которая считывает настроечные параметры и формирует отчет по имеющимся данным в файлах. Отчет записывается в текстовый файл. Число записей в каждом из описанных выше файлов произвольно. Вся указанная информация представлена текстовыми файлами, каждое поле отделяется запятой, запись – это строка текста.

Имеется информация о клиентах телефонной компании и предоставляемых им услугах. Каждая услуга имеет собственный тариф, а каждый клиент может пользоваться произвольным набором услуг в течение ограниченного интервала времени (соответственно срокам договора). Имеются данные о фактическом использовании услуг.

Вариант 22.

Для отчета необходимо отобрать все номера телефонов с заданной суммой баланса, которые в течение последней недели говорили каждый день не менее заданного количества минут.

# Ход работы

## Алоритм решения

1. Получить настроечные параметры (искомые минимальные баланс номера и время использования услуги) из файла параметров. Алгоритм чтения .ini файла в виде блок-схемы описан на Рисунке 1.
2. Записать значения настроечных параметров в программные переменные.
3. Получить данные (номер телефона, код услуги, дата оказания услуги, время оказания услуги, баланс номера на момент оказания услуги) из файла исходных данных об оказанных услугах.
4. Записать полученные данные в массив структур.
5. Выполнить алгоритм поиска (Рисунок 2):
   1. Перебрать массив с исходными данными на выполнение условий.
      1. Если баланс номера и время использования услуги больше либо равны заданным параметрам и время оказания услуги находится в диапазоне недели с текущей даты, то занести данный элемент в буфер.
   2. Перебрать массив-буфер с подходящими значениями:
      1. Сравнивать номера телефонов пока не будут найдены семь элементов (на каждый день недели) или пока не закончится массив-буфер.
      2. Если номера телефонов совпадают и даты находятся в диапазоне, тогда пополняется счетчик, подсчитывающий подходящие записи одного и того же номера в течении недели.
      3. Когда счетчик будет равен семи, выполнить остановку и взять следующий номер для сравнения.
   3. Если не будет найдено ни одно значение, вывести соответствующее сообщение. Иначе – вывести найденные номера.

## БЛОК-СХЕМЫ

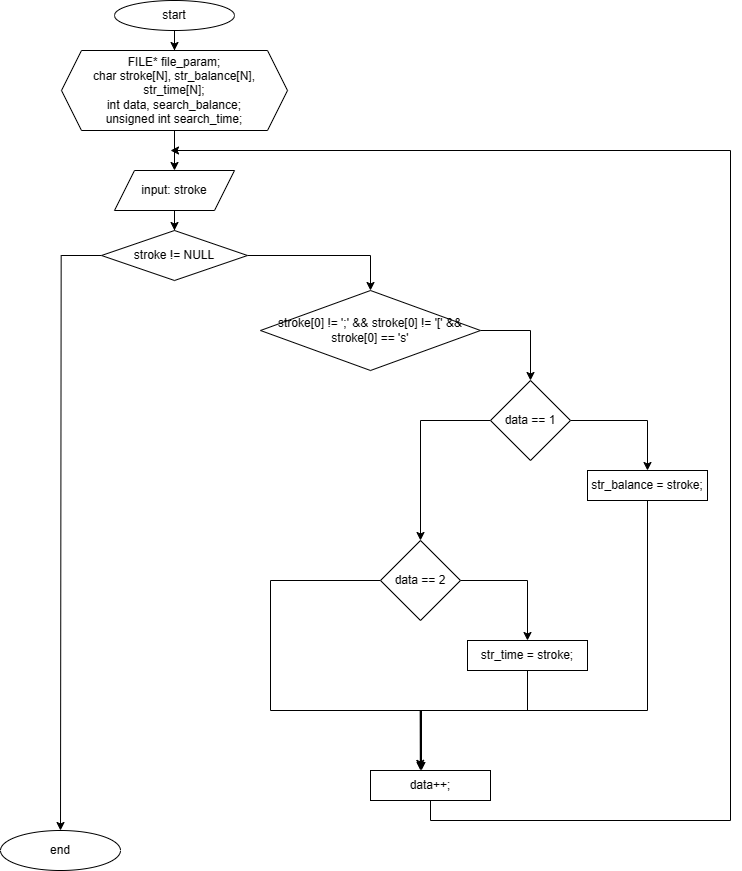


Рисунок 1 – Блок-схема алгоритма чтения .ini

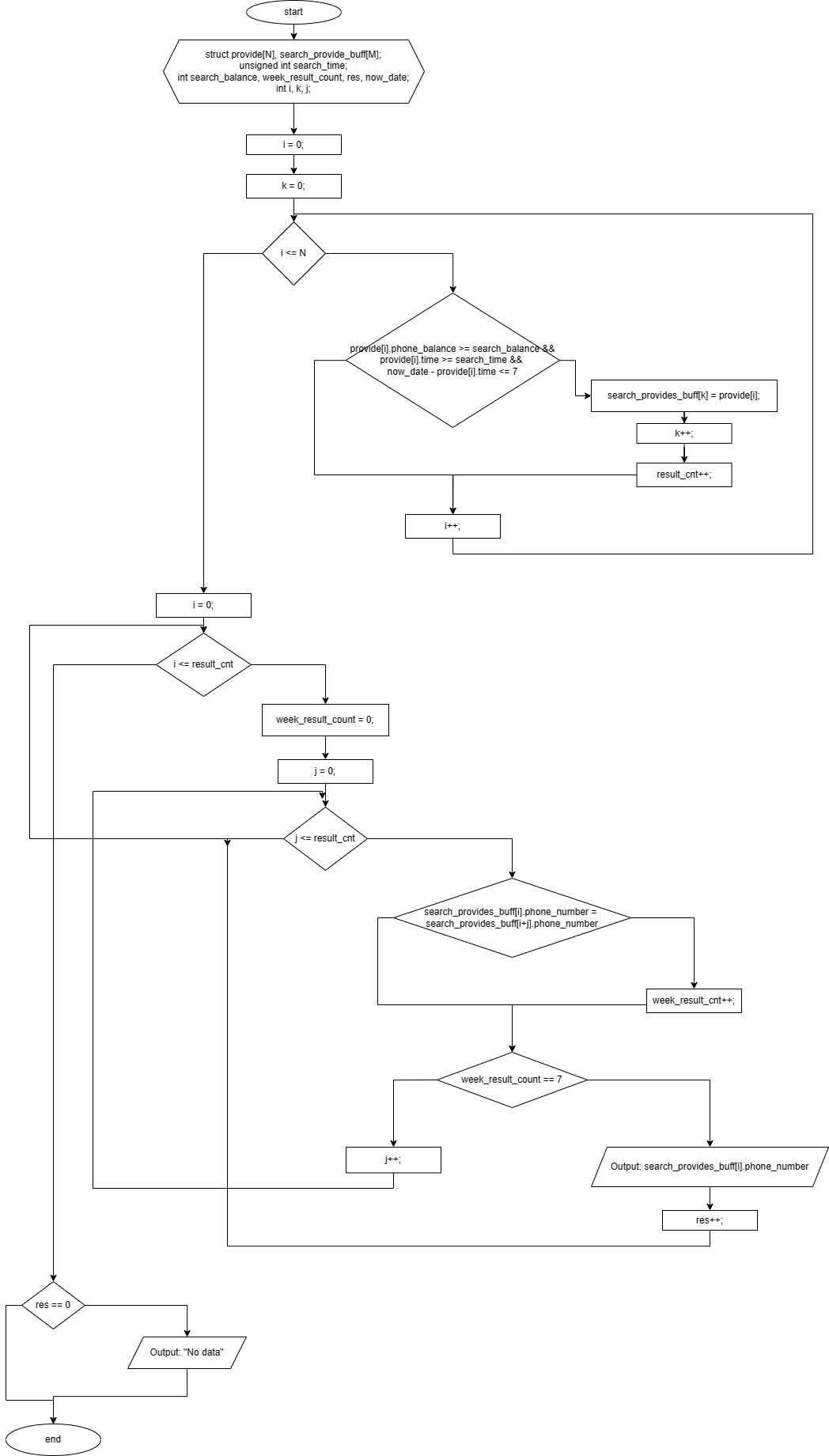


Рисунок 2 – Блок-схема алгоритма поиска

## Написание программы

Используемые библиотеки:

#include <iostream> - получение функций ввода-вывода.

#include <time.h> - работа с временем, получение текущей даты.

#include <stdio.h> - получение функций ввода-вывода.

#include <stdlib.h> - содержит функции для преобразования чисел в текст, выделения памяти и пр.

#include <string.h> - работа со строками.

1. Объявление общих используемых переменных в функции main().

Для того, чтобы упорядочено хранить считываемые данные, используется структура complete\_provides, хранящая переменные:

char phone\_number[16] - номер телефона.

unsigned int code - код услуги. Не может быть отрицательным.

int date\_day, date\_mon, date\_year - дата: день, мес, год. В отдельных переменных для доступа к отдельному сравнению дней.

unsigned int time - время услуги в минутах. Не может быть отрицательным.

float phone\_balance - баланс номера.

Для хранения списка всех данных, используется массив структур complete\_provides\* complete\_provides\_list, динамично выделяющий память, не более MAX\_LIST\_SIZE (50) элементов.

Также, используются переменные:

unsigned int complete\_provides\_cnt - кол-во занесенных записей в массив структур. Используется как длина массива. Не может быть отрицательной.

float search\_balance - значение баланса для поиска по списку.

unsigned int search\_time - значение для поиска, время используемой услуги в минутах. Не может быть отрицательной.

1. Функции, используемые в программе.

Программа разделена на 4 функции, не включая главную - main(), для удобства и улучшения читаемости кода: time\_mir, file\_params\_read, file\_complete\_provides\_read, search. Каждая функция отдельно описана ниже.

Функция сравнения времени с текущей датой - time\_mir:

Возвращает значение int result – итог сравнения. Считывает параметр complete\_provides\* provide – массив данных.

В функции идет проверка на вхождение считанной даты в диапазон между текущей датой и датой неделю назад. Предусмотрена проверка на случай, если неделю назад был прошлый месяц. Если проверка пройдена, возвращается значение result = 1.

Функция считывания параметров из файла param.ini - file\_params\_read:

Не возвращает значения. Считывает параметры int search\_balance и unsigned int search\_time для дальнейшего изменения значений.

Локальные переменные:

FILE\* file\_param – указатель на файл param.ini. Хранит путь до файла и режим открытия (чтение).

char stroke[256], str\_balance[256], str\_time[256] - переменные-буферы: строка файла, хранение строки первого параметра, хранение строки второго параметра.

int data - счетчик строк-параметров, для корректного назначения значений.

Если файл открыт корректно, функция перебирает строки из файла, ищет строки, где прописаны необходимые параметры и сохраняет в соответствующий буфер. Далее из буфера идет форматированное считывание переменных - search\_balance и search\_time.

Функция чтения данных из файла с информацией об услугах и запись в структуры - file\_complete\_provides\_read:

Возвращает значение int id. Считывает параметры complete\_provides\* provide.

Локальные переменные:

FILE\* file - указатель на файл service.txt. Хранит путь до файла и режим открытия (чтение).

unsigned int id – хранит количество считанных записей из файла исходных данных. В дальнейшем используется как длина массива структур исходных значений.

char file\_line[256] – строка-буфер для считывания строки из файла.

char complete\_provide\_date[20] – буфер для считывания части строки с датой для последующей конвертации в числовые значения.

Если файл открыт корректно, функция построчно считывает данные и из полученных строк форматировано заполняет значения переменных структуры. Т.к. есть сложности с чтением даты, полученная строка, хранящая символьную дату, дополнительно форматировано разбивается на силовые составляющие: день, месяц, год. При каждом заполненном элементе структуры шагом в единицу повышается значение переменной id. В конце выполнения функции ее полученное значение возвращается.

Функция поиска по заданным параметрам – search:

Не возвращает значений. Считывает параметры complete\_provides\* provide, unsigned int complete\_provides\_cnt, int search\_balance, unsigned int search\_time.

Локальные переменные:

FILE\* output – указатель на файл report.txt. Хранит путь до файла и режим открытия (запись).

complete\_provides\* search\_provides\_buff – массив-буфер структур, подходящих под заданные параметры.

int week\_result\_count – счетчик количества дней на неделе, когда услугами пользовались с одного и того же номера.

int result\_cnt – счетчик количества записей, выполняющих условия поиска. В дальнейшем используется как длина массива-буфера.

int res – счетчик количества номеров, отобранных при поиске.

Функция сначала перебирает массив исходных данных для отбора элементов по заданным значениям и помещает найденные элементы в массив-буфер.

Затем каждый элемент сравнивается с последующими, чтобы определить, сколько раз на неделе номер телефона использовал услуги. Если набирается счетчик week\_result\_count будет == 7, что значит, услугами пользовались ежедневно, выводится соответствующий номер телефона, счетчик res увеличивается и цикл берет следующий элемент массива за основу для сравнения.

Если после перебора всех элементов не найдено совпадений, выводится соответствующее сообщение.

1. Поэтапные действия в функции main():
   1. Определение всех необходимых переменных.
   2. Вызов функции file\_params\_read.
   3. Заполнение переменной complete\_provides\_cnt с помощью функции file\_complete\_provides\_read.
   4. Вызов функции search.
2. Итог работы программы:

Итогом работы программы является выведенные в консоль настроечные параметры, список считанных из файла исходных данных и, в случае существования, найденные по итогам поиска номера телефонов. Файл output.txt с занесенным в него результатом поиска.

## Пример работы

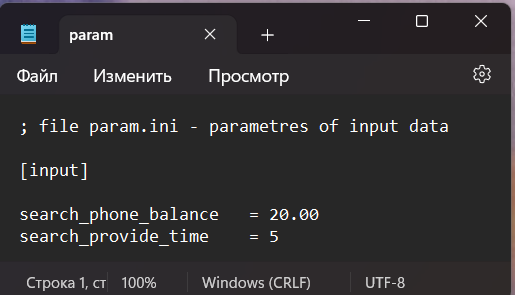


Рисунок 3 – Пример файла .ini

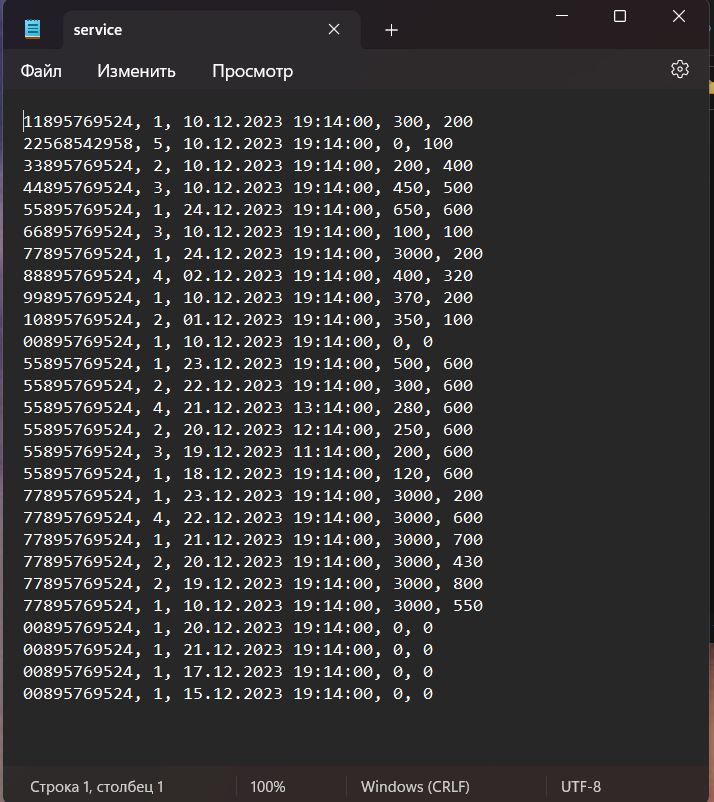


Рисунок 4 – Пример файла service.txt с исходными данными для поиска

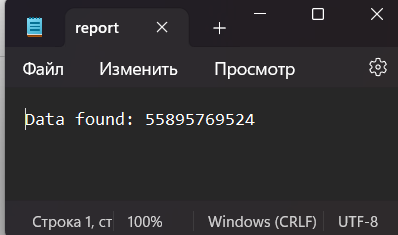


Рисунок 5 – Вывод результата при одном найденном совпадении

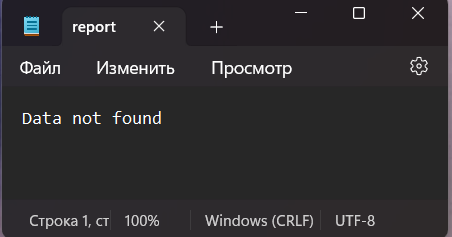


Рисунок 6 – Вывод результата при отсутствии совпадений

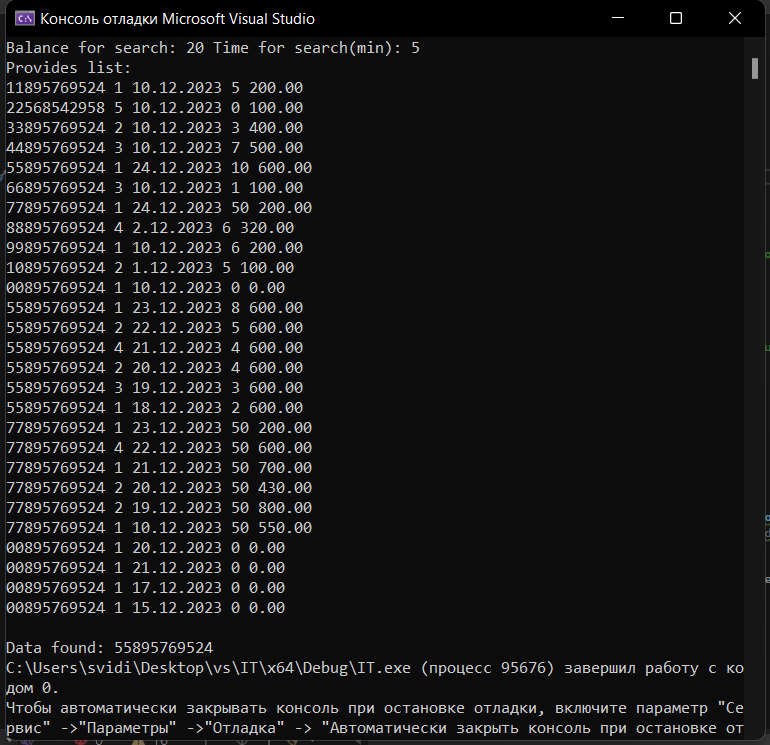


Рисунок 7 – Пример работы программы

# Вывод

Итогом работы стала программа, считывающая исходные данные из двух файлов: с настроечными параметрами и списком проведенных услуг, и выполняющая среди полученного списка данных об услугах поиск по заданным значениям.

Были закреплены знания о работе с файлами, строками, структурами, динамической памятью, временем.

# Приложение А – Листинг

## main.cpp

#include <iostream>

#include <time.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#pragma warning(disable : 4996)

#define MAX\_DATA\_SIZE 50 // Макс размер для строк данных в структуре

#define MAX\_LIST\_SIZE 50 // Макс размер для динамического выделения памяти массивов

#define REPORT\_FN "..\\report.txt"

#define PARAM\_FN "..\\param.ini"

#define SERVICE\_FN "..\\service.txt"

// Хранение данных о Записях Совершенных Услуг

struct complete\_provides {

char phone\_number[16]; // Номер телефона

unsigned int code; // Код услуги

int date\_day, date\_mon, date\_year; // Дата: день, мес, год. В отдельных переменных для доступа к отдельному сравнению дней

unsigned int time; // Время услуги в минутах

float phone\_balance; // Баланс номера

};

// Функция сравнения времени с текущей датой. Работает корректно - наглядная доп проверка в функции file\_complete\_provides\_read

int time\_mir(complete\_provides\* provide) {

// Определение сегодняшней даты

time\_t mytime = time(NULL);

struct tm\* now = localtime(&mytime);

int now\_day = now->tm\_mday, now\_mon = now->tm\_mon + 1, now\_year = now->tm\_year + 1900;

int result = 0; // Доп переменная для определения совпадения

// Проверка - было ли начало недели в прошлом месяце, и входит ли считываемая дата в диапозон недели

if (now\_day >= 8) {

if (provide->date\_day <= now\_day && provide->date\_day >= now\_day - 7) {

result = 1;

}

else result = 0;

}

else if (now\_day < 8) {

if (provide->date\_mon + (now\_day - provide->date\_day) <= 7) {

result = 1;

}

else result = 0;

}

return result;

}

// Функция считывания параметров из файла param.ini

void file\_params\_read(int search\_balance, unsigned int search\_time) {

FILE\* file\_param = fopen(PARAM\_FN, "r");

// Переменные - буферы: строка файла, хранение строки первого параметра, хранение строки второго параметра

char stroke[256], str\_balance[256], str\_time[256];

int data = 1; // Счетчик строк - параметров, для корректного назначения значений

if (file\_param == NULL) perror("Opening param.ini file error");

else {

while (fgets(stroke, sizeof(stroke), file\_param) != NULL) {

// Если строка не является комментарием или заголовком, считывание строк - параметров

if (stroke[0] != ';' && stroke[0] != '[' && stroke[0] == 's') {

if (data == 1) strcpy(str\_balance, stroke);

if (data == 2) strcpy(str\_time, stroke);

data++;

}

}

}

// Заполнение из строк - параметров значений в программные переменные

sscanf(str\_balance, "search\_phone\_balance = %i ", &search\_balance);

sscanf(str\_time, "search\_provide\_time = %u ", &search\_time);

printf("Balance for search: %d Time for search(min): %u\n", search\_balance, search\_time);

fclose(file\_param);

}

// Функция чтения данных из файла с информацией об услугах и запись в структуры. Работает корректно - функция printf выводит уже заполненные структуры из массива

int file\_complete\_provides\_read(complete\_provides\* provide) {

FILE\* file = fopen(SERVICE\_FN, "r");

unsigned int id = 0; // Кол-во считанных данных, в дальнейшем используется как длина списка массива

if (file == NULL) perror("Opening file for reading error");

else {

// Переменные - буферы

char file\_line[256], complete\_provide\_date[20];

puts("Provides list:");

while (fgets(file\_line, sizeof(file\_line), file) != NULL) {

sscanf(file\_line, "%[^,], %u, %[^,], %u, %f ", &provide[id].phone\_number, &provide[id].code, complete\_provide\_date, &provide[id].time, &provide[id].phone\_balance); // Чтение строки из файла

sscanf(complete\_provide\_date, "%u.%u.%u", &provide[id].date\_day, &provide[id].date\_mon, &provide[id].date\_year); // Разделение полученной строки с датой на день, мес, год

provide[id].time = provide[id].time / 60; // Конвертация секунд в минуты

printf("%s %i %i.%i.%i %i %.2f\n", provide[id].phone\_number, provide[id].code, provide[id].date\_day, provide[id].date\_mon, provide[id].date\_year, provide[id].time, provide[id].phone\_balance);

id++;

}

}

fclose(file);

return id;

}

// Функция поиска по заданным параметрам. Работает корректно - выводит номер, если семь совпадений, если хотя бы шесть - нет данных. Доработать - если день одинаковый

void search(complete\_provides\* provide, unsigned int complete\_provides\_cnt, int search\_balance, unsigned int search\_time) {

FILE\* output = fopen(REPORT\_FN, "a");

if (output == NULL) perror("Opening file output.txt for writing error");

complete\_provides\* search\_provides\_buff = (complete\_provides\*)malloc(sizeof(complete\_provides) \* complete\_provides\_cnt); // Буфер для найденных значений

int week\_result\_count = 0, result\_cnt = 0, res = 0; // Кол-во записей в списках результатов

// Если данные подходят под условия, заполняется массив - буфер, подсчет заполненных результатов (в дальнейшем - длина массива-буфера)

for (int i = 0, k = 0; i <= complete\_provides\_cnt; i++) {

if (provide[i].phone\_balance >= search\_balance && provide[i].time >= search\_time && time\_mir(&provide[i]) == 1) {

search\_provides\_buff[k] = provide[i];

k++;

result\_cnt++;

}

}

// Поиск по списку подходящих под условия значений. Если найдено 7 одинаковых номеров за период в неделю - вывод номера

for (int i = 0; i <= result\_cnt; i++) {

week\_result\_count = 0;

for (int j = 0; j <= result\_cnt; j++) {

if ( strcmp(search\_provides\_buff[i].phone\_number, search\_provides\_buff[i+j].phone\_number) == 0) {

week\_result\_count++;

}

if (week\_result\_count == 7) {

printf("\nData found: %s", search\_provides\_buff[i].phone\_number);

fprintf(output, "Data found: %s\n", search\_provides\_buff[i].phone\_number);

res++;

break;

}

}

}

// Если ни один номер не найден, вывод об отсутствии данных

if (res == 0) {

puts("\nData not found");

fputs("Data not found", output);

}

free(search\_provides\_buff);

fclose(output);

}

int main() {

FILE\* output = fopen(REPORT\_FN, "w"); // В функции поиска, при открытии файла используется режим добавления строк, при котором сохраняются записи с прошлого запуска программы

fclose(output); // Для избежания этого, в начале программы файл открывается в режиме перезаписи, чтобы стереть прошлые данные

complete\_provides\* complete\_provides\_list = (complete\_provides\*)malloc(sizeof(complete\_provides) \* MAX\_LIST\_SIZE); // Массив структур с данными об услугах

unsigned int complete\_provides\_cnt = 0; // Кол-во занесенных записей в массив структур

float search\_balance = 150; // Значение баланса для поиска по списку

unsigned int search\_time = 1; // Значение для поиска, время исп услуги в мин

file\_params\_read(search\_balance, search\_time); // Считывание параметров из файла param.ini значений для поиска

complete\_provides\_cnt = file\_complete\_provides\_read(complete\_provides\_list) - 1; // Считывание данных об услугах из файла info.txt в массив структур

search(complete\_provides\_list, complete\_provides\_cnt, search\_balance, search\_time); // Вызов функции поиска по массиву записей с считанными параметрами

free(complete\_provides\_list); // Освобождение памяти занимаемой массивом

return 0;

}