

Отчёт по практикуму по циклу дисциплин «Информатика».

Задание №6.

Выполнил студент группы М8О-111Б-23: Тимофеева Ирина Александровна
№ по списку 21.

Работа выполнена: «24» марта 2024 г.

Преподаватель: каф. 806 Никулин Сергей
Петрович

Входной контроль знаний с оценкой: ____

Отчет сдан «24»

марта 2024г.Итоговая

оценка: _____

Подпись преподавателя: ____

1. **Тема:** обработка последовательной файловой структуры на языке Си.

2. **Цель работы:** разработать последовательную структуру данных для представления простейшей базы данных на файлах в СП Си в соответствии с заданным вариантом.

3. **Задание:** Информация о пассажирах аэропорта: фамилия, инициалы, количество вещей, общий вес вещей, пункт назначения, время вылета, наличие пересадок, сведения о детях.

ВАРИАНТ №36: Определить, имеются ли два пассажира, багаж которых совпадает по числу вещей и различается по весу не более чем на р кг.

Оборудование: Оборудование ПЭВМ студента, если использовалось:

Процессор Intel Core i 5-12450 H с ОП 8 Гб НМД 256 Гб.

Монитор 1920 x 1080~144 HzДругие устройства

4. **Программное обеспечение** (лабораторное):

Операционная система семейства Linux, наименование Ubuntu, версия 23.10
интерпретатор

команд GNU Bash версия 5.2.15.

Система программирования C версия _____

Редактор текстов nvim версия 0.7.2

Утилиты операционной системы

bash

Прикладные системы и программы__vim, gcc__

5. Сценарий выполнения работы (план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты либо соображения по тестированию).

Тексты программ:

Первая часть(описание):

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

const char FILE_NAME[] = "passengers.bin";

typedef struct SPassenger
{
    char last_name[256]; //фамилия
    char first_name[256]; //имя
    char patronymic[256]; //отчество
    int number_of_items; //количество вещей
    int weight; //общий вес вещей
    char destination[50]; //пункт назначения
    char time_d[5]; //время вылета
    int transfer; //наличие пересадок
    int child_cnt; //количество детей
} Passenger;

//вывод информации в таблицу
void ListPassengers(const char file_name[])
{
    FILE *fw;
    Passenger passenger;
    fw = fopen(file_name, "rb");
    if (fw != NULL)
    {
        int counter = 1;

        printf("-----\n");
        printf("|№ |Фамилия |Имя |Отчество |Вещей|Общий вес|Пункт\n");
        printf("назначения|Время вылета|Наличие пересадок|Детей|\n");

        printf("-----\n");
```

```

while (!feof(fw))
{
    if (fread(&passenger, 1, sizeof(Passenger), fw) > 0)
    {
        printf("|%3d|%14s|%12s|%15s|%5d|%9d|%16s|%12s|%17s|%5d|\n", counter,
passenger.last_name, passenger.first_name, passenger.patronymic,
passenger.number_of_items, passenger.weight, passenger.destination, passenger.time_d,
passenger.transfer > 0 ? "Yes" : "No", passenger.child_cnt);
        counter++;
    }
}

printf("-----\n");
fclose(fw);
}
else
{
    printf("Файл БД не найден!\n");
}
}

```

//создание записи

```

void FormatPassenger(char* info[], Passenger *pass)
{
    strcpy(pass->first_name, info[3]);
    strcpy(pass->patronymic, info[4]);
    strcpy(pass->last_name, info[2]);
    pass->number_of_items = atoi(info[5]);
    pass->weight = atoi(info[6]);
    strcpy(pass->destination, info[7]);
    strcpy(pass->time_d, info[8]);
    pass->child_cnt = atoi(info[10]);
    if (strcmp(info[9], "0") == 0 || strcmp(info[9], "нет") == 0 || strcmp(info[9], "no") == 0)
    {
        pass->transfer = 0;
    }
    else
    {
        pass->transfer = 1;
    }
}

```

//создание структуры из массива строк, 1-й и 2-й элементы пропускаю, т.к. зарезервированы под название программы и 1-й параметр

```

Passenger CreatePassenger(char info[11][256])
{
    Passenger tmp;
    strcpy(tmp.first_name, info[3]);
    strcpy(tmp.patronymic, info[4]);
    strcpy(tmp.last_name, info[2]);
    tmp.number_of_items = atoi(info[5]);
    tmp.weight = atoi(info[6]);
    strcpy(tmp.destination, info[7]);
    strcpy(tmp.time_d, info[8]);
    if (strcmp(info[9], "0") == 0 || strcmp(info[9], "нет") == 0 || strcmp(info[9], "no") == 0)
    {
        tmp.transfer = 0;
    }
    else
    {
        tmp.transfer = 1;
    }
    tmp.child_cnt = atoi(info[10]);
    return tmp;
}

```

//количество записей в БД

```

int CountPassengers(const char file_name[])
{
    FILE *fr;
    Passenger passenger;
    fr = fopen(file_name, "rb");
    int counter = 0;
    if (fr != NULL)
    {
        while (!feof(fr))
        {
            if (fread(&passenger, 1, sizeof(Passenger), fr) > 0)
            {
                counter++;
            }
        }
        fclose(fr);
    }
    return counter;
}

```

//добавление в файл записи о пассажире

```

int AddPassenger(const char file_name[], Passenger pass)

```

```

{
    FILE *fw;
    fw = fopen(file_name, "a+b");
    int counter = 0;
    if (fw != NULL)
    {
        fwrite(&pass, 1, sizeof(pass), fw);
        fclose(fw);
        counter++;
    }
    return counter;
}

```

//вывод информации о пассажире
void ViewPassenger(Passenger pass)

```

{
    printf("Информация о пассажире\n");
    printf("Имя - %s\n", pass.first_name);
    printf("Отчество - %s\n", pass.patronymic);
    printf("Фамилия - %s\n", pass.last_name);
    printf("Число вещей - %d\n", pass.number_of_items);
    printf("Вес - %d\n", pass.weight);
    printf("Пункт назначения - %s\n", pass.destination);
    printf("Время вылета - %s\n", pass.time_d);
    printf("Наличие пересадок - %s\n", pass.transfer == 0 ? "Да" : "Нет");
    printf("Количество детей - %d\n", pass.child_cnt);

```

```

printf("-----\n");
}

```

//загрузка информации из файла

int ImportPassengers(char from_file_name[], const char to_file_name[])

```

{
    FILE *f;
    f = fopen(from_file_name, "r");
    char str[11][256];
    Passenger passenger;
    int counter = 0;
    if (f != NULL)
    {
        while (!feof(f))
        {
            if (fscanf(f, "%s\n%s\n%s\n%s\n%s\n%s\n%s\n%s\n%s\n", str[2], str[3], str[4],
str[5], str[6], str[7], str[8], str[9], str[10]) > 0)

```

```

    {
        passenger = CreatePassenger(str);
        AddPassenger(to_file_name, passenger);
        counter++;
    }
}
fclose(f);
}
return counter;
}

```

//удаление записи под номером pos

```
int DeletePassenger(const char file_name[], int pos)
```

```

{
    char* TEMPFILE = "temp.bin";
    Passenger passenger;
    FILE *fr;
    int res = 1;
    if(pos > 0)
    {
        fr = fopen(file_name, "rb");
        int counter = 1;
        if (fr != NULL)
        {
            while (!feof(fr))
            {
                if (fread(&passenger, 1, sizeof(Passenger), fr) > 0 && counter != pos)
                {
                    AddPassenger(TEMPFILE, passenger);
                }
                counter++;
            }
        }
        fclose(fr);
        remove(file_name);
        res = rename(TEMPFILE, file_name);
        if (pos > counter) res = 1;
    }
    return res;
}

```

//вывод информации в таблицу

```
void ViewPassengers(Passenger passengers[], int size)
```

```

{
    if (size > 0)

```

```

{

printf("-----\n");
printf("№ |Фамилия |Имя |Отчество |Вещей|Общий вес|Пункт
назначения|Время вылета|Наличие пересадок|Детей\n");

printf("-----\n");
for (int i = 0; i < size; i++)
{
printf("|%3d|%14s|%12s|%15s|%5d|%9d|%16s|%12s|%17s|%5d\n", i+1,
passengers[i].last_name, passengers[i].first_name, passengers[i].patronymic,
passengers[i].number_of_items, passengers[i].weight, passengers[i].destination,
passengers[i].time_d, passengers[i].transfer > 0 ? "Yes" : "No", passengers[i].child_cnt);

}

printf("-----\n");
}

//вывод информации о двух пассажирах в таблицу
void View2Passengers(Passenger p1, Passenger p2)
{
if (&p1 != NULL && &p2 != NULL)
{
printf("найжены!\n");

printf("-----\n");
printf("№ |Фамилия |Имя |Отчество |Вещей|Общий вес|Пункт
назначения|Время вылета|Наличие пересадок|Детей\n");

printf("-----\n");
printf("|%3d|%14s|%12s|%15s|%5d|%9d|%16s|%12s|%17s|%5d\n", 1, p1.last_name,
p1.first_name, p1.patronymic, p1.number_of_items, p1.weight, p1.destination, p1.time_d,
p1.transfer > 0 ? "Yes" : "No", p1.child_cnt);
printf("|%3d|%14s|%12s|%15s|%5d|%9d|%16s|%12s|%17s|%5d\n", 2, p2.last_name,
p2.first_name, p2.patronymic, p2.number_of_items, p2.weight, p2.destination, p2.time_d,
p2.transfer > 0 ? "Yes" : "No", p2.child_cnt);
}
}

```

```
printf("-----\n");
}
}
```

//поиск в файле 2 пассажиров из общего количества kol с одинаковым числом вещей и разницей в весе не более p кг

```
int FindP(const char file_name[], int p, int kol)
{
```

```
    FILE *fr;
    fr = fopen(file_name, "rb");
    Passenger p1, p2;
```

```
    //индикатор поиска - в случае успеха будет 1
```

```
    int find = 0;
```

```
    if (fr != NULL)
```

```
    {
```

```
        //проверим список пассажиров начиная с 1-го
```

```
        //сравниваем его по критерию со всеми остальными
```

```
        //если не нашли - переходим ко 2-му и повторяем поиск
```

```
        for (int position=0; position<kol; position++)
```

```
        {
```

```
            int ind = position;
```

```
            // перемещаем указатель в файле на позицию position
```

```
            fseek(fr, position*sizeof(Passenger), SEEK_SET);
```

```
            //в цикле проверим список пассажиров
```

```
            while (!feof(fr))
```

```
            {
```

```
                if (fread(&p2, 1, sizeof(Passenger), fr) > 0)
```

```
                {
```

```
                    if (ind == position)
```

```
                    {
```

```
                        p1 = p2;
```

```
                    }
```

```
                else if (p1.number_of_items == p2.number_of_items && abs(p1.weight - p2.weight)<=p)
```

```
                {
```

```
                    //если критерий поиска сработал - выведем информацию об этих 2 пассажирах и закончим поиск
```

```
                    View2Passengers(p1, p2);
```

```
                    find = 1;
```

```
                    break;
```



```

        }
        ind++;
    }
}
if (find == 1) break;
}

fclose(fr);
}
else
    printf("Файл БД не найден!\n");

return find;
}

```

Вторая часть:

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <locale.h>
#include "passenger.h"

int main(int argc, char* argv[])
{
    char *locale = setlocale(LC_ALL, "Russian_Russia.1251");

    Passenger passenger;

    //проверим параметры (аргументы) запуска программы
    if (argc < 2)
    {
        printf("Запустите программу с ключом ? для справки\n");
    }
    else
    {
        if (!strcmp(argv[1], "?"))
        {
            printf("Программа работы с базой данных пассажиров\n");
            printf("Файл базы данных passengers.bin должен быть в папке приложения\n");
            printf("Синтакс: files [-f] [-imp] [Имя_файла] [-del] [n] [all] [-add] [Фамилия Имя  
Отчество Кол-во_вещей Общй_вес_вещей Пункт_назначения Время_вылета  
Наличие_пересадок Сведения_о_детях]\n");
            printf("-f - вывести содержимое базы данных пассажиров\n");

```

```

printf("-add - добавить запись о пассажире в виде\n[Фамилия Имя Отчество
Кол-во_вещей Общий_вес_вещей Пункт_назначения Время_вылета
Наличие_пересадок Сведения_о_детях]\n");
printf("-imp - добавить информацию из текстового файла [Имя_файла] с
данными в виде\n[Фамилия Имя Отчество Кол-во_вещей Общий_вес_вещей
Пункт_назначения Время_вылета Наличие_пересадок Сведения_о_детях]\n");
printf("-del - удалить информацию о пассажире с номером [n] или всех
пассажиров [all]\n");
}
if (!strcmp(argv[1], "-add"))
{
    if (argc < 11)
    {
        printf("Недостаточное число параметров для добавления записи в БД!\n");
    }
    else
    {
        //passenger = CreatePassenger(argv);
        FormatPassenger(argv, &passenger);
        int kol = AddPassenger(FILE_NAME, passenger);
        if (kol > 0)
            printf("Добавлена запись!\n");
        else
            printf("Ошибка добавления!\n");
    }
}
if (!strcmp(argv[1], "-imp"))
{
    if (argc < 3)
    {
        printf("Недостаточное число параметров для импорта записей в БД!\n");
    }
    else
    {
        int kol = ImportPassengers(argv[2], FILE_NAME);
        if (kol > 0)
        {
            printf("Добавлены %d записей\n", kol);
        }
        else
        {
            printf("Не удалось записать информацию в файл БД\n");
        }
    }
}
}

```

```

if (!strcmp(argv[1], "-del"))
{
    if (argc < 2)
    {
        printf("Недостаточное число параметров для удаления записи в БД!\n");
    }
    else
    {
        if (argc == 3)
        {
            if (!strcmp(argv[2], "all"))
            {
                remove(FILE_NAME);
                printf("Файл БД удалён!\n");
            }
            else
            {
                int pos = atoi(argv[2]);
                int res = DeletePassenger(FILE_NAME, pos);
                if (res == 0)
                {
                    printf("Выполнено удаление записи %d\n", pos);
                }
                else
                {
                    printf("Ошибка удаления!\n");
                }
            }
        }
        else
        {
            ListPassengers(FILE_NAME);
            int pos = 10000;
            while (pos != 0)
            {
                printf("Введите номер записи для удаления (0-выход): ");
                scanf("%d", &pos);
                int res = DeletePassenger(FILE_NAME, pos);
                if (res == 0)
                {
                    printf("Выполнено удаление записи %d\n", pos);
                    ListPassengers(FILE_NAME);
                }
                else
                {
                    printf("Ошибка удаления!\n");
                }
            }
        }
    }
}

```

```

    }
}
if (!strcmp(argv[1], "-f"))
{
    ListPassengers(FILE_NAME);
}
}
return 0;
}

```

Третья часть:

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <locale.h>
#include "passenger.h"

int main(int argc, char* argv[])
{
    char *locale = setlocale(LC_ALL, "Russian_Russia.1251");

    Passenger passenger;

    //проверим параметры (аргументы) запуска программы
    if (argc < 2)
    {
        printf("Запустите программу с ключом ? для справки\n");
    }
    else
    {
        if (!strcmp(argv[1], "?"))
        {
            printf("Программа работы с базой данных пассажиров\n");
            printf("Файл базы данных passengers.bin должен быть в папке приложения\n");
            printf("Синтакс: files [-f] [-p] [w]\n");
            printf("-f - вывести содержимое базы данных пассажиров\n");
            printf("-p - найти пассажиров, у которых багаж совпадает по числу вещей и  
различается не более чем на [w] кг\n");
        }
        if (!strcmp(argv[1], "-f"))
        {
            ListPassengers(FILE_NAME);
        }
        if (!strcmp(argv[1], "-p"))
    }

```

```

{
    if (argc < 3)
    {
        printf("Недостаточное число параметров для поиска в БД!\n");
    }
    else
    {
        int p = atoi(argv[2]);
        int kol = CountPassengers(FILE_NAME);
        printf("Пассажиры, у которых багаж совпадает по числу вещей и различается
не более чем на %d кг: ", p);
        if (!FindP(FILE_NAME, p, kol)) printf("не найдены!\n");
    }
}
}
return 0;
}

```

6. Распечатка протокола (подклеить листинг окончательного варианта программы с тестовыми примерами, подписанный преподавателем).

#текст программы работы с БД пассажиров

irina@Irina-VivoBook:~/Prog/Prog_C/Lab22/Filesdb\$ cat filesdb.c

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
#include <locale.h>
```

```
#include "passenger.h"
```

```
int main(int argc, char* argv[])
```

```
{
```

```
    char *locale = setlocale(LC_ALL, "Russian_Russia.1251");
```

```
    Passenger passenger;
```

```
    //проверим параметры (аргументы) запуска программы
```

```
    if (argc < 2)
```

```
    {
```

```
        printf("Запустите программу с ключом ? для справки\n");
```

```
    }
```

```
    else
```

```
    {
```

```
        if (!strcmp(argv[1], "?"))
```

```
        {
```

```
            printf("Программа работы с базой данных пассажиров\n");
```

```

    printf("Файл базы данных passengers.bin должен быть в папке приложения\n");
    printf("Синтакс: files [-f] [-imp] [Имя_файла] [-del] [n] [all] [-add] [Фамилия Имя
Отчество Кол-во_вещей Общий_вес_вещей Пункт_назначения Время_вылета
Наличие_пересадок Сведения_о_детях]\n");
    printf("-f - вывести содержимое базы данных пассажиров\n");
    printf("-add - добавить запись о пассажире в виде\n[Фамилия Имя Отчество
Кол-во_вещей Общий_вес_вещей Пункт_назначения Время_вылета
Наличие_пересадок Сведения_о_детях]\n");
    printf("-imp - добавить информацию из текстового файла [Имя_файла] с
данными в виде\n[Фамилия Имя Отчество Кол-во_вещей Общий_вес_вещей
Пункт_назначения Время_вылета Наличие_пересадок Сведения_о_детях]\n");
    printf("-del - удалить информацию о пассажире с номером [n] или всех
пассажиров [all]\n");
}
if (!strcmp(argv[1], "-add"))
{
    if (argc < 11)
    {
        printf("Недостаточное число параметров для добавления записи в БД!\n");
    }
    else
    {
        //passenger = CreatePassenger(argv);
        FormatPassenger(argv, &passenger);
        int kol = AddPassenger(FILE_NAME, passenger);
        if (kol > 0)
            printf("Добавлена запись!\n");
        else
            printf("Ошибка добавления!\n");
    }
}
if (!strcmp(argv[1], "-imp"))
{
    if (argc < 3)
    {
        printf("Недостаточное число параметров для импорта записей в БД!\n");
    }
    else
    {
        int kol = ImportPassengers(argv[2], FILE_NAME);
        if (kol > 0)
        {
            printf("Добавлены %d записей\n", kol);
        }
        else

```

```

        {
            printf("Не удалось записать информацию в файл БД\n");
        }
    }
}
if (!strcmp(argv[1], "-del"))
{
    if (argc < 2)
    {
        printf("Недостаточное число параметров для удаления записи в БД!\n");
    }
    else
    {
        if (argc == 3)
        {
            if (!strcmp(argv[2], "all"))
            {
                remove(FILE_NAME);
                printf("Файл БД удалён!\n");
            }
            else
            {
                int pos = atoi(argv[2]);
                int res = DeletePassenger(FILE_NAME, pos);
                if (res == 0)
                {
                    printf("Выполнено удаление записи %d\n", pos);
                }
                else
                {
                    printf("Ошибка удаления!\n");
                }
            }
        }
        else
        {
            ListPassengers(FILE_NAME);
            int pos = 10000;
            while (pos != 0)
            {
                printf("Введите номер записи для удаления (0-выход): ");
                scanf("%d", &pos);
                int res = DeletePassenger(FILE_NAME, pos);
                if (res == 0)
                {
                    printf("Выполнено удаление записи %d\n", pos);
                    ListPassengers(FILE_NAME);
                }
            }
        }
    }
}

```

```

        }
        else
            printf("Ошибка удаления!\n");
    }
}
}
}
if (!strcmp(argv[1], "-f"))
{
    ListPassengers(FILE_NAME);
}
}
return 0;
}

```

#текст программы поиска пассажиров по критерию

irina@Irina-VivoBook:~/Prog/Prog_C/Lab22/Filesdb\$ cat findp.c

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
#include <locale.h>
```

```
#include "passenger.h"
```

```
int main(int argc, char* argv[])
```

```
{
```

```
    char *locale = setlocale(LC_ALL, "Russian_Russia.1251");
```

```
    Passenger passenger;
```

```
    //проверим параметры (аргументы) запуска программы
```

```
    if (argc < 2)
```

```
    {
```

```
        printf("Запустите программу с ключом ? для справки\n");
```

```
    }
```

```
    else
```

```
    {
```

```
        if (!strcmp(argv[1], "?"))
```

```
        {
```

```
            printf("Программа работы с базой данных пассажиров\n");
```

```
            printf("Файл базы данных passengers.bin должен быть в папке приложения\n");
```

```
            printf("Синтакс: files [-f] [-p] [w]\n");
```

```
            printf("-f - вывести содержимое базы данных пассажиров\n");
```

```
            printf("-p - найти пассажиров, у которых багаж совпадает по числу вещей и  
различается не более чем на [w] кг\n");
```

```
        }
```



```

if (!strcmp(argv[1], "-f"))
{
    ListPassengers(FILE_NAME);
}
if (!strcmp(argv[1], "-p"))
{
    if (argc < 3)
    {
        printf("Недостаточное число параметров для поиска в БД!\n");
    }
    else
    {
        int p = atoi(argv[2]);
        int kol = CountPassengers(FILE_NAME);
        printf("Пассажиры, у которых багаж совпадает по числу вещей и различается
не более чем на %d кг: ", p);
        if (!FindP(FILE_NAME, p, kol)) printf("не найдены!\n");
    }
}
}
return 0;
}

```

#текст библиотеки общих функций программ
irina@Irina-VivoBook:~/Prog/Prog_C/Lab22/Filesdb\$ cat passenger.h

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>

```

```

const char FILE_NAME[] = "passengers.bin";

```

```

typedef struct SPassenger
{
    char last_name[256]; //фамилия
    char first_name[256]; //имя
    char patronymic[256]; //отчество
    int number_of_items; //количество вещей
    int weight; //общий вес вещей
    char destination[50]; //пункт назначения
    char time_d[5]; //время вылета
    int transfer; //наличие пересадок
    int child_cnt; //количество детей
} Passenger;

```

```

//вывод информации в таблицу

```

```

void ListPassengers(const char file_name[])
{
    FILE *fw;
    Passenger passenger;
    fw = fopen(file_name, "rb");
    if (fw != NULL)
    {
        int counter = 1;

        printf("-----\n");
        printf("|№ |Фамилия |Имя |Отчество |Вещей|Общий вес|Пункт  

назначения|Время вылета|Наличие пересадок|Детей\n");

        printf("-----\n");
        while (!feof(fw))
        {
            if (fread(&passenger, 1, sizeof(Passenger), fw) > 0)
            {
                printf("|%3d|%14s|%12s|%15s|%5d|%9d|%16s|%12s|%17s|%5d\n", counter,
passenger.last_name, passenger.first_name, passenger.patronymic,
passenger.number_of_items, passenger.weight, passenger.destination, passenger.time_d,
passenger.transfer > 0 ? "Yes" : "No", passenger.child_cnt);
                counter++;
            }
        }

        printf("-----\n");
        fclose(fw);
    }
    else
    {
        printf("Файл БД не найден!\n");
    }
}

//создание записи
void FormatPassenger(char* info[], Passenger *pass)
{
    strcpy(pass->first_name, info[3]);
    strcpy(pass->patronymic, info[4]);
    strcpy(pass->last_name, info[2]);
    pass->number_of_items = atoi(info[5]);
}

```

```

pass->weight = atoi(info[6]);
strcpy(pass->destination, info[7]);
strcpy(pass->time_d, info[8]);
pass->child_cnt = atoi(info[10]);
if (strcmp(info[9], "0") == 0 || strcmp(info[9], "нет") == 0 || strcmp(info[9], "no") == 0)
{
    pass->transfer = 0;
}
else
{
    pass->transfer = 1;
}
}

```

//создание структуры из массива строк, 1-й и 2-й элементы пропускам, т.к.
зарезервированы под название программы и 1-й параметр

```

Passenger CreatePassenger(char info[11][256])
{
    Passenger tmp;
    strcpy(tmp.first_name, info[3]);
    strcpy(tmp.patronymic, info[4]);
    strcpy(tmp.last_name, info[2]);
    tmp.number_of_items = atoi(info[5]);
    tmp.weight = atoi(info[6]);
    strcpy(tmp.destination, info[7]);
    strcpy(tmp.time_d, info[8]);
    if (strcmp(info[9], "0") == 0 || strcmp(info[9], "нет") == 0 || strcmp(info[9], "no") == 0)
    {
        tmp.transfer = 0;
    }
    else
    {
        tmp.transfer = 1;
    }
    tmp.child_cnt = atoi(info[10]);
    return tmp;
}

```

//количество записей в БД

```

int CountPassengers(const char file_name[])
{
    FILE *fr;
    Passenger passenger;
    fr = fopen(file_name, "rb");
    int counter = 0;

```

```

if (fr != NULL)
{
    while (!feof(fr))
    {
        if (fread(&passenger, 1, sizeof(Passenger), fr) > 0)
        {
            counter++;
        }
    }
    fclose(fr);
}
return counter;
}

```

//добавление в файл записи о пассажире

```

int AddPassenger(const char file_name[], Passenger pass)
{
    FILE *fw;
    fw = fopen(file_name, "a+b");
    int counter = 0;
    if (fw != NULL)
    {
        fwrite(&pass, 1, sizeof(pass), fw);
        fclose(fw);
        counter++;
    }
    return counter;
}

```

//вывод информации о пассажире

```

void ViewPassenger(Passenger pass)
{
    printf("Информация о пассажире\n");
    printf("Имя - %s\n", pass.first_name);
    printf("Отчество - %s\n", pass.patronymic);
    printf("Фамилия - %s\n", pass.last_name);
    printf("Число вещей - %d\n", pass.number_of_items);
    printf("Вес - %d\n", pass.weight);
    printf("Пункт назначения - %s\n", pass.destination);
    printf("Время вылета - %s\n", pass.time_d);
    printf("Наличие пересадок - %s\n", pass.transfer == 0 ? "Да" : "Нет");
    printf("Количество детей - %d\n", pass.child_cnt);
}

```

```

printf("-----\n");

```

```
}
```

```
//загрузка информации из файла
```

```
int ImportPassengers(char from_file_name[], const char to_file_name[])
```

```
{
```

```
    FILE *f;
```

```
    f = fopen(from_file_name, "r");
```

```
    char str[11][256];
```

```
    Passenger passenger;
```

```
    int counter = 0;
```

```
    if (f != NULL)
```

```
    {
```

```
        while (!feof(f))
```

```
        {
```

```
            if (fscanf(f, "%s\n%s\n%s\n%s\n%s\n%s\n%s\n%s\n%s\n", str[2], str[3], str[4],  
str[5], str[6], str[7], str[8], str[9], str[10]) > 0)
```

```
            {
```

```
                passenger = CreatePassenger(str);
```

```
                AddPassenger(to_file_name, passenger);
```

```
                counter++;
```

```
            }
```

```
        }
```

```
        fclose(f);
```

```
    }
```

```
    return counter;
```

```
}
```

```
//удаление записи под номером pos
```

```
int DeletePassenger(const char file_name[], int pos)
```

```
{
```

```
    char* TEMPFILE = "temp.bin";
```

```
    Passenger passenger;
```

```
    FILE *fr;
```

```
    int res = 1;
```

```
    if(pos > 0)
```

```
    {
```

```
        fr = fopen(file_name, "rb");
```

```
        int counter = 1;
```

```
        if (fr != NULL)
```

```
        {
```

```
            while (!feof(fr))
```

```
            {
```

```
                if (fread(&passenger, 1, sizeof(Passenger), fr) > 0 && counter != pos)
```

```
                {
```

```
                    AddPassenger(TEMPFILE, passenger);
```

```

        }
        counter++;
    }
}
fclose(fr);
remove(file_name);
res = rename(TEMPFILE, file_name);
if (pos > counter) res = 1;
}
return res;
}

```

//вывод информации в таблицу

```
void ViewPassengers(Passenger passengers[], int size)
```

```

{
    if (size > 0)
    {

```

```

printf("-----\n");

```

```

        printf("|№ |Фамилия |Имя |Отчество |Вещей|Общий вес|Пункт
назначения|Время вылета|Наличие пересадок|Детей|\n");

```

```

printf("-----\n");

```

```

        for (int i = 0; i < size; i++)
        {
            printf("|%3d|%14s|%12s|%15s|%5d|%9d|%16s|%12s|%17s|%5d|\n", i+1,
passengers[i].last_name, passengers[i].first_name, passengers[i].patronymic,
passengers[i].number_of_items, passengers[i].weight, passengers[i].destination,
passengers[i].time_d, passengers[i].transfer > 0 ? "Yes" : "No", passengers[i].child_cnt);

```

```

        }

```

```

printf("-----\n");

```

```

    }
}

```

//вывод информации о двух пассажирах в таблицу

```
void View2Passengers(Passenger p1, Passenger p2)
```

```

{
    if (&p1 != NULL && &p2 != NULL)
    {
        printf("найжены!\n");

```

```

printf("-----\n");
printf("|№ |Фамилия |Имя |Отчество |Вещей|Общий вес|Пункт
назначения|Время вылета|Наличие пересадок|Детей|\n");

printf("-----\n");
printf("|%3d|%14s|%12s|%15s|%5d|%9d|%16s|%12s|%17s|%5d|\n", 1, p1.last_name,
p1.first_name, p1.patronymic, p1.number_of_items, p1.weight, p1.destination, p1.time_d,
p1.transfer > 0 ? "Yes" : "No", p1.child_cnt);
printf("|%3d|%14s|%12s|%15s|%5d|%9d|%16s|%12s|%17s|%5d|\n", 2, p2.last_name,
p2.first_name, p2.patronymic, p2.number_of_items, p2.weight, p2.destination, p2.time_d,
p2.transfer > 0 ? "Yes" : "No", p2.child_cnt);

printf("-----\n");
}
}

```

//поиск в файле 2 пассажиров из общего количества kol с одинаковым числом вещей и разницей в весе не более p кг

```
int FindP(const char file_name[], int p, int kol)
```

```

{
    FILE *fr;
    fr = fopen(file_name, "rb");
    Passenger p1, p2;

    //индикатор поиска - в случае успеха будет 1
    int find = 0;
    if (fr != NULL)
    {
        //проверим список пассажиров начиная с 1-го
        //сравниваем его по критерию со всеми остальными
        //если не нашли - переходим ко 2-му и повторяем поиск
        for (int position=0; position<kol; position++)
        {
            int ind = position;

            // перемещаем указатель в файле на позицию position
            fseek(fr, position*sizeof(Passenger), SEEK_SET);

            //в цикле проверим список пассажиров
            while (!feof(fr))
            {

```

```

        if (fread(&p2, 1, sizeof(Passenger), fr) > 0)
        {
            if (ind == position)
            {
                p1 = p2;
            }
            else if (p1.number_of_items == p2.number_of_items && abs(p1.weight -
p2.weight)<=p)
            {
                //если критерий поиска сработал - выведем информацию об этих 2
пассажирах и закончим поиск
                View2Passengers(p1, p2);
                find = 1;
                break;
            }
            ind++;
        }
    }
    if (find == 1) break;
}

fclose(fr);
}
else
    printf("Файл БД не найден!\n");

return find;
}

```

#компиляция и запуск программы создания БД пассажиров

irina@Irina-VivoBook:~/Prog/Prog_C/Lab22/Filesdb\$ gcc -o filesdb filesdb.c

irina@Irina-VivoBook:~/Prog/Prog_C/Lab22/Filesdb\$./filesdb ?

Программа работы с базой данных пассажиров

Файл базы данных passengers.bin должен быть в папке приложения

Синтакс: files [-f] [-imp] [Имя_файла] [-del] [n] [all] [-add] [Фамилия Имя Отчество

Кол-во_вещей Общий_вес_вещей Пункт_назначения Время_вылета

Наличие_пересадок Сведения_о_детях]

-f - вывести содержимое базы данных пассажиров

-add - добавить запись о пассажире в виде

[Фамилия Имя Отчество Кол-во_вещей Общий_вес_вещей Пункт_назначения
Время_вылета Наличие_пересадок Сведения_о_детях]

-imp - добавить информацию из текстового файла [Имя_файла] с данными в виде

[Фамилия Имя Отчество Кол-во_вещей Общий_вес_вещей Пункт_назначения
Время_вылета Наличие_пересадок Сведения_о_детях]

-del - удалить информацию о пассажире с номером [n] или всех пассарижор [all]

#добавление пассажира

irina@Irina-VivoBook:~/Prog/Prog_C/Lab22/Filesdb\$./filesdb -add Sayko Nikolay Leonidovich 5 69 Troick 23:12 no 1

Добавлена запись!

irina@Irina-VivoBook:~/Prog/Prog_C/Lab22/Filesdb\$./filesdb -f

```
-----
|№ |Фамилия |Имя |Отчество |Вещей|Общий вес|Пункт назначения|Время
вылета|Наличие пересадок|Детей|
-----
| 1| Volkov| Afanasiy| Yurievich| 4| 46| Vologda| 19:35| No|
3|
| 2| Osipova| Yulia| Olegovna| 3| 38| Sankt-Peterburg| 17:10|
No| 1|
| 3| Sayko| Alexander| Nikolaevich| 2| 34| Vladivostok| 18:30|
Yes| 0|
| 4| Golubev| Maxim| Borisovich| 1| 21| Samara| 10:12|
No| 1|
| 5| Korotkov| Dmitry| Vladimirovich| 2| 29| Ufa| 22:19| No|
3|
| 6| Konoplev| Boris| Alexeevich| 3| 44| Perm| 04:45| Yes|
2|
| 7| Novoselov| Nikolay| Ivanovich| 3| 51| Salehard| 10:18|
No| 3|
| 8| Kovalev| Konstantin| Nikolaevich| 4| 55| Ekaterinburg| 15:50|
Yes| 0|
| 9| Bilyushova| Kristina| Alexandrovna| 4| 37| Krasnoyarsk| 20:15|
No| 1|
| 10| Berezhnoy| Alexander| Pavlovich| 3| 34| Kenerovo| 17:15|
Yes| 2|
-----
```

#удаление пассажиров

irina@Irina-VivoBook:~/Prog/Prog_C/Lab22/Filesdb\$./filesdb -del 10

Выполнено удаление записи 10

irina@Irina-VivoBook:~/Prog/Prog_C/Lab22/Filesdb\$./filesdb -del 7

Выполнено удаление записи 7

irina@Irina-VivoBook:~/Prog/Prog_C/Lab22/Filesdb\$./filesdb -del 5

Выполнено удаление записи 5

irina@Irina-VivoBook:~/Prog/Prog_C/Lab22/Filesdb\$./filesdb -del 4

Выполнено удаление записи 4

irina@Irina-VivoBook:~/Prog/Prog_C/Lab22/Filesdb\$./filesdb -f

```
-----
|№ |Фамилия |Имя |Отчество |Вещей|Общий вес|Пункт назначения|Время
вылета|Наличие пересадок|Детей|
-----
```

```
| 1| Volkov| Afanasiy| Yurievich| 4| 46| Vologda| 19:35| No|
3|
| 2| Osipova| Yulia| Olegovna| 3| 38| Sankt-Peterburg| 17:10|
No| 1|
| 3| Sayko| Alexander| Nikolaevich| 2| 34| Vladivostok| 18:30|
Yes| 0|
| 4| Konoplev| Boris| Alexeevich| 3| 44| Perm| 04:45| Yes|
2|
| 5| Kovalev| Konstantin| Nikolaevich| 4| 55| Ekaterinburg| 15:50|
Yes| 0|
| 6| Bilyushova| Kristina| Alexandrovna| 4| 37| Krasnoyarsk| 20:15|
No| 1|
-----
```

```
#удаление всех записей
```

```
irina@Irina-VivoBook:~/Prog/Prog_C/Lab22/Filesdb$ ./filesdb -del all
```

```
Файл БД удалён!
```

```
irina@Irina-VivoBook:~/Prog/Prog_C/Lab22/Filesdb$ ./filesdb -f
```

```
Файл БД не найден!
```

```
#импорт 24 записей из файла people.imp
```

```
irina@Irina-VivoBook:~/Prog/Prog_C/Lab22/Filesdb$ ./filesdb -imp people.imp
```

```
Добавлены 24 записей
```

```
irina@Irina-VivoBook:~/Prog/Prog_C/Lab22/Filesdb$ ./filesdb -f
```

```
-----
|№ |Фамилия |Имя |Отчество |Вещей|Общий вес|Пункт назначения|Время
вылета|Наличие пересадок|Детей|
-----
```

```
| 1| Volkov| Afanasiy| Yurievich| 4| 46| Vologda| 19:35| No|
3|
| 2| Pegova| Anastasia| Alexandrovna| 3| 17| Tyumen| 20:12|
Yes| 2|
| 3| Osipova| Yulia| Olegovna| 3| 38| Sankt-Peterburg| 17:10|
No| 1|
| 4| Sayko| Alexander| Nikolaevich| 2| 34| Vladivostok| 18:30|
Yes| 0|
```

5	Golubev	Maxim	Borisovich	1	21	Samara	10:12	
No	1							
6	Korotkov	Dmitry	Vladimirovich	2	29	Ufa	22:19	No
3								
7	Konoplev	Boris	Alexeevich	3	44	Perm	04:45	Yes
2								
8	Novoselov	Nikolay	Ivanovich	3	51	Salehard	10:18	
No	3							
9	Kovalev	Konstantin	Nikolaevich	4	55	Ekaterinburg	15:50	
Yes	0							
10	Bilyushova	Kristina	Alexandrovna	4	37	Krasnoyarsk	20:15	
No	1							
11	Berezhnoy	Alexander	Pavlovich	3	34	Kenerovo	17:15	
Yes	2							
12	Ivanov	Petr	Vasilevich	2	24	Moskva	14:18	No
1								
13	Vasilev	Kirill	Igorevich	4	35	Tomsk	21:45	Yes
1								
14	Platoshkin	Nikolay	Nikolayevich	5	46	Irkutsk	19:55	
No	0							
15	Vladova	Alla	Yurievna	2	19	Magadan	23:10	Yes
2								
16	Kachurin	Yuri	Vladimirovich	1	15	Surgut	18:29	No
3								
17	Chikina	Ekaterina	Vasilevna	1	24	Petrozavodsk	16:56	
No	2							
18	Malynko	Sergey	Alexeevich	3	20	Kursk	17:48	No
0								
19	Popov	Yury	Valerievich	2	12	Volgograd	10:10	Yes
2								
20	Novikova	Natalia	Nikolaevna	3	15	Novgorod	14:20	
No	3							
21	Bilan	Dmitry	Gennadievich	2	10	Murmansk	09:25	
No	1							
22	Voronin	Pavel	Ivanovich	3	16	Saratov	12:15	Yes
3								
23	Korolev	Vasiliy	Sidorovich	1	28	Cheboksary	07:19	
No	1							
24	Stoyanov	Yury	Stanislavovich	2	14	Ivanovo	03:40	
No	2							

#компиляция и запуск программы поиска

irina@Irina-VivoBook:~/Prog/Prog_C/Lab22/Filesdb\$ gcc -o findp findp.c

irina@Irina-VivoBook:~/Prog/Prog_C/Lab22/Filesdb\$./findp ?

Программа работы с базой данных пассажиров

Файл базы данных passengers.bin должен быть в папке приложения

Синтакс: files [-f] [-p] [w]

-f - вывести содержимое базы данных пассажиров

-p - найти пассажиров, у которых багаж совпадает по числу вещей и различается не более чем на [w] кг

#вывести список всех пассажиров

irina@Irina-VivoBook:~/Prog/Prog_C/Lab22/Filesdb\$./findp -f

```
-----
|№|Фамилия|Имя|Отчество|Вещей|Общий вес|Пункт назначения|Время
|вылета|Наличие пересадок|Детей|
-----
| 1|Volkov|Afanasiy|Yurievich| 4| 46|Vologda| 19:35|No|
3|
| 2|Pegova|Anastasia|Alexandrovna| 3| 17|Tyumen| 20:12|
Yes| 2|
| 3|Osipova|Yulia|Olegovna| 3| 38|Sankt-Peterburg| 17:10|
No| 1|
| 4|Sayko|Alexander|Nikolaevich| 2| 34|Vladivostok| 18:30|
Yes| 0|
| 5|Golubev|Maxim|Borisovich| 1| 21|Samara| 10:12|
No| 1|
| 6|Korotkov|Dmitry|Vladimirovich| 2| 29|Ufa| 22:19|No|
3|
| 7|Konoplev|Boris|Alexeevich| 3| 44|Perm| 04:45|Yes|
2|
| 8|Novoselov|Nikolay|Ivanovich| 3| 51|Salehard| 10:18|
No| 3|
| 9|Kovalev|Konstantin|Nikolaevich| 4| 55|Ekaterinburg| 15:50|
Yes| 0|
|10|Bilyushova|Kristina|Alexandrovna| 4| 37|Krasnoyarsk| 20:15|
No| 1|
|11|Berezhnoy|Alexander|Pavlovich| 3| 34|Kenerovo| 17:15|
Yes| 2|
|12|Ivanov|Petr|Vasilevich| 2| 24|Moskva| 14:18|No|
1|
|13|Vasilev|Kirill|Igorovich| 4| 35|Tomsk| 21:45|Yes|
1|
|14|Platoshkin|Nikolay|Nikolayevich| 5| 46|Irkutsk| 19:55|
No| 0|
```

15	Vladova	Alla	Yurievna	2	19	Magadan	23:10	Yes
2								
16	Kachurin	Yuri	Vladimirovich	1	15	Surgut	18:29	No
3								
17	Chikina	Ekaterina	Vasilevna	1	24	Petrozavodsk	16:56	
No	2							
18	Malynko	Sergey	Alexeevich	3	20	Kursk	17:48	No
0								
19	Popov	Yury	Valerievich	2	12	Volgograd	10:10	Yes
2								
20	Novikova	Natalia	Nikolaevna	3	15	Novgorod	14:20	
No	3							
21	Bilan	Dmitry	Gennadievich	2	10	Murmansk	09:25	
No	1							
22	Voronin	Pavel	Ivanovich	3	16	Saratov	12:15	Yes
3								
23	Korolev	Vasiliy	Sidorovich	1	28	Cheboksary	07:19	
No	1							
24	Stoyanov	Yury	Stanislavovich	2	14	Ivanovo	03:40	
No	2							

#поиск пассажиров у которых одинаковое число вещей и вес различается не более чем на 3 кг

irina@Irina-VivoBook:~/Prog/Prog_C/Lab22/Filesdb\$./findp -p 3

Пассажиры, у которых багаж совпадает по числу вещей и различается не более чем на 3 кг: найдены!

№	Фамилия	Имя	Отчество	Вещей	Общий вес	Пункт назначения	Время вылета	Наличие пересадок	Детей
---	---------	-----	----------	-------	-----------	------------------	--------------	-------------------	-------

1	Pegova	Anastasia	Alexandrovna	3	17	Tyumen	20:12		
Yes	2								
2	Malynko	Sergey	Alexeevich	3	20	Kursk	17:48	No	
0									

#запуск программы после удаления одного из пассажиров предыдущего поиска

irina@Irina-VivoBook:~/Prog/Prog_C/Lab22/Filesdb\$./filesdb -del 2

Выполнено удаление записи 2

irina@Irina-VivoBook:~/Prog/Prog_C/Lab22/Filesdb\$./findp -p 3

Пассажиры, у которых багаж совпадает по числу вещей и различается не более чем на 3 кг: найдены!

|№ |Фамилия |Имя |Отчество |Вещей|Общий вес|Пункт назначения|Время
вылета|Наличие пересадок|Детей|

| 1| Golubev| Maxim| Borisovich| 1| 21| Samara| 10:12|
No| 1|
| 2| Chikina| Ekaterina| Vasilevna| 1| 24| Petrozavodsk| 16:56|
No| 2|

#запуск программы после удаления одного из пассажиров предыдущего поиска

irina@Irina-VivoBook:~/Prog/Prog_C/Lab22/Filesdb\$./filesdb -del 4

Выполнено удаление записи 4

irina@Irina-VivoBook:~/Prog/Prog_C/Lab22/Filesdb\$./findp -p 3

Пассажиры, у которых багаж совпадает по числу вещей и различается не более чем на 3 кг: найдены!

|№ |Фамилия |Имя |Отчество |Вещей|Общий вес|Пункт назначения|Время
вылета|Наличие пересадок|Детей|

| 1| Bilyushova| Kristina| Alexandrovna| 4| 37| Krasnoyarsk| 20:15|
No| 1|
| 2| Vasilev| Kirill| Igorevich| 4| 35| Tomsk| 21:45| Yes| 1|

#запуск программы после удаления одного из пассажиров предыдущего поиска

irina@Irina-VivoBook:~/Prog/Prog_C/Lab22/Filesdb\$./filesdb -del 8

Выполнено удаление записи 8

irina@Irina-VivoBook:~/Prog/Prog_C/Lab22/Filesdb\$./findp -p 3

Пассажиры, у которых багаж совпадает по числу вещей и различается не более чем на 3 кг: найдены!

|№ |Фамилия |Имя |Отчество |Вещей|Общий вес|Пункт назначения|Время
вылета|Наличие пересадок|Детей|

1	Popov	Yury	Valerievich	2	12	Volgograd	10:10	Yes
2								
2	Bilan	Dmitry	Gennadievich	2	10	Murmansk	09:25	
No	1							

9. **Дневник отладки** должен содержать дату и время сеансов отладки и основные события (ошибки в сценарии и программе, нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание

10. **Замечания автора** по существу работы: замечания отсутствуют.

11. **Вывод:** в ходе выполнения данного задания практикума я научилась работать с файлами с СП Си.

Подпись студента _____