

Отчет по лабораторной работе №VI по курсу практикум на ЭВМ

Студент группы М8О-101Б-20 Ядров Артем Леонидович, № по списку 28

Контакты www, e-mail, icq, skype temayadrow@gmail.com

Работа выполнена: « » _____ 202__ г.

Преподаватель: доцент каф. 806 Никулин Сергей Петрович

Входной контроль знаний с оценкой _____

Отчет сдан « » _____ 202__ г., итоговая оценка _____

Подпись преподавателя _____

1. **Тема:** Обработка последовательной файловой структуры на Си

2. **Цель работы:** Научиться обрабатывать последовательные файловые структуры на Си

3. **Задание (вариант № 38):** Информация о пассажирах аэропорта: фамилия, инициалы, количество вещей, общий вес вещей, пункт назначения, время вылета, наличие пересадок, информация о детях

Дать сведения о пассажирах, число вещей которых не меньше, чем в любом другом багаже, а общий вес вещей не больше, чем в любом другом багаже с этим же числом вещей

4. **Оборудование (лабораторное):**

ЭВМ Intel Pentium G2140, процессор 3.30 GHz, имя узла сети Cameron с ОП 8096 Мб, НМД 7906 Мб. Терминал ASUS адрес dev/pets/3 Принтер HP Laserjet 6P
Другие устройства _____

Оборудование ПЭВМ студента, если использовалось:

Процессор Intel core i5-7300HQ 2.50 GHz с ОП 8096 Мб, НМД 131072 Мб. Монитор ASUS
Другие устройства _____

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

Операционная система семейства Unix, наименование Ubuntu версия 18.15.0
интерпретатор команд bash версия 4.4.20
Система программирования GNU версия 5.8.13
Редактор текстов emacs версия 25.2.2
Утилиты операционной системы cat
Прикладные системы и программы _____
Местонахождение и имена файлов программ и данных stud/208104

Программное обеспечение ЭВМ студента, если использовалось:

Операционная система семейства Unix, наименование Fedora версия 33
интерпретатор команд bash версия 5.0.17
Система программирования Clion версия 2020.3
Редактор текстов emacs версия 25.2.2
Утилиты операционной системы cat, gcc
Прикладные системы и программы _____

Местонахождение и имена файлов программ и данных на домашнем компьютере home/Temi4

6. Идея, метод, алгоритм решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)

Для представления базы данных создадим в заголовочном файле 2 структуры:

- typedef struct {
int hour, minute;
} TIME;

Структура время является вспомогательной структурой. С помощью нее определим дальнейшую структуру

- typedef struct {
char surname[25];
char initials[2];
int num;
int weight;
char to[50];
TIME time;
int transfer;
int child;
} passenger;

Структура «passanger» и будет представлять нашу базу данных. В ней содержатся поля, необходимые для инициализации пассажира аэропорта (фамилия, инициалы, количество вещей, общий вес вещей, пункт назначения, время вылета, наличие пересадок, информация о детях (количество детей))

Опишем **первую программу** (passanger_dump.c), которая считывает базу данных с текстового файла и заносит данные в бинарный файл. С помощью функции стандартной библиотечной функции определим количество параметров, поданных в терминале во время запуска программы. Если argc возвращает значение, не равное 3 (программа, входной файл, выходной файл), то вызовем функцию void Usage(), выводящую подсказку по использованию программы. Если значение функции argc равно 3, то зададим входной (in) и выходной файл (out) с помощью функций fopen и argv(B argv[1] должно лежать имя входного файла, argv[2] - выходного). Если хотя бы один файл не удалось открыть, то завершаем программу. Далее запускаем цикл while, в котором условие задается функцией считывания данных пассажира (int readpassanger), которая выводит 1, если удалось считать данные пассажира, и 0 - в противном случае. Записываем с помощью функции fwrite данные пассажира в бинарный файл.

Опишем **вторую программу** (cool_passangers.c), которая считывает базу данных с бинарного файла и выводит данные о пассажирах с максимальным количеством вещей и минимальным (среди пассажиров с таким же количеством вещей) общим весом вещей. Для начала с помощью функции argc определим количество заданных параметров. Если количество не равно 3 (имя программы, ключ, входной файл), то выведем при помощи функции void Usage() подсказку по использованию программы. Далее проверим, какой ключ задан. Ключ содержится в argv[1], а имя файла - в argv[2]. Будем хранить 2 переменные (f, p), принимающие 1, если задан соответствующий ключ, и 0 в противном случае. Будем использовать функцию strcmp для сравнения строк (возвращает 0, если строки одинаковые, и 1, если разные). Если файл невозможно открыть, то выведем соответствующую ошибку. Если задан ключ «-f», то выведем шапку таблицы. Далее пройдемся в цикле по бинарному файлу, считывая данные пассажиров. Если задан ключ «-p», то будем искать максимальное количество вещей у пассажиров. Запишем это значение в переменную n (Условие $r \& \& pas.nam > n$ прекращает проверяться, если $r = 0$). Если задан ключ «-f», то просто выведем очередную строку таблицы. После завершения цикла завершаем программу, если задан ключ «-f». Далее еще один цикл, в котором ищем минимальный общий вес вещей w среди пассажиров, число вещей которых равно n. После этого выведем шапку таблицы ответа и запустим еще один цикл, в котором будем выводить данные пассажиров, число вещей которых максимально, а общий вес минимальный (среди пассажиров с таким же числом вещей).

Оценим **временную сложность** алгоритма: Мы совершаем 3 (или 1) цикла (цикл) по всему файлу, что означает, что наша сложность равна $O(3*n)=O(n)$.

Теперь оценим **пространственную сложность** алгоритма: Мы храним лишь ссылку на файл, переменные (n, w, f, p), которые не зависят от размера файла, и одну переменную (pas) типа «passanger», которая тоже не зависит от размера файла. Следовательно, пространственная сложность алгоритма равна $O(1)$.

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

Ivanov	AL	0	0	Moscow	00:00	1	0
Smirnov	LA	3	40	New York	13:33	0	3
Kuznetsov	AD	20	80	London	17:20	1	2
Popov	DA	19	59	Saint-Peterburg	18:03	0	0
Vasiliev	IV	10	69	Kazan	19:00	1	0
Petrov	VI	10	77	Yoshkar-Ola	19:28	1	4
Sokolov	MV	1	33	Berlin	14:50	1	0
Mikhailov	IP	15	45	Munchen	15:30	0	3
Novikov	PI	17	34	Manchester	12:00	1	2

Fedorov MI	14	28	Barcelona	00:45	1	1		
Morozov MA	3	6	Madrid	01:30	0	0		
Volkov AM	2	4	Milan	2:37	1	1		
Alekseev	KL	5	10	Rim	23:35	1	0	
Lebedev LK	20	80	Tokio	22:00	0	0		
Semyonov	NB	20	99	Moscow	00:00	1	0	
Egorov BN	20	81	Ekaterinburg	07:10	0	0		
Pavlov GH	14	100	Moscow	00:00	1	0		
Kozlov HG	13	101	Kazan	19:00	0	1		
Stepanov	IC	11	31	London	17:20	1	0	
Nikolaev	CI	10	41	Leeds	16:30	0	0	
Orlov AV	8	21	Chicago	21:30	1	0		
Andreev VA	6	14	Paris	0	0			
Makarov AO	4	11	Monaco	23:14	0	0		
Nikitin OA	13	10	Cheboksary	01:06	1	1		
Zakharov	OO	12	14	Kazan	19:00	1	1	
Ivanov AL	0	0	Moscow	00:00	1	0		
Smirnov	LA	0	0	New York	13:33	0	3	
Kuznetsov	AD	0	0	London	17:20	1	2	
Popov DA	0	0	Saint-Peterburg	18:03	0	0		
Vasiliev IV	0	0	Kazan	19:00	1	0		
Petrov VI	0	0	Yoshkar-Ola	19:28	1	4		
Sokolov MV	0	0	Berlin	14:50	1	0		
Mikhailov	IP	0	0	Munchen	15:30	0	3	
Novikov PI	0	0	Manchester	12:00	1	2		
Fedorov MI	0	0	Barcelona	00:45	1	1		
Morozov	MA	0	0	Madrid	01:30	0	0	
Volkov AM	0	0	Milan	2:37	1	1		
Alekseev	KL	0	0	Rim	23:35	1	0	
Lebedev	LK	0	0	Tokio	22:00	0	0	
Semyonov	NB	0	0	Moscow	00:00	1	0	
Egorov BN	0	0	Ekaterinburg	07:10	0	0		
Pavlov GH	0	0	Moscow	00:00	1	0		
Kozlov HG	0	0	Kazan	19:00	0	1		
Stepanov	IC	0	0	London	17:20	1	0	
Nikolaev	CI	0	0	Leeds	16:30	0	0	
Orlov AV	0	0	Chicago	21:30	1	0		
Andreev	VA	0	0	Paris	0	0		
Makarov	AO	0	0	Monaco	23:14	0	0	
Nikitin OA	0	0	Cheboksary	01:06	1	1		
Zakharov	OO	0	0	Kazan	19:00	1	1	
Ivanov AL	-1	0	Moscow	00:00	1	0		
Smirnov	LA	-3	40	New York	13:33	0	3	
Kuznetsov	AD	-20	80	London	17:20	1	2	
Popov DA	-19	59	Saint-Peterburg	18:03	0	0		
Vasiliev IV	-10	69	Kazan	19:00	1	0		
Petrov VI	-10	77	Yoshkar-Ola	19:28	1	4		
Sokolov MV	-1	33	Berlin	14:50	1	0		
Mikhailov	IP	-15	45	Munchen	15:30	0	3	
Novikov PI	-17	34	Manchester	12:00	1	2		
Fedorov MI	-14	28	Barcelona	00:45	1	1		
Morozov	MA	-3	6	Madrid	01:30	0	0	
Volkov AM	-2	4	Milan	2:37	1	1		
Alekseev	KL	-5	10	Rim	23:35	1	0	
Lebedev	LK	-20	80	Tokio	22:00	0	0	
Semyonov	NB	-20	99	Moscow	00:00	1	0	
Egorov BN	-20	81	Ekaterinburg	07:10	0	0		
Pavlov GH	-14	100	Moscow	00:00	1	0		
Kozlov HG	-13	101	Kazan	19:00	0	1		
Stepanov	IC	-11	31	London	17:20	1	0	
Nikolaev	CI	-10	41	Leeds	16:30	0	0	
Orlov AV	-8	21	Chicago	21:30	1	0		
Andreev	VA	-6	14	Paris	0	0		
Makarov	AO	-4	11	Monaco	23:14	0	0	
Nikitin OA	-13	10	Cheboksary	01:06	1	1		
Zakharov	OO	-12	14	Kazan	19:00	1	1	

Пункты 1-7 отчета составляются строго до начала лабораторной работы.

Допущен к выполнению работы. Подпись преподавателя _____

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

[Temi4@localhost laabs]\$ cat input1

```
Ivanov AL 0 0 Moscow 00:00 1 0
Smirnov LA 3 40 New York 13:33 0 3
Kuznetsov AD 20 80 London 17:20 1 2
Popov DA 19 59 Saint-Peterburg 18:03 0 0
Vasiliev IV 10 69 Kazan 19:00 1 0
Petrov VI 10 77 Yoshkar-Ola 19:28 1 4
Sokolov MV 1 33 Berlin 14:50 1 0
Mikhailov IP 15 45 Munchen 15:30 0 3
Novikov PI 17 34 Manchester 12:00 1 2
Fedorov MI 14 28 Barselona 00:45 1 1
Morozov MA 3 6 Madrid 01:30 0 0
Volkov AM 2 4 Milan 2:37 1 1
Alekseev KL 5 10 Rim 23:35 1 0
Lebedev LK 20 80 Tokio 22:00 0 0
Semyonov NB 20 99 Moscow 00:00 1 0
Egorov BN 20 81 Ekaterinburg 07:10 0 0
Pavlov GH 14 100 Moscow 00:00 1 0
Kozlov HG 13 101 Kazan 19:00 0 1
Stepanov IC 11 31 London 17:20 1 0
Nikolaev CI 10 41 Leeds 16:30 0 0
Orlov AV 8 21 Chicago 21:30 1 0
Andreev VA 6 14 Paris 0 0
Makarov AO 4 11 Monaco 23:14 0 0
Nikitin OA 13 10 Cheboksary 01:06 1 1
Zakharov OO 12 14 Kazan 19:00 1 1
```

[Temi4@localhost laabs]\$ cat input2

```
Ivanov AL 0 0 Moscow 00:00 1 0
Smirnov LA 0 0 New York 13:33 0 3
Kuznetsov AD 0 0 London 17:20 1 2
Popov DA 0 0 Saint-Peterburg 18:03 0 0
Vasiliev IV 0 0 Kazan 19:00 1 0
Petrov VI 0 0 Yoshkar-Ola 19:28 1 4
Sokolov MV 0 0 Berlin 14:50 1 0
Mikhailov IP 0 0 Munchen 15:30 0 3
Novikov PI 0 0 Manchester 12:00 1 2
Fedorov MI 0 0 Barselona 00:45 1 1
Morozov MA 0 0 Madrid 01:30 0 0
Volkov AM 0 0 Milan 2:37 1 1
Alekseev KL 0 0 Rim 23:35 1 0
Lebedev LK 0 0 Tokio 22:00 0 0
Semyonov NB 0 0 Moscow 00:00 1 0
Egorov BN 0 0 Ekaterinburg 07:10 0 0
Pavlov GH 0 0 Moscow 00:00 1 0
Kozlov HG 0 0 Kazan 19:00 0 1
Stepanov IC 0 0 London 17:20 1 0
Nikolaev CI 0 0 Leeds 16:30 0 0
Orlov AV 0 0 Chicago 21:30 1 0
Andreev VA 0 0 Paris 0 0
Makarov AO 0 0 Monaco 23:14 0 0
Nikitin OA 0 0 Cheboksary 01:06 1 1
Zakharov OO 0 0 Kazan 19:00 1 1
```

[Temi4@localhost laabs]\$ cat input3

```
Ivanov AL -1 0 Moscow 00:00 1 0
Smirnov LA -3 40 New York 13:33 0 3
Kuznetsov AD -20 80 London 17:20 1 2
Popov DA -19 59 Saint-Peterburg 18:03 0 0
Vasiliev IV -10 69 Kazan 19:00 1 0
Petrov VI -10 77 Yoshkar-Ola 19:28 1 4
Sokolov MV -1 33 Berlin 14:50 1 0
Mikhailov IP -15 45 Munchen 15:30 0 3
Novikov PI -17 34 Manchester 12:00 1 2
Fedorov MI -14 28 Barselona 00:45 1 1
Morozov MA -3 6 Madrid 01:30 0 0
Volkov AM -2 4 Milan 2:37 1 1
Alekseev KL -5 10 Rim 23:35 1 0
Lebedev LK -20 80 Tokio 22:00 0 0
```

```

Semyonov NB -20 99 Moscow 00:00 1 0
Egorov BN -20 81 Ekaterinburg 07:10 0 0
Pavlov GH -14 100 Moscow 00:00 1 0
Kozlov HG -13 101 Kazan 19:00 0 1
Stepanov IC -11 31 London 17:20 1 0
Nikolaev CI -10 41 Leeds 16:30 0 0
Orlov AV -8 21 Chicago 21:30 1 0
Andreev VA -6 14 Paris 0 0
Makarov AO -4 11 Monaco 23:14 0 0
Nikitin OA -13 10 Cheboksary 01:06 1 1
Zakharov OO -12 14 Kazan 19:00 1 1

```

```
[Temi4@localhost laabs]$ cat passenger.h
```

```
//
```

```
// Created by Temi4 on 04.03.2021.
```

```
//
```

```
#ifndef LAABS_PASSENGER_H
```

```
#define LAABS_PASSENGER_H
```

```
typedef struct{
```

```
    int hour, minute;
```

```
} TIME;
```

```
typedef struct {
```

```
    char surname[25];
```

```
    char initials[2];
```

```
    int num;
```

```
    int weight;
```

```
    char to[50];
```

```
    TIME time;
```

```
    int transfer;
```

```
    int child;
```

```
} passenger;
```

```
#endif //LAABS_PASSENGER_H
```

```
[Temi4@localhost laabs]$ cat passenger_dump.c
```

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

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

```
#include <errno.h>
```

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

```
void usage(){
```

```
    printf("Usage: program input_filename output_filename\n");
```

```
}
```

```
int readpassenger(FILE *in, passenger *p){
```

```
    return fscanf(in, " %[^\\t]\\t%[^\\t]\\t%d\\t%d%[^\\t]\\t%d: %d\\t%d\\t%d\\n", p->surname, p->initials, &p->num, &p->weight,
```

```
p->to, &p->time.hour, &p->time.minute, &p->transfer, &p->child) == 9;
```

```
}
```

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

```
    if (argc != 3){
```

```
        usage();
```

```
        return 1;
```

```
    }
```

```
    passenger p;
```

```
    FILE *out = fopen(argv[2], "w");
```

```
    FILE *in = fopen(argv[1], "r");
```

```
    if (!(out && in)){
```

```
        perror("Can't open file");
```

```
        return 2;
```

```
    }
```

```
    while (readpassenger(in, &p)){
```

```
        fwrite(&p, sizeof (p), 1, out);
```

```
    }
```

```
    return 0;
```

```
}[Temi4@localhost laabs]$ cat cool_passangers.c
```

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

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

```
#include <limits.h>
```

```

#include "passenger.h"

void usage() {
    printf("Usage: program [-key] filename\nKeys: -f xor -p\n");
}

int main(int argc, char *argv[]) {
    FILE *in;
    int f = 0, p = 0;
    if (argc != 3) {
        usage();
        return 1;
    }
    if (strcmp(argv[1], "-f") == 0) {
        f = 1;
        in = fopen(argv[2], "r");
    } else if (strcmp(argv[1], "-p") == 0) {
        p = 1;
        in = fopen(argv[2], "r");
    } else {
        usage();
        return 2;
    }
    passenger pas;
    if (!in) {
        perror("Can not open file\n");
        return 3;
    }
    if (f) {
        printf("_____\n");
        printf("| SURNAME | I | N | W | TO | TIME | TR | CH |\n");
        printf("_____\n");
    }
    int n = -1;
    int w = INT_MAX;
    while (fread(&pas, sizeof(pas), 1, in) == 1) {
        if (p && pas.num > n) {
            n = pas.num;
        }
        if (f) {
            printf("| %-25s| %-3s| %-7d| %-7d| %-19s| %d: %d| %-4d| %-4d|\n", pas.surname, pas.initials, pas.num, pas.weight,
                pas.to, pas.time.hour, pas.time.minute, pas.transfer, pas.child);
            printf("_____\n");
        }
    }
    if (f) {
        return 0;
    }
    fseek(in,
        0, SEEK_SET);
    while (fread(&pas, sizeof(pas), 1, in) == 1) {
        if (pas.num == n && pas.weight < w) {
            w = pas.weight;
        }
    }
    fseek(in,
        0, SEEK_SET);
    if (n == -1) {
        printf("Данные введены неправильно");
        return 4;
    }
    printf("\t\t\t\tANSWER\n");
    printf("_____\n");
    printf("| SURNAME | I | N | W | TO | TIME | TR | CH |\n");
    printf("_____\n");
    while (fread(&pas, sizeof(pas), 1, in) == 1) {
        if (pas.num >= n && pas.weight <= w) {
            printf("| %-25s| %-3s| %-7d| %-7d| %-19s| %d: %d| %-4d| %-4d|\n", pas.surname, pas.initials, pas.num, pas.weight,

```

```

        pas.to, pas.time.hour, pas.time.minute, pas.transfer, pas.child);
    printf("_____ \n");
}
}
return 0;

```

```

}[Temi4@localhost laabs]$ gcc p_
gcc: ошибка: p_: Нет такого файла или каталога
gcc: фатальная ошибка: не заданы входные файлы
компиляция прервана.

```

```

[Temi4@localhost laabs]$ gcc passenger_dump.c

```

```

[Temi4@localhost laabs]$ ./a.out

```

```

Usage: program input_filename output_filename

```

```

[Temi4@localhost laabs]$ ./a.out input1 output1

```

```

[Temi4@localhost laabs]$ ./a.out input2 output2

```

```

[Temi4@localhost laabs]$ ./a.out input3 output3

```

```

[Temi4@localhost laabs]$ gcc cool_passangers.c

```

```

[Temi4@localhost laabs]$ ./a.out

```

```

Usage: program [-key] filename

```

```

Keys: -f xor -p

```

```

[Temi4@localhost laabs]$ ./a.out -f output1

```

SURNAME	I	N	W	TO	TIME	TR	CH
Ivanov	AL	0	0	Moscow	0:0	1	0
Smirnov	LA	3	40	New York	13:33	0	3
Kuznetsov	AD	20	80	London	17:20	1	2
Popov	DA	19	59	Saint-Peterburg	18:3	0	0
Vasiliev	IV	10	69	Kazan	19:0	1	0
Petrov	VI	10	77	Yoshkar-Ola	19:28	1	4
Sokolov	MV	1	33	Berlin	14:50	1	0
Mikhailov	IP	15	45	Munchen	15:30	0	3
Novikov	PI	17	34	Manchester	12:0	1	2
Fedorov	MI	14	28	Barselona	0:45	1	1
Morozov	MA	3	6	Madrid	1:30	0	0
Volkov	AM	2	4	Milan	2:37	1	1
Alekseev	KL	5	10	Rim	23:35	1	0
Lebedev	LK	20	80	Tokio	22:0	0	0
Semyonov	NB	20	99	Moscow	0:0	1	0
Egorov	BN	20	81	Ekaterinburg	7:10	0	0
Pavlov	GH	14	100	Moscow	0:0	1	0
Kozlov	HG	13	101	Kazan	19:0	0	1
Stepanov	IC	11	31	London	17:20	1	0
Nikolaev	CI	10	41	Leeds	16:30	0	0
Orlov	AV	8	21	Chicago	21:30	1	0

```

[Temi4@localhost laabs]$ ./a.out -p output1

```

ANSWER

SURNAME	I	N	W	TO	TIME	TR	CH
---------	---	---	---	----	------	----	----

Kuznetsov	AD 20	80	London	17:20 1	2	
Lebedev	LK 20	80	Tokio	22:0 0	0	
Temi4@localhost laabs \$./a.out -p output2 ANSWER						
SURNAME	I	N	W	TO	TIME	TR CH
Ivanov	AL 0	0	Moscow	0:0 1	0	
Smirnov	LA 0	0	New York	13:33 0	3	
Kuznetsov	AD 0	0	London	17:20 1	2	
Popov	DA 0	0	Saint-Peterburg	18:3 0	0	
Vasiliev	IV 0	0	Kazan	19:0 1	0	
Petrov	VI 0	0	Yoshkar-Ola	19:28 1	4	
Sokolov	MV 0	0	Berlin	14:50 1	0	
Mikhailov	IP 0	0	Munchen	15:30 0	3	
Novikov	PI 0	0	Manchester	12:0 1	2	
Fedorov	MI 0	0	Barselona	0:45 1	1	
Morozov	MA 0	0	Madrid	1:30 0	0	
Volkov	AM 0	0	Milan	2:37 1	1	
Alekseev	KL 0	0	Rim	23:35 1	0	
Lebedev	LK 0	0	Tokio	22:0 0	0	
Semyonov	NB 0	0	Moscow	0:0 1	0	
Egorov	BN 0	0	Ekaterinburg	7:10 0	0	
Pavlov	GH 0	0	Moscow	0:0 1	0	
Kozlov	HG 0	0	Kazan	19:0 0	1	
Stepanov	IC 0	0	London	17:20 1	0	
Nikolaev	CI 0	0	Leeds	16:30 0	0	
Orlov	AV 0	0	Chicago	21:30 1	0	
Temi4@localhost laabs \$./a.out -p output3 Данные введены неправильно						

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 _____

11. Выводы

Я научился обрабатывать файловые структуры на Си.

Недочёты при выполнении задания могут быть устранены следующим образом:

Подпись студента _____