



Отчет по лабораторной работе № VI по курсу Алгоритмы и структуры данных

Студент группы М8О-103Б-22 Клименко Виталий Максимович, № по списку 11

Контакты www, e-mail, icq, skype vitalikklimenko96@gmail.com

Работа выполнена: 10 апреля 2023 г.

Преподаватель: доцент Никулин С.П.

Входной контроль знаний с оценкой _____

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

Подпись преподавателя _____

- Тема:** Обработка последовательной файловой структуры на языке Си
- Цель работы:** Разработать последовательную структуру данных для представления простейшей базы данных на файлах в СП Си в соответствии с заданным вариантом.
- Задание (вариант № 22):** Найти абитуриентов-медалистов, не набравших проходной балл р.
- Оборудование (лабораторное):**
ЭВМ _____, процессор _____, имя узла сети _____ с ОП _____ Мб,
НМД _____ Мб. Терминал _____ адрес _____. Принтер _____
Другие устройства _____
Оборудование ПЭВМ студента, если использовалось:
Процессор Intel 4x 3.5GHz _____ с ОП 16 ГБ _____ НМД HDD 200 ГБ _____. Монитор Встроенный 1920x1080
Другие устройства Touchpad Synaptics
- Программное обеспечение (лабораторное):**
Операционная система семейства _____, наименование _____ версия _____
интерпретатор команд _____ версия _____
Система программирования _____ версия _____
Редактор текстов _____ версия _____
Утилиты операционной системы _____
Прикладные системы и программы _____
Местонахождение и имена файлов программ и данных _____
Программное обеспечение ЭВМ студента, если использовалось:
Операционная система семейства UNIX _____, наименование Pop!_OS _____ версия 22.04 jammy
интерпретатор команд bash _____ версия 5.1.16
Система программирования _____ версия _____
Редактор текстов _____ версия _____
Утилиты операционной системы _____
Прикладные системы и программы _____
Местонахождение и имена файлов программ и данных на домашнем компьютере _____

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

Структура бинарного файла, в который сохраняется база данных:

размер базы данных \n

Имя Инициалы Гендер Школа Медаль Баллы Эссе \n

...

Причем все поля идут подряд, без пробелов. Также, числа в базе данных (все кроме имен, инициалов) хранятся в бинарном виде. Строки базы данных разделяются специальным знаком новой строки.

Поиск полей происходит через отведенную функцию. На вход подаются два списка - список с полями, в которых нужно искать и список с соответствующим запросом на поиск, и количество элементов этих списков. Элементы второго списка - строки вида операцияЧИСЛО (операции - компараторы бэша - gt, ge, lt, le, eq, ne). Функция обрабатывает запрос и проверяет каждую строку один раз, следовательно по времени этот алгоритм - линейный

На вход программе подаются один обязательный флаг - -f с именем файла, из которого нужно считать базу данных. Опциональный флаг -r подается в таком виде -r ИМЯПОЛЯ ОПЕРАЦИЯЧИСЛО. Именно этот флаг отвечает за поиск в базе данных. Если не подать флаг -r, будет выведена вся база данных

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

1. Создать структуру данных Database
2. Реализовать функции добавления рядов в эту структуру
3. Реализовать функцию вывода этой структуры
4. Реализовать функцию сохранения структуры в бинарный файл
5. Реализовать функцию поиска и вывода нужных строк

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)

```
vital@vitos-hp16 MINGW64 /c/important/docs/mai/labs/lvi (master)
$ cat makefile
CC = gcc
CFLAGS = -Wall -Wextra
FILES = main.c db.c

main:
    $(CC) $(CFLAGS) -o main.out $(FILES) -lm

vital@vitos-hp16 MINGW64 /c/important/docs/mai/labs/lvi (master)
$ cat main.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include "db.h"

void print_usage() {
    printf(
        "Usage of program:\n"
        "\t\tNecessary:\n"
        "\t\t-f [FILE]\t\tFilename of database that needs to be read\n"
        "\t\tUnnecessary:\n"
        "\t\t-t-p [FIELD] [MIN]\tPrint all rows that have given field with minimum value\n"
    );
}

void crash() {
    print_usage();
    exit(1);
}

int main(int argc, char const *argv[]) {
    if (argc < 2) {
        crash();
    }

    char *filename = (char*) calloc(256, sizeof(char));

    int fields_num = (argc - 3)/3;

    char **fields = (char**) calloc(fields_num, sizeof(char*));
    for (int i = 0; i < fields_num; ++i) {
        fields[i] = (char*) calloc(10, sizeof(char));
    }

    char **values = (char**) calloc(fields_num, sizeof(char*));
    for (int i = 0; i < fields_num; ++i) {
        values[i] = (char*) calloc(10, sizeof(char));
    }

    int off = 0;
    for (int arg_ind = 1; arg_ind < argc; ++arg_ind) {
        // printf("param: %s\n", argv[arg_ind]);
        switch (argv[arg_ind][1]) {
            case 'f':
                if (arg_ind + 1 < argc) {
                    filename = argv[arg_ind + 1];
                    arg_ind++;
                }
                break;
            case 'p':
                if (arg_ind + 2 < argc) {
                    fields[off] = argv[arg_ind + 1];
                    values[off] = argv[arg_ind + 2];
                    off++;
                    arg_ind++;
                    arg_ind++;
                }
                break;
        }
    }

    if (*filename == NULL) {
        printf("No filename given!\n");
        crash();
    }
}
```

```

Database db;
FILE *f = fopen(filename, "rb");

database_read(&db, f);

database_print_matching(db, fields, values, fields_num);

// printf("%s\n", filename);
// printf("%s\n%d\n", field, maximal_value);

fclose(f);
return 0;
}
vital@vitos-hp16 MINGW64 /c/important/docs/mai/labs/lvi (master)
$ cat db.h
#ifndef DB_H
#define DB_H

typedef struct Database {
    unsigned long long size;
    char **name;
    char **initials;
    short *gender;
    short *school;
    short *medal;
    short *points;
    short *essay;
} Database;

void database_init(Database *db, unsigned long long size);
void database_dump(Database db, FILE *f);
void database_read(Database *db, FILE *f);

void database_print_header();
void database_print_row(Database db, unsigned long long i);
void database_print(Database db);
void database_print_all_min(Database db, char *field, short min);
void database_print_all_max(Database db, char *field, short max);
void database_print_matching(Database db, char **fields, char **values, int count);

void database_add
(
    Database *db,
    char *name, char *initials,
    short gender, short school, short medal, short points, short essay
);
void database_add_stdin(Database *db);

#endif
vital@vitos-hp16 MINGW64 /c/important/docs/mai/labs/lvi (master)
$ cat db.c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <ctype.h>

#include "db.h"

void database_init(Database *db, unsigned long long size) {
    db->size = size;
    db->name = (char**) calloc(size, sizeof(char*));
    db->initials = (char**) calloc(size, sizeof(char*));
    db->gender = (short*) calloc(size, sizeof(short));
    db->school = (short*) calloc(size, sizeof(short));
    db->medal = (short*) calloc(size, sizeof(short));
    db->points = (short*) calloc(size, sizeof(short));
    db->essay = (short*) calloc(size, sizeof(short));

    for (unsigned long long i = 0; i < size; ++i) {
        db->name[i] = (char*) calloc(15, sizeof(char));
        db->initials[i] = (char*) calloc(10, sizeof(char));
        db->gender[i] = -1;
        db->school[i] = -1;
        db->medal[i] = -1;
        db->points[i] = -1;
        db->essay[i] = -1;
    }
}

void database_dump(Database db, FILE *f) {
    fwrite(&db.size, sizeof(unsigned long long), 1, f);
    for (unsigned long long i = 0; i < db.size; ++i) {
        if (db.essay[i] == -1) break;

```

```

        fwrite("\n", sizeof(char), 1, f);

        int for_name = strlen(db.name[i]);
        for (int _ = for_name; _ < 15; ++_) {
            fwrite(" ", sizeof(char), 1, f);
        }
        fwrite(db.name[i], for_name, 1, f);

        int for_initials = strlen(db.initials[i]);
        for (int _ = for_initials; _ < 10; ++_) {
            fwrite(" ", sizeof(char), 1, f);
        }
        fwrite(db.initials[i], for_initials, 1, f);

        fwrite(&db.gender[i], sizeof(short), 1, f);
        fwrite(&db.school[i], sizeof(short), 1, f);
        fwrite(&db.medal[i], sizeof(short), 1, f);
        fwrite(&db.points[i], sizeof(short), 1, f);
        fwrite(&db.essay[i], sizeof(short), 1, f);
    }
}

void database_read(Database *db, FILE *f) {
    fread(&db->size, sizeof(unsigned long long), 1, f);
    database_init(db, db->size);

    char *buf = (char*) calloc(1, sizeof(char));
    for (unsigned long long i = 0; i < db->size; ++i) {
        fread(buf, sizeof(char), 1, f);

        fread(db->name[i], sizeof(char), 15, f);

        fread(db->initials[i], sizeof(char), 10, f);

        fread(&db->gender[i], sizeof(short), 1, f);
        fread(&db->school[i], sizeof(short), 1, f);
        fread(&db->medal[i], sizeof(short), 1, f);
        fread(&db->points[i], sizeof(short), 1, f);
        fread(&db->essay[i], sizeof(short), 1, f);
    }
}

void database_print_header() {
    printf(
        "Name          |"
        "Initials      |"
        "Gender         |"
        "School         |"
        "Medal          |"
        "Points         |"
        "Essay          |"
        "\n"
    );
}

void database_print_row(Database db, unsigned long long i) {
    printf("%16s|", db.name[i]);
    printf("%11s|", db.initials[i]);
    printf("%11hd|", db.gender[i]);
    printf("%11hd|", db.school[i]);
    printf("%11hd|", db.medal[i]);
    printf("%11hd|", db.points[i]);
    printf("%11hd|", db.essay[i]);
    printf("\n");
}

void database_print(Database db) {
    database_print_header();

    for (unsigned long long i = 0; i < db.size; ++i) {
        database_print_row(db, i);
    }
}

void database_print_all_min(Database db, char *field, short min) {
    short g = 0, s = 0, m = 0, p = 0, e = 0; // every single field

    switch (tolower(field[0])) {
        case 'g': g = 1; break;
        case 's': s = 1; break;
        case 'm': m = 1; break;
        case 'p': p = 1; break;
        case 'e': e = 1; break;
    }
}

```

```

if (g == s && s == m && m == p && p == e) { // can't be all 1 because of switch case
    printf("No field with such name!\n");
    return;
}

database_print_header();

for (unsigned long long i = 0; i < db.size; ++i) {
    if (db.essay[i] != -1 &&
        (
            (g && db.gender[i] >= min) || (s && db.school[i] >= min) ||
            (m && db.medal[i] >= min) || (p && db.points[i] >= min) ||
            (e && db.essay[i] >= min)
        )
    ) database_print_row(db, i);
}

}

void database_print_all_max(Database db, char *field, short max) {
    short g = 0, s = 0, m = 0, p = 0, e = 0; // every single field

    switch (tolower(field[0])) {
        case 'g': g = 1; break;
        case 's': s = 1; break;
        case 'm': m = 1; break;
        case 'p': p = 1; break;
        case 'e': e = 1; break;
    }

    if (g == s && s == m && m == p && p == e) { // can't be all 1 because of switch case
        printf("No field with such name!\n");
        return;
    }

    database_print_header();

    for (unsigned long long i = 0; i < db.size; ++i) {
        if (db.essay[i] != -1 &&
            (
                (g && db.gender[i] < max) || (s && db.school[i] < max) ||
                (m && db.medal[i] < max) || (p && db.points[i] < max) ||
                (e && db.essay[i] < max)
            )
        ) database_print_row(db, i);
    }

}

void database_print_matching(Database db, char **fields, char **values, int count) {
    int field_count = count;
    short **ops = (short**) calloc(field_count, sizeof(short*));
    for (int i = 0; i < field_count; ++i) {
        ops[i] = (short*) calloc(3, sizeof(short));
    }
    // ops = [[1, 0, 1], [4, 1, 50], [-1, -1], ...]
    // where ops[i][0] = field
    // ops[i][1] = 0 - less, 1 - greater, 2 - equals, 3 - not equals
    // ops[i][2] = value
    // ops[i][0] in 0..=4, ops[i][1] in 0..=3, ops[i][2] in [_SHORT_MIN, _SHORT_MAX]

    short value = 0;
    for (int i = 0; i < field_count; ++i) {
        short fieldname = -1;
        switch (tolower(fields[i][0])) {
            case 'g': fieldname = 0; break;
            case 's': fieldname = 1; break;
            case 'm': fieldname = 2; break;
            case 'p': fieldname = 3; break;
            case 'e': fieldname = 4; break;
        }

        if (values[i][0] == 'l') { // bash-style comparison
            // ops[i][1] = 0;
            if (values[i][1] == 'e') {
                memmove(values[i], values[i] + 2, strlen(values[i]));
                value = (short) atoi(values[i]) + 1;
            } else {
                memmove(values[i], values[i] + 2, strlen(values[i]));
                value = (short) atoi(values[i]);
            }
        } else if (values[i][0] == 'g') {
            ops[i][1] = 1;
            if (values[i][1] == 'e') {
                memmove(values[i], values[i] + 2, strlen(values[i]));
                value = (short) atoi(values[i]) - 1;
            }
        }
    }
}

```

```

        } else {
            memmove(values[i], values[i] + 2, strlen(values[i]));
            value = (short) atoi(values[i]);
        }
    } else if (values[i][0] == 'e') {
        ops[i][1] = 2;
        memmove(values[i], values[i] + 2, strlen(values[i]));
        value = (short) atoi(values[i]);
    } else if (values[i][0] == 'n') {
        ops[i][1] = 3;
        memmove(values[i], values[i] + 2, strlen(values[i]));
        value = (short) atoi(values[i]);
    } else if ('9' >= values[i][0] && values[i][0] >= '0') {
        ops[i][1] = 2;
        value = (short) atoi(values[i]);
    } else {
        printf("Invalid input!\n");
        return;
    }

    ops[i][0] = fieldname;
    ops[i][2] = value;
}

database_print_header();

short condition, to_check;
for (unsigned long long i = 0; i < db.size; ++i) {
    if (db.essay[i] == -1) break;
    condition = 1;
    to_check = 0;
    for (int j = 0; j < field_count; ++j) {
        value = ops[j][2];
        switch (ops[j][0]) {
            case 0: to_check = db.gender[i]; break;
            case 1: to_check = db.school[i]; break;
            case 2: to_check = db.medal[i]; break;
            case 3: to_check = db.points[i]; break;
            case 4: to_check = db.essay[i]; break;
        }
        switch (ops[j][1]) {
            case 0: condition &= to_check < value; break;
            case 1: condition &= to_check > value; break;
            case 2: condition &= to_check == value; break;
            case 3: condition &= to_check != value; break;
        }
    }
    if (condition) database_print_row(db, i);
}

void database_add
(
    Database *db,
    char *name, char *initials,
    short gender, short school, short medal, short points, short essay
) {
    unsigned long long i;
    for (i = 0; i < db->size; ++i) {
        if (db->essay[i] == -1) break;
    }

    if (i == db->size - 1) {
        printf("No space left!\n");
        return;
    }

    db->name[i] = name;
    db->initials[i] = initials;
    db->gender[i] = gender;
    db->school[i] = school;
    db->medal[i] = medal;
    db->points[i] = points;
    db->essay[i] = essay;
}

void database_add_stdin(Database *db) {
    unsigned long long i;
    for (i = 0; i < db->size; ++i) {
        if (db->essay[i] == -1) break;
    }

    if (i == db->size - 1) {
        printf("No space left!\n");
    }
}

```

```

        return;
    }

    printf("Adding to database...\n");

    printf("Input name: ");
    scanf("%s", db->name[i]);

    printf("Input initials: ");
    scanf("%s", db->initials[i]);

    printf("Input gender (0 - male, 1 - female): ");
    scanf("%hd", &db->gender[i]);

    printf("Input school number: ");
    scanf("%hd", &db->school[i]);

    printf("Input medal (0 - no medal, 1 - has medal): ");
    scanf("%hd", &db->medal[i]);

    printf("Input points: ");
    scanf("%hd", &db->points[i]);

    printf("Input essay (0 - no essay, 1 - has essay): ");
    scanf("%hd", &db->essay[i]);
}

vital@vitos-hp16 MINGW64 /c/important/docs/mai/labs/lvi (master)
$ cat sample_db.c
#include <stdio.h>

#include "db.h"

int create_sample_db() {
    Database db;
    database_init(&db, 30);
    database_add(&db, "zniatos", "KM", 0, 5, 1, 100, 0);
    database_add(&db, "xzzniascasw", "mv", 0, 5, 1, 0, 1);
    database_add(&db, "asczniaw", "xasd", 1, 2, 0, 130, 1);
    database_add(&db, "ascznczxciw", "Kasd", 1, 5, 0, 100, 1);
    database_add(&db, "asazniaw", "sd", 0, 5, 0, 10, 0);
    database_add(&db, "aszniaaw", "ab", 1, 10, 1, 60, 0);
    database_add(&db, "xasczniaw", "as", 0, 3, 0, 135, 1);
    database_add(&db, "azniaasw", "d", 1, 134, 1, 2, 0);
    database_add(&db, "bniaxs", "KM", 0, 5, 1, 100, 0);
    database_add(&db, "bzznxscasw", "mv", 0, 5, 1, 0, 1);
    database_add(&db, "bsczxaw", "xasd", 1, 2, 0, 130, 1);
    database_add(&db, "bsczxzcxiw", "Kasd", 1, 5, 0, 100, 1);
    database_add(&db, "bsazxaw", "sd", 0, 5, 0, 10, 1);
    database_add(&db, "bsznxxw", "ab", 1, 10, 1, 12, 0);
    database_add(&db, "baszniaw", "as", 0, 3, 0, 175, 1);
    database_add(&db, "bznixsw", "d", 1, 14, 0, 2, 0);
    database_add(&db, "aaxw", "aaab", 1, 10, 1, 60, 0);
    database_add(&db, "xniaw", "aaas", 0, 3, 0, 135, 1);
    database_add(&db, "aasw", "aaa", 1, 134, 1, 2, 0);
    database_add(&db, "bs", "aaaM", 0, 77, 1, 100, 0);
    database_add(&db, "bscasw", "aaav", 0, 5, 1, 0, 1);
    database_add(&db, "baw", "aaaasd", 1, 228, 0, 50, 1);
    database_add(&db, "bzxciw", "aaaasd", 1, 1337, 0, 100, 1);
    database_add(&db, "baxw", "aaad", 0, 5, 0, 10, 0);
    database_add(&db, "bxw", "aaab", 1, 10, 1, 13, 0);
    database_add(&db, "biaw", "aaas", 0, 3, 0, 175, 0);
    database_add(&db, "bsw", "aaa", 1, 14, 0, 2, 0);

    // database_print(db);
    FILE *fw = fopen("test.db41", "wb");
    database_dump(db, fw);
    fclose(fw);

    // Database dbr;
    // FILE *fr = fopen("test.db41", "rb");
    // database_read(&dbr, fr);
    // database_print(dbr);
    // fclose(fr);

    return 0;
}

vital@vitos-hp16 MINGW64 /c/important/docs/mai/labs/lvi (master)
$ cat test.db41

```

```

        zniatos      KMd
    xzzniascasw      mv
        asczniaw     xasd

```



```

ascznczxciw    Kasdd
asazniaw       sd

    aszniaw     ab
<
xasczniaw      as
azniaasw       d
bniaks         KMd
bzznxscasw     mv
bsczxaw        xasd
bsczxxxiw      Kasdd
bsazxaw        sd

    bsznxxw     ab

    bascxiaw    as
    bznixsw     d
    aaxw        aaab
<
    xniaw       aaas
    aasw        aaa
    bs          aaaMMd
    bscasw      aaav
    baw         aaaasd2
    bzxciw      aaaasd9d
    baxw        aaad

    bxw         aaab

    biaw        aaas
    bsw         aaa
vital@vitos-hp16 MINGW64 /c/important/docs/mai/labs/LVI (master)
$ make
gcc -Wall -Wextra -o main.out main.c db.c -lm
main.c: In function 'main':
main.c:49:30: warning: assignment discards 'const' qualifier from pointer target type [-Wdiscarded-qualifiers]
      filename = argv[arg_ind + 1];
                        ^
main.c:55:33: warning: assignment discards 'const' qualifier from pointer target type [-Wdiscarded-qualifiers]
      fields[off] = argv[arg_ind + 1];
                        ^
main.c:56:33: warning: assignment discards 'const' qualifier from pointer target type [-Wdiscarded-qualifiers]
      values[off] = argv[arg_ind + 2];
                        ^
main.c:65:19: warning: comparison between pointer and integer
      if (*filename == NULL) {
          ^~

vital@vitos-hp16 MINGW64 /c/important/docs/mai/labs/LVI (master)
$ ./main.out -f test.db41
Name      |Initials  |Gender  |School  |Medal  |Points  |Essay  |
-----|-----|-----|-----|-----|-----|-----|
zniatos|      KM|      0|      5|      1|     100|      0|
xzzniascasw|    mv|      0|      5|      1|      0|      1|
asczniaw|    xasd|      1|      2|      0|     130|      1|
ascznczxciw|  Kasd|      1|      5|      0|     100|      1|
asazniaw|    sd|      0|      5|      0|      10|      0|
aszniaw|    ab|      1|     10|      1|      60|      0|
xasczniaw|    as|      0|      3|      0|     135|      1|
azniaasw|    d|      1|    134|      1|      2|      0|
bniaks|    KM|      0|      5|      1|     100|      0|
bzznxscasw|  mv|      0|      5|      1|      0|      1|
bsczxaw|    xasd|      1|      2|      0|     130|      1|
bsczxxxiw|  Kasd|      1|      5|      0|     100|      1|
bsazxaw|    sd|      0|      5|      0|      10|      1|
bsznxxw|    ab|      1|     10|      1|      12|      0|
bascxiaw|    as|      0|      3|      0|     175|      1|
bznixsw|    d|      1|     14|      0|      2|      0|
aaxw|    aaab|      1|     10|      1|      60|      0|
xniaw|    aaas|      0|      3|      0|     135|      1|
aasw|    aaa|      1|    134|      1|      2|      0|
bs|    aaMM|      0|     77|      1|     100|      0|
bscasw|    aaav|      0|      5|      1|      0|      1|
baw|    aaaasd|      1|    228|      0|      50|      1|
bzxciw|    aaaasd|      1|   1337|      0|     100|      1|
baxw|    aaad|      0|      5|      0|      10|      0|
bxw|    aaab|      1|     10|      1|      13|      0|
biaw|    aaas|      0|      3|      0|     175|      0|
bsw|    aaa|      1|     14|      0|      2|      0|

```

```

vital@vitos-hp16 MINGW64 /c/important/docs/mai/labs/LVI (master)
$ ./main.out -f test.db41 -p Points gt10
Name      |Initials  |Gender  |School  |Medal  |Points  |Essay  |
-----|-----|-----|-----|-----|-----|-----|
zniatos|      KM|      0|      5|      1|     100|      0|
asczniaw|    xasd|      1|      2|      0|     130|      1|

```

ascznczxcw	Kasd	1	5	0	100	1
aszniaw	ab	1	10	1	60	0
xasczniaw	as	0	3	0	135	1
bniaxs	KM	0	5	1	100	0
bsczxaw	xasd	1	2	0	130	1
bsczxxcwi	Kasd	1	5	0	100	1
bsznxxw	ab	1	10	1	12	0
bascxiaw	as	0	3	0	175	1
aaxw	aaab	1	10	1	60	0
xniaw	aaas	0	3	0	135	1
bs	aaaM	0	77	1	100	0
baw	aaaasd	1	228	0	50	1
bzxciw	aaaasd	1	1337	0	100	1
bxw	aaab	1	10	1	13	0
biaw	aaas	0	3	0	175	0

vital@vitos-hp16 MINGW64 /c/important/docs/mai/labs/lvi (master)

\$./main.out -f test.db4l -p Points gt130

Name	Initials	Gender	School	Medal	Points	Essay	
xasczniaw	as	0	3	0	135	1	1
bascxiaw	as	0	3	0	175	1	1
xniaw	aaas	0	3	0	135	1	1
biaw	aaas	0	3	0	175	0	0

vital@vitos-hp16 MINGW64 /c/important/docs/mai/labs/lvi (master)

\$./main.out -f test.db4l -p Points ge130

Name	Initials	Gender	School	Medal	Points	Essay	
asczniaw	xasd	1	2	0	130	1	1
xasczniaw	as	0	3	0	135	1	1
bsczxaw	xasd	1	2	0	130	1	1
bascxiaw	as	0	3	0	175	1	1
xniaw	aaas	0	3	0	135	1	1
biaw	aaas	0	3	0	175	0	0

vital@vitos-hp16 MINGW64 /c/important/docs/mai/labs/lvi (master)

\$./main.out -f test.db4l -p Points 130

Name	Initials	Gender	School	Medal	Points	Essay	
asczniaw	xasd	1	2	0	130	1	1
bsczxaw	xasd	1	2	0	130	1	1

vital@vitos-hp16 MINGW64 /c/important/docs/mai/labs/lvi (master)

\$./main.out -f test.db4l -p Points lt50 -p Medal 1

Name	Initials	Gender	School	Medal	Points	Essay	
xzzniascasw	mv	0	5	1	0	1	1
azniaasw	d	1	134	1	2	0	0
bzznxscasw	mv	0	5	1	0	1	1
bsznxxw	ab	1	10	1	12	0	0
aasw	aaa	1	134	1	2	0	0
bscasw	aaav	0	5	1	0	1	1
bxw	aaab	1	10	1	13	0	0

vital@vitos-hp16 MINGW64 /c/important/docs/mai/labs/lvi (master)

\$./main.out -f test.db4l -p Points lt100 -p Medal 1

Name	Initials	Gender	School	Medal	Points	Essay	
xzzniascasw	mv	0	5	1	0	1	1
aszniaw	ab	1	10	1	60	0	0
azniaasw	d	1	134	1	2	0	0
bzznxscasw	mv	0	5	1	0	1	1
bsznxxw	ab	1	10	1	12	0	0
aaxw	aaab	1	10	1	60	0	0
aasw	aaa	1	134	1	2	0	0
bscasw	aaav	0	5	1	0	1	1
bxw	aaab	1	10	1	13	0	0

vital@vitos-hp16 MINGW64 /c/important/docs/mai/labs/lvi (master)

\$./main.out -f test.db4l -p Points le100 -p Medal 1

Name	Initials	Gender	School	Medal	Points	Essay	
zniatos	KM	0	5	1	100	0	0
xzzniascasw	mv	0	5	1	0	1	1
aszniaw	ab	1	10	1	60	0	0
azniaasw	d	1	134	1	2	0	0
bniaxs	KM	0	5	1	100	0	0
bzznxscasw	mv	0	5	1	0	1	1
bsznxxw	ab	1	10	1	12	0	0
aaxw	aaab	1	10	1	60	0	0
aasw	aaa	1	134	1	2	0	0
bs	aaaM	0	77	1	100	0	0
bscasw	aaav	0	5	1	0	1	1
bxw	aaab	1	10	1	13	0	0

vital@vitos-hp16 MINGW64 /c/important/docs/mai/labs/lvi (master)

\$./main.out -f test.db4l -p Points le100 -p Medal 1 -p Essay 1

Name	Initials	Gender	School	Medal	Points	Essay	
xzzniascasw	mv	0	5	1	0	1	1
bzznxscasw	mv	0	5	1	0	1	1

```

      bscasw|      aaav|      0|      5|      1|      0|      1|
vital@vitos-hp16 MINGW64 /c/important/docs/mai/labs/lvi (master)
$ ./main.out -f test.db41 -p Points 1e30 -p Medal 1
Name      |Initials  |Gender  |School  |Medal  |Points  |Essay  |
xzzniascasw|      mv|      0|      5|      1|      0|      1|
azniaasw|      d|      1|     134|      1|      2|      0|
bzznxscasw|      mv|      0|      5|      1|      0|      1|
bsznxxw|      ab|      1|     10|      1|     12|      0|
aasw|      aa|      1|     134|      1|      2|      0|
bscasw|      aaav|      0|      5|      1|      0|      1|
bxw|      aaab|      1|     10|      1|     13|      0|

```

```

vital@vitos-hp16 MINGW64 /c/important/docs/mai/labs/lvi (master)
$ ./main.out -f test.db41 -p Points 1e10 -p Medal 1
Name      |Initials  |Gender  |School  |Medal  |Points  |Essay  |
xzzniascasw|      mv|      0|      5|      1|      0|      1|
azniaasw|      d|      1|     134|      1|      2|      0|
bzznxscasw|      mv|      0|      5|      1|      0|      1|
aasw|      aa|      1|     134|      1|      2|      0|
bscasw|      aaav|      0|      5|      1|      0|      1|

```

```

vital@vitos-hp16 MINGW64 /c/important/docs/mai/labs/lvi (master)
$ ./main.out -f test.db41 -p Points 1e50 -p Medal 1 -p Points ne0
Name      |Initials  |Gender  |School  |Medal  |Points  |Essay  |
azniaasw|      d|      1|     134|      1|      2|      0|
bsznxxw|      ab|      1|     10|      1|     12|      0|
aasw|      aa|      1|     134|      1|      2|      0|
bxw|      aaab|      1|     10|      1|     13|      0|

```

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 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: Под имя выделено всего 15 знаков, под инициалы только 10, однако считаю этого достаточно (самое длинное русское имя - Абдурахмангаджи) и при надобности легко меняется. Кроме этого нет функции, которая бы динамически выделяла память под новые строки базы данных

11. **Выводы:** Я стал лучше понимать то, как работать с оперативной и долговременной памятью компьютера на языке Си. Я научился использовать бинарные файлы в языке Си. Понял как еще можно взаимодействовать со строками в Си.

Недочёты при выполнении задания могут быть устранены следующим образом: _____

Подпись студента _____