

Отчет по лабораторной работе № VI по курсу Архитектура компьютера и информационных сетей

Студент группы М8О-103Б-22 Ахметшин Булат Рамилевич, № по списку 2

Контакты www, e-mail, icq, skype ahmbulat04@yandex.ru

Работа выполнена: 30.04.2023 г.

Преподаватель: доцент каф. 806 Никулин С.П.

Входной контроль знаний с оценкой _____

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

Подпись преподавателя _____

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

2. **Цель работы:** Научиться разрабатывать программы, использующие файловые структуры. _____

3. **Задание (вариант № 13):** Разработать последовательную структуру данных для представления простейшей базы данных на файлах в СП Си в соответствии с заданным вариантом. Написать программу генерации и вывода таблицы. _____

4. **Оборудование (лабораторное):**

ЭВМ _____, процессор _____, имя узла сети _____ с ОП _____ Мб,
НМД _____ Мб. Терминал _____ адрес _____. Принтер _____
Другие устройства _____

Оборудование ПЭВМ студента, если использовалось:

Процессор Intel(R) Core(TM) i7-10510U с ОП 8 ГБ НМД SSD 512 ГБ . Монитор Встроенный 1920x1080
Другие устройства _____

5. **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 Ubuntu версия 22.04
интерпретатор команд GNU bash версия 5.1.16
Система программирования vscode версия 1.77.3
Редактор текстов Sublime Text 3 версия 3211
Утилиты операционной системы Стандартные утилиты OS Linux

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

Планируется написать две программы: генератор бд (или таблицы) и программу для печати элементов бд. Генератор будет обладать несколькими параметрами: имя таблицы, режим генерации (запись или дополнение), множество используемых имен, групп и оценок, количество генерируемых объектов и краткий мануал пользователя. Программа для печати будет обрабатывать параметры имени таблицы, количество выводимых объектов, режим печати (последовательная печать или лишь определенных элементов, согласно варианту) и вывода мануальной страницы.

Генератор будет записывать сгенерированные элементы в выбранный бинарный файл последовательно, так же будет происходить и чтение элементов таблицы.

Количество полей объектов статично, так что их пространственная и временная сложность генерации и печати пропорциональна их количеству $O(n)$.

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

- (a) Составить списки используемых имен, групп и оценок, вынести их в отдельный файл.
- (b) Написать программу генерации бд.
- (c) Написать программу печати элементов указанной в параметрах программы бд, согласно варианту.

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)

```
bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI$ script proto
Script started, output log file is 'proto'.
bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI$ ls
cell.h  functions.c  functions.h  names_rus  print_table.c  proto  stdgrouplist  stdmarklist  stdnamelist  table_generator.c
bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI$ cat cell.h
#ifndef _CELL_H

#define _CELL_H

#include <stdlib.h>
#include <inttypes.h>

#define NAME_SIZE 20
#define GROUP_SIZE 15
#define MARKS_COUNT 4

#define TABLENAME_LENGTH 20

typedef struct mark {
    int8_t type; // Зачет/экзамен - 0/1
    int8_t mark; // Оценка (от 2 до 5, если экзамен и 0 или 1, если зачет)
} mark;

typedef struct cell {
    char NAME[NAME_SIZE]; // ФИО
    int8_t sex; // Пол
    char group[GROUP_SIZE]; // Номер группы
    mark marks[MARKS_COUNT]; // Оценки по предметам:
                                // АСД, Дискретная математика,
                                // ЛинАл, МатАн, Практикум программирования,
                                // Разработка IT-проектов, Физическая культура,
                                // Философия, Экономическая теория, Иностранный язык
} cell;

#endif
bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI$ cat functions.c
#include "functions.h"

void cut_last(char* s, uint64_t size) {
    char* s_ = (char*)malloc(sizeof(char) * (size - 1));
    for (uint64_t i = 0; i < size - 1; ++i) {
        s_[i] = s[i];
    }
    free(s);
    s = s_;
}

void copy_str(char* s, const char* s_, uint64_t sz1, uint64_t sz2) {
    uint64_t m = (sz1 < sz2 ? sz1 : sz2);
    for (uint64_t i = 0; i < m; ++i) {
        s[i] = s_[i];
    }
}
bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI$ cat table_generator.c
#include <stdio.h>
#include <inttypes.h>
#include <time.h>
#include <stdlib.h>
#include <string.h>

#include "cell.h"
#include "functions.h"

#define close_files() { \
    if (! (namelist == NULL)) { \
        fclose(namelist); \
    } \
    if (! (grouplist == NULL)) { \
        fclose(grouplist); \
    } \
    if (! (marklist == NULL)) { \
        fclose(marklist); \
    } \
}

#define free_data() { \
    free(names); \
    free(groups); \
}
```

```

    free(marks);      \
    free(tablename);  \
}

void print_help() {
    printf("This program generates data tables with information of students as their name, sex, group and marks.\n");
    printf("Parameters:\n");
    printf("\t-t or --tablename \tset name of generating table.           Default value is my_table\n");
    printf("\t-r or --regime   \tset regime of writing: rewrite or append. Default value is ab\n");
    printf("\t-n or --namelist  \tset list of using names.           Default value is stdnamelist\n");
    printf("\t-g or --grouplist \tset list of using groups.          Default value is stdgrouplist\n");
    printf("\t-m or --marklist  \tset list of using marks.           Default value is stdmarklist\n");
    printf("\t-c or --count     \tset count of generating objects.      If not specified - program will create empty table\n");
    printf("\t-h or --help      \tshow this help page.\n");
    exit(0);
}

int main(int8_t argc, char** argv) {

    // Count of objects to generate
    uint64_t count = 5;
    // Pointers to data files
    FILE* namelist = NULL, * grouplist = NULL, * marklist = NULL;
    // Name of table
    char* tablename = (char*)malloc(sizeof(char) * TABLENAME_LENGTH);
    int8_t name_given = 0;
    char regime[2] = "ab"; // "w" - rewrite, "a" - augment.

    // Parse input
    for (int8_t i = 1; i < argc; ++i) {
        if (! (strcmp(argv[i], "-t") && strcmp(argv[i], "--tablename"))) {
            strcpy(tablename, argv[i + 1]); name_given = 1; ++i;
        } else if (! (strcmp(argv[i], "-r") && strcmp(argv[i], "--regime"))) {
            strcpy(regime, argv[i + 1]); ++i;
            if ((regime[0] != 'w') && (regime[0] != 'a')) {
                printf("Wrong regime given: %s\n", regime);
                close_files();
                exit(-1);
            }
        }

        } else if (! (strcmp(argv[i], "-n") && strcmp(argv[i], "--namelist"))) {
            namelist = fopen(argv[i + 1], "r"); ++i;
        } else if (! (strcmp(argv[i], "-g") && strcmp(argv[i], "--grouplist"))) {
            grouplist = fopen(argv[i + 1], "r"); ++i;
        } else if (! (strcmp(argv[i], "-m") && strcmp(argv[i], "--marklist"))) {
            marklist = fopen(argv[i + 1], "r"); ++i;
        } else if (! (strcmp(argv[i], "-c") && strcmp(argv[i], "--count"))) {
            char* end;
            count = strtoul(argv[i + 1], &end, 10); ++i;
            if (*end) {
                char* arg = (char*)malloc(sizeof(char) * 20);
                strcpy(arg, argv[i + 1]);
                printf("Wrong count was given: %s\n", arg);
                free(arg);
                free(tablename);
                close_files();
                exit(-1);
            }
        } else if (! (strcmp(argv[i], "-h") && strcmp(argv[i], "--help"))) {
            print_help();
            free(tablename);
            close_files();
            exit(0);
        } else {
            char* arg = (char*)malloc(sizeof(char) * 20);
            strcpy(arg, argv[i]);
            printf("Wrong parameter: %s\n", arg);
            free(arg);
            free(tablename);
            close_files();
            exit(-1);
        }
    }

    if (! name_given) {
        strcpy(tablename, "my_table");
    }

    // Creating pointer to table file
    FILE* table = fopen(tablename, regime);

    // Check if opened
    if (table == NULL) {
        printf("Error: Can not create or open table with filename: %s\n", tablename);
        close_files();
    }
}

```

```

    exit(-1);
}

// Set default data files
if (namelist == NULL) {
    namelist = fopen("stdnamelist", "r");
    if (namelist == NULL) {
        printf("Error: Can not open the standard list of names.\n");
        free(tablename);
        close_files();
        exit(-1);
    }
}

if (grouplist == NULL) {
    grouplist = fopen("stdgrouplist", "r");
    if (grouplist == NULL) {
        printf("Error: Can not open the standard list of groups.\n");
        free(tablename);
        close_files();
        exit(-1);
    }
}

if (marklist == NULL) {
    marklist = fopen("stdmarklist", "r");
    if (marklist == NULL) {
        printf("Error: Can not open the standard list of marks.\n");
        free(tablename);
        close_files();
        exit(-1);
    }
}

if (count == 0) {
    printf("%s table created, 0 objects was generated\n", tablename);
    free(tablename);
    close_files();
    exit(0);
}

// Upload data in memory
char** names, ** groups; int8_t* marks,* sexes;
uint32_t cnames, csexes, cgroups, cmarks;

// Names and sexes
{
    size_t len = 20; size_t chrs;
    char* buff = (char*)malloc(sizeof(char) * len);
    if ((chrs = getline(&buff, &len, namelist)) != -1) {
        sscanf(buff, "%d", &cnames);
        csexes = cnames;
        names = (char**)malloc(sizeof(char*) * cnames);
        sexes = (int8_t*)malloc(sizeof(int8_t) * csexes);
        uint8_t i = 0;
        while ((chrs = getline(&buff, &len, namelist)) != -1) {
            if (i % 2 == 0) {
                names[i / 2] = (char*)malloc(sizeof(char) * len);
                copy_str(names[i / 2], buff, chrs - 1, len); ++i;
            } else {
                sexes[i / 2] = buff[0] - 48; ++i;
            }
        }
        if (i % 2 != 0) {
            printf("Error: Incorrectly set namelist file: odd NAME or sex\n");
        }
        if ((i / 2) < cnames) {
            cnames = i / 2;
            csexes = i / 2;
        }
    } else {
        printf("Error: Can not read from namelist\n");
        close_files();
        exit(-1);
    }
    free(buff);
}

// Groups
{
    size_t len = 12; int8_t chrs;
    char* buff = (char*)malloc(sizeof(char) * len);
    if ((chrs = getline(&buff, &len, grouplist)) != -1) {
        sscanf(buff, "%d", &cgroups);
        groups = (char**)malloc(sizeof(char*) * cgroups);
        uint8_t i = 0;
        while ((chrs = getline(&buff, &len, grouplist)) != -1) {

```

```

        groups[i] = (char*)malloc(sizeof(char) * len);
        copy_str(groups[i], buff, chrs - 1, len); ++i;
    }
    if (i < cgroups) {
        cgroups = i;
    }
} else {
    printf("Error: Can not read from grouplist\n");
    close_files();
    exit(-1);
}

}
// Marks
{
    size_t len = 2; int8_t chrs;
    char* buff = (char*)malloc(sizeof(char) * len);
    if ((chrs = getline(&buff, &len, marklist)) != -1) {
        sscanf(buff, "%d", &cmarks);
        marks = (int8_t*)malloc(sizeof(int8_t) * cmarks);
        uint8_t i = 0;
        while ((chrs = getline(&buff, &len, marklist)) != -1) {
            marks[i] = buff[0] - 48; ++i;
        }
        if (i < cmarks) {
            cmarks = i;
        }
    } else {
        printf("Error: Can not read from marklist\n");
        close_files();
        exit(-1);
    }
}

}

// Close files
close_files();

// List of generating objects
cell student;

srand(time(NULL)); // Setting random

size_t r;
for (uint64_t i = 0; i < count; ++i) {
    uint64_t r = rand();
    strcpy(student.NAME, names[r % cnames]);
    student.sex = sexes[r % csexes];
    strcpy(student.group, groups[rand() % cgroups]);
    for (uint8_t j = 0; j < MARKS_COUNT; ++j) {
        student.marks[j].mark = marks[rand() % cmarks];
        if (1 < student.marks[j].mark) {
            student.marks[j].type = 0; // Test
        } else {
            student.marks[j].type = 1; // Exam
        }
    }
    fwrite(&student, sizeof(student), 1, table);
}

if (regime == "wb")
    printf("%s table created, %ld objects was generated\n", tablename, count);
else if (regime == "ab")
    printf("%s table appended, %ld objects was generated\n", tablename, count);

free_data();
fclose(table);

return 0;
} bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI$ cat print_table.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include "cell.h"
#include "functions.h"

void print_object(const cell* s, uint64_t ind) {
    printf("%ld\t", ind);
    uint8_t j = 0;
    while (s->NAME[j] != '\0') {
        printf("%c", s->NAME[j]); ++j;
    }
    while (j < 20) {
        printf(" "); ++j;
    }
}

```

```

    }
    if (s->sex) {
        printf("|female |");
    } else {
        printf("| male |");
    }
    j = 0;
    while (s->group[j] != '\0') {
        printf("%c", s->group[j]); ++j;
    }
    while (j < 15) {
        printf(" "); ++j;
    }
    printf("| %d | %d | %d | %d | \n", \
        s->marks[0].mark, s->marks[1].mark, s->marks[2].mark, s->marks[3].mark);
}

void print_help() {
    printf("This program prints objects of given table.\n");
    printf("Parameters:\n");
    printf("\t-t or --tablename\tset name of table\n");
    printf("\t-c or --count\t\tset count of objects to be printed. if not specified prints all objects depending on regime\n");
    printf("\t-p or --parameter\tset group that only female students of group have only one 5 mark\n");
    printf("\t-f or --footprint\tstandart printing, set as default\n");
    printf("\t-h or --help\t\tshow this help page\n");
}

uint8_t count_of_marks(const cell* s, uint8_t mark) {
    uint8_t count = 0;
    for (uint8_t i = 0; i < MARKS_COUNT; ++i)
        if (s->marks[i].mark == mark)
            ++count;
    return count;
}

// int8_t cmp_groups(char g1[GROUP_SIZE], char g2[GROUP_SIZE]) {
//     uint8_t i = 0;
//     while (g1[i] == g2[i]) {
//         ++i;
//     }
//     if (0 < i && (g1[i] == '\0' || g2[i] == '\0'))
//         return i;
// }

int main(uint8_t argc, char** argv) {

    char* tablename = (char*)malloc(sizeof(char) * TABLENAME_LENGTH);
    char group[GROUP_SIZE];
    uint64_t count = 0;
    int8_t name_given = 0, count_given = 0;

    int8_t regime = -1;

    for (uint8_t k = 1; k < argc; ++k) {
        if (! (strcmp(argv[k], "-t") && strcmp(argv[k], "--tablename"))) {
            strcpy(tablename, argv[k + 1]); name_given = 1; ++k;
        } else if (! (strcmp(argv[k], "-c") && strcmp(argv[k], "--count"))) {
            char* end;
            count = strtoul(argv[k + 1], &end, 10); count_given = 1; ++k;
            if (*end) {
                char* arg = (char*)malloc(sizeof(char) * 20);
                strcpy(arg, argv[k + 1]);
                printf("Wrong count was given: %s\n", arg);
                free(arg);
                free(tablename);
                exit(0);
            }
        } else if (! (strcmp(argv[k], "-p") && strcmp(argv[k], "--parameter"))) {
            if (regime != -1) {
                printf("Please choose only one regime of printing\n");
                free(tablename);
                exit(-1);
            }
            regime = 1;
            strcpy(group, argv[k + 1]); ++k;
        } else if (! (strcmp(argv[k], "-f") && strcmp(argv[k], "--footprint"))) {
            if (regime != -1) {
                printf("Please choose only one regime of printing\n");
                free(tablename);
                exit(-1);
            }
            regime = 0;
        } else if (! (strcmp(argv[k], "-h") && strcmp(argv[k], "--help"))) {

```

```

        print_help();
        free(tablename);
        exit(0);
    } else {
        char* arg = (char*)malloc(sizeof(char) * 20);
        strcpy(arg, argv[k]);
        printf("Wrong parameter: %s\n", arg);
        free(arg);
        free(tablename);
        exit(-1);
    }
}

if (!name_given) {
    printf("No name given, please input name of list that is supposed to be printed\n");
    free(tablename);
    exit(-1);
}

if (!count_given) {
    count = UINT64_MAX;
}

if (count == 0) {
    free(tablename);
    exit(0);
}

FILE* f = fopen(tablename, "r");

if (f == NULL) {
    printf("Can not open file\n");
    exit(-1);
}

if (regime == -1) {
    regime = 0;
}

cell student;

uint64_t i = 1;
while (1) {
    if (i > count) {
        break;
    }
    if (fread(&student, sizeof(student), 1, f) == 1) {
    } else {
        if (feof(f)) {
            //printf("End of file\n");
            free(tablename);
            exit(0);
        } else {
            printf("Error while reading file\n");
            exit(-1);
        }
    }

    if (i == 1) {
        printf("#\t|          NAME          | Sex |      Group      |"
               " ADS | DM | LinA1 | MathA1 | \n");
    }

    if (regime == 1) {
        if (student.sex == 1 && count_of_marks(&student, 5) == 1 && ! (strcmp(student.group, group)))
            print_object(&student, i);
    } else {
        print_object(&student, i);
    }
    ++i;
}

free(tablename);

return 0;
}bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI$ cat stdnamelist
40
David Hughes
0
Jenny Morris
1
Linda Zuniga
1
Kathy Valdez

```


1
Daniel Marshall
0
Brian Doyle
0
Ana Davis
0
Lindsay Johnson
1
Brad Garner
0
Ebony Cooper
1
Maria Dunn
1
Janet Harrell
1
Karen Wall
1
Mary Walsh
1
Rachel Garcia
1
Richard Carroll
0
Angela Bowen
1
Edwin Miller
0
Janet Jordan
1
Christopher Jordan
0
Ann Jones
1
Lauren Valentine
1
Ana Wiggins
1
Steven Lewis
0
Vincent Benitez
0
Carlos Roberts
0
Lisa Parks
1
Benjamin Collier
0
Jessica Hernandez
1
Nicholas Gonzalez
0
Ryan Castillo
0
Wendy Tran
1
Dr. David Mitchell
0
Ronald Brown
0
Michelle Lewis
1
Dylan Peterson
0
Lindsey Smith
1
Teresa Cruz
1
Kerry Jones
1
Micheal Brewer
Obulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI\$ cat stdgrouplist
12
M1I-101B-22
M2I-102B-22
M3I-103B-22
M4I-104B-22
M5I-105B-22
M6I-106B-22
M7I-107B-22
M8I-108B-22
M9I-100B-22
M10I-110B-22
M11I-111B-22

M12I-112B-22bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI\$ cat stdmarklist

4
2
3
4

5bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI\$ gcc -g functions.c table_generator.c -o table_generator

bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI\$./table_generator -c 20

bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI\$ gcc -g functions.c print_table.c -o print_table

bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI\$./print_table -t my_table -c 20

#	NAME	Sex	Group	ADS	DM	LinAl	MathAn
1	Lindsey Smith	female	M3I-103B-22	5	2	3	3
2	Ann Jones	female	M6I-106B-22	4	5	4	2
3	Lindsey Smith	female	M10I-110B-22	2	4	2	5
4	Steven Lewis	male	M11I-111B-22	2	5	4	4
5	Maria Dunn	female	M9I-100B-22	5	4	3	2
6	Kathy Valdez	female	M3I-103B-22	5	5	4	2
7	Daniel Marshall	male	M7I-107B-22	3	4	3	5
8	Michelle Lewis	female	M10I-110B-22	2	5	5	3
9	Kerry Jones	female	M11I-111B-22	5	4	3	3
10	Ann Jones	female	M8I-108B-22	4	2	4	5
11	Jessica Hernandez	female	M2I-102B-22	3	5	2	5
12	Kathy Valdez	female	M6I-106B-22	4	2	5	5
13	David Hughes	male	M3I-103B-22	3	2	3	2
14	Teresa Cruz	female	M4I-104B-22	5	2	3	2
15	Ryan Castillo	male	M6I-106B-22	2	2	3	4
16	Christopher Jordan	male	M7I-107B-22	5	3	3	2
17	Vincent Benitez	male	M2I-102B-22	3	4	3	2
18	Ana Davis	male	M2I-102B-22	4	5	4	5
19	Janet Harrell	female	M4I-104B-22	4	4	2	5
20	Lindsay Johnson	female	M11I-111B-22	3	5	4	4

bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI\$./print_table -t my_table -c 20 -p M3I-103B-22

#	NAME	Sex	Group	ADS	DM	LinAl	MathAn
1	Lindsey Smith	female	M3I-103B-22	5	2	3	3

bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI\$./print_table -t my_table -c 20 -p M6I-106B-22

#	NAME	Sex	Group	ADS	DM	LinAl	MathAn
2	Ann Jones	female	M6I-106B-22	4	5	4	2

bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI\$./print_table -t my_table -c 20 -p M11I-111B-22

#	NAME	Sex	Group	ADS	DM	LinAl	MathAn
9	Kerry Jones	female	M11I-111B-22	5	4	3	3
20	Lindsay Johnson	female	M11I-111B-22	3	5	4	4

bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI\$ cat > new_groups

10
A
B
C
D
E
F
G
H
M
N

bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI\$./table_generator -t my_table -r ab -c 10 -g new_groups

bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI\$./print_table -t my_table -c 30

#	NAME	Sex	Group	ADS	DM	LinAl	MathAn
1	Lindsey Smith	female	M3I-103B-22	5	2	3	3
2	Ann Jones	female	M6I-106B-22	4	5	4	2
3	Lindsey Smith	female	M10I-110B-22	2	4	2	5
4	Steven Lewis	male	M11I-111B-22	2	5	4	4
5	Maria Dunn	female	M9I-100B-22	5	4	3	2
6	Kathy Valdez	female	M3I-103B-22	5	5	4	2
7	Daniel Marshall	male	M7I-107B-22	3	4	3	5
8	Michelle Lewis	female	M10I-110B-22	2	5	5	3
9	Kerry Jones	female	M11I-111B-22	5	4	3	3
10	Ann Jones	female	M8I-108B-22	4	2	4	5
11	Jessica Hernandez	female	M2I-102B-22	3	5	2	5
12	Kathy Valdez	female	M6I-106B-22	4	2	5	5
13	David Hughes	male	M3I-103B-22	3	2	3	2
14	Teresa Cruz	female	M4I-104B-22	5	2	3	2
15	Ryan Castillo	male	M6I-106B-22	2	2	3	4
16	Christopher Jordan	male	M7I-107B-22	5	3	3	2
17	Vincent Benitez	male	M2I-102B-22	3	4	3	2
18	Ana Davis	male	M2I-102B-22	4	5	4	5
19	Janet Harrell	female	M4I-104B-22	4	4	2	5
20	Lindsay Johnson	female	M11I-111B-22	3	5	4	4
21	Angela Bowen	female	E	5	4	3	2
22	Maria Dunn	female	M	4	2	4	5
23	Janet Harrell	female	G	4	3	3	4
24	Lindsay Johnson	female	E	4	3	5	3
25	Rachel Garcia	female	G	2	5	3	4
26	Christopher Jordan	male	H	2	5	2	3
27	Dylan Peterson	male	A	3	3	4	2
28	Lauren Valentine	female	C	4	5	5	5
29	Lauren Valentine	female	C	5	5	5	5
30	David Hughes	male	B	3	2	3	5

```

bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI$ ./print_table -t my_table -c 30 -p E
# | NAME | Sex | Group | ADS | DM | LinA1 | MathAn |
21 | Angela Bowen | female | E | 5 | 4 | 3 | 2 |
24 | Lindsay Johnson | female | E | 4 | 3 | 5 | 3 |
bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI$ ./print_table -t my_table -c 30 -p G
# | NAME | Sex | Group | ADS | DM | LinA1 | MathAn |
25 | Rachel Garcia | female | G | 2 | 5 | 3 | 4 |
bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI$ ./print_table -t my_table -c 30 -p A
# | NAME | Sex | Group | ADS | DM | LinA1 | MathAn |
bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI$ ./table_generator -t my_table -r wb -c 20 -g new_groups
bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI$ ./print_table -t my_table -c 20
# | NAME | Sex | Group | ADS | DM | LinA1 | MathAn |
1 | Ronald Brown | male | G | 4 | 4 | 5 | 5 |
2 | Daniel Marshall | male | E | 5 | 2 | 2 | 3 |
3 | Carlos Roberts | male | M | 4 | 3 | 3 | 3 |
4 | Linda Zuniga | female | B | 3 | 5 | 5 | 5 |
5 | Ana Wiggins | female | M | 4 | 2 | 2 | 3 |
6 | Ann Jones | female | D | 5 | 5 | 5 | 4 |
7 | Michelle Lewis | female | M | 3 | 3 | 2 | 3 |
8 | Wendy Tran | female | B | 3 | 3 | 4 | 4 |
9 | Lisa Parks | female | M | 4 | 2 | 5 | 3 |
10 | Lindsay Johnson | female | B | 3 | 4 | 3 | 4 |
11 | Christopher Jordan | male | C | 5 | 4 | 3 | 5 |
12 | Ebony Cooper | female | N | 5 | 4 | 2 | 5 |
13 | Kathy Valdez | female | F | 2 | 2 | 2 | 4 |
14 | Maria Dunn | female | N | 4 | 2 | 5 | 3 |
15 | Edwin Miller | male | G | 4 | 5 | 2 | 5 |
16 | Brian Doyle | male | D | 3 | 2 | 4 | 4 |
17 | Steven Lewis | male | D | 3 | 4 | 3 | 4 |
18 | Nicholas Gonzalez | male | C | 3 | 3 | 2 | 4 |
19 | Wendy Tran | female | C | 3 | 3 | 5 | 2 |
20 | Kerry Jones | female | G | 4 | 2 | 5 | 5 |
bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI$ ./print_table -t my_table -c 20 -p A
# | NAME | Sex | Group | ADS | DM | LinA1 | MathAn |
bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI$ ./print_table -t my_table -c 20 -p B
# | NAME | Sex | Group | ADS | DM | LinA1 | MathAn |
bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI$ ./print_table -t my_table -c 20 -p C
# | NAME | Sex | Group | ADS | DM | LinA1 | MathAn |
19 | Wendy Tran | female | C | 3 | 3 | 5 | 2 |
bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI$ ./print_table -t my_table -c 20 -p D
# | NAME | Sex | Group | ADS | DM | LinA1 | MathAn |
bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI$ ./print_table -t my_table -c 20 -p E
# | NAME | Sex | Group | ADS | DM | LinA1 | MathAn |
bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI$ ./print_table -t my_table -c 20 -p F
# | NAME | Sex | Group | ADS | DM | LinA1 | MathAn |
bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI$ ./print_table -t my_table -c 20 -p G
# | NAME | Sex | Group | ADS | DM | LinA1 | MathAn |
bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI$ ./print_table -t my_table -c 20 -p H
# | NAME | Sex | Group | ADS | DM | LinA1 | MathAn |
bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI$ ./print_table -t my_table -c 20 -p M
# | NAME | Sex | Group | ADS | DM | LinA1 | MathAn |
9 | Lisa Parks | female | M | 4 | 2 | 5 | 3 |
bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI$ ./print_table -t my_table -c 20 -p N
# | NAME | Sex | Group | ADS | DM | LinA1 | MathAn |
14 | Maria Dunn | female | N | 4 | 2 | 5 | 3 |
bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI$ ./print_table -t my_table -c 20 -p U
# | NAME | Sex | Group | ADS | DM | LinA1 | MathAn |
bulat@bulat-Swift-SF314-58:~/Studying/prprm/cr/VI$ exit
exit
Script done.

```

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:

11. Выводы: В ходе этой лабораторной работы я получил опыт написания программ, использующий чтение и запись из/в бинарные файлы на языке Си.

Недочёты при выполнении задания могут быть устранены следующим образом:

Подпись студента