

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра Информационных систем**

**ОТЧЕТ**  
**по практической работе №1**  
**по дисциплине «Программирование»**  
**ТЕМА: изучение и организация структур**

СТУДЕНТ ГР.0324

\_\_\_\_\_

КОШЕЛЯЕВ А.С

ПРЕПОДАВАТЕЛЬ

\_\_\_\_\_

ГЛУЩЕНКО А.Г

САНКТ-ПЕТЕРБУРГ

2021

## Цель работы.

Изучение и организация структур; получение практических навыков работы со структурами; определение преимуществ и недостатков использования структур.

## Основные теоретические положения.

Структуры представляют собой группы связанных между собой, как правило, разнотипных переменных, объединенных в единый объект, в отличие от массива, все элементы которого однотипны. В языке C++ структура является видом класса и обладает всеми его свойствами.

Для определения структуры применяется ключевое слово **struct**, а сам формат определения выглядит следующим образом:

```
struct [имя_типа] {  
    тип_1 элемент_1;  
    тип_2 элемент_2;  
    ...  
    тип_k элемент_k;  
} [ список_описателей ];
```

Каждая входящая в структуру переменная называется членом (полем, элементом) структуры и описывается типом данных и именем. Поля структуры могут быть любого типа данных. Их количество не лимитировано.

Инициализация структур аналогична инициализации массивов: в фигурных скобках передаются значения для элементов структуры по порядку. Так как в структуре `student` первым определено свойство, которое представляет тип **int** - число, то в фигурных скобках вначале идет число. И так далее для всех элементов структуры по порядку.

Все поля структурных переменных располагаются в непрерывной области памяти одно за другим. Общий объем памяти, занимаемый структурой, равен сумме размеров всех полей структуры.

Кроме базовых примитивных типов данных как **int** или **char**, массивов и указателей в качестве элементов структуры можно использовать другие структуры.

В программировании очень часто используются такие конструкции, как массивы структур. Все форматы определения массива структур будут аналогичны определению массивов других типов:

```
struct Students students[30];
```

Был определен 30-элементный массив, каждый элемент которого предназначен для хранения данных одного студента. Получение доступа к данным некоторого студента из группы *N* осуществляется обычной индексацией переменной массива.

Для того чтобы записать данные в структурную переменную, необходимо каждому полю структуры присвоить определенное значение. Для этого необходимо использовать оператор ‘.’ («точка»).

При доступе к определенному полю его следует рассматривать как обычную переменную, тип данных которой соответствует типу этого поля. Поля структур могут участвовать в качестве операндов любых выражений, допускающих использование операндов соответствующего типа данных.

Копирование данных из одной структурной переменной в другую осуществляется простой операцией присваивания, независимо от количества полей и размера структуры (это можно делать только в том случае, когда обе переменные одного и того же типа).

Любая структурная переменная занимает в памяти определенное положение, характеризующееся конкретным адресом. Для работы с адресами структурных переменных (как и для простых переменных) можно использовать указатели. Указатели на структурные переменные определяются точно так же, как и для обычных переменных. Разыменование

указателя (обращение к данным по адресу, хранящемуся в указателе) осуществляется также обычным образом.

Через указатели можно работать с отдельными полями структур. Для доступа к полю структуры через указатель используется оператор ‘->’ («стрелка»), а не «точка».

Структуры можно использовать в качестве параметров функций, как и обычные переменные. Для структур поддерживаются все три механизма передачи данных: по значению, через указатели и по ссылке.

### **Постановка задачи.**

Необходимо создать массив структур, содержащий информацию о студентах: ФИО, пол, номер группы, номер в списке группы, оценки за прошедшую сессию (всего 3 экзамена и 5 дифференцированных зачетов), форма обучения, отметка времени о внесении или изменении данных. Ввод и изменение данных обо всех студентах должен осуществляться в файл students.

Написать функции, реализующие операции со структурами (ввод данных с клавиатуры):

1. Создание новой записи о студенте.
2. Внесение изменений в уже имеющуюся запись.
3. Вывод всех данных о студентах.
4. Вывод информации обо всех студентах группы  $N$ .  $N$  – инициализируется пользователем.
5. Вывод топа самых успешных студентов с наивысшим по рейтингу средним баллом за прошедшую сессию.

6. Вывод количества студентов мужского и женского пола.
7. Определение количества студентов, которые будут получать стипендию (стипендия начисляется, если у студента нет троек и очная форма обучения).
8. Вывод данных о студентах, которые не получают стипендию; учатся только на «хорошо» и «отлично»; учатся только на «отлично»;
9. Вывод данных о студентах, имеющих номер в списке –  $k$ .
10. Вывод всех записей, сделанных в день, который введет пользователь.  
Вывод всех записей, сделанных после полудня. Вывод всех записей, сделанных до полудня.

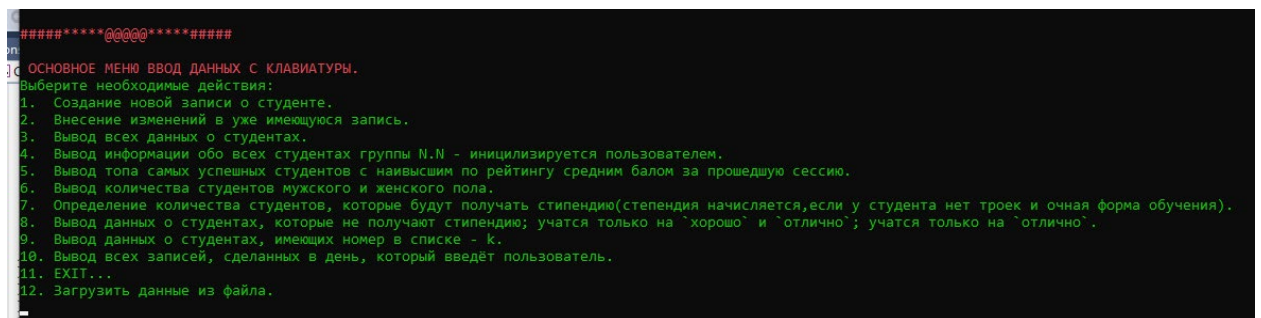
Пример тестовых данных:

Файл с текстовыми данными.

Необходимо подготовить собственный файл с тестовыми данными.

### Выполнение работы.

Выполнение работы начинается с написания пользовательского интерфейса.



```
#####*****@#####
осн. ОСНОВНОЕ МЕНЮ ВВОД ДАННЫХ С КЛАВИАТУРЫ.
Выберите необходимые действия:
1. Создание новой записи о студенте.
2. Внесение изменений в уже имеющуюся запись.
3. Вывод всех данных о студентах.
4. Вывод информации обо всех студентах группы N.N - инициализируется пользователем.
5. Вывод топa самых успешных студентов с наивысшим по рейтингу средним балом за прошедшую сессию.
6. Вывод количества студентов мужского и женского пола.
7. Определение количества студентов, которые будут получать стипендию(стипендия начисляется,если у студента нет троек и очная форма обучения).
8. Вывод данных о студентах, которые не получают стипендию; учатся только на `хорошо` и `отлично`; учатся только на `отлично`.
9. Вывод данных о студентах, имеющих номер в списке - k.
10. Вывод всех записей, сделанных в день, который введет пользователь.
11. EXIT...
12. Загрузить данные из файла.
```

Рисунок 1

Необходимо создать массив структур, содержащий информацию о студентах: ФИО, пол, номер группы, номер в списке группы, оценки за прошедшую сессию (всего 3 экзамена и 5 дифференцированных зачетов), форма обучения, отметка времени о внесении или изменении данных. Ввод и изменение данных обо всех студентах должен осуществляться в файл Text.txt

```

int kol;
struct student //создание структуры студент
{
    int num_gr; //номер группы
    int num_list; //номер в списке группы
    int ochenki[8]; // оценки
    string forma; // форма обучения
    string fio[20]; //ФИО
    char sex; // пол
    int date[40]; //дата год месяц день час мин сек
}mstudent[45];

```

Рисунок 2 Создание структуры с требуемыми полями.

Теперь создадим текстовый документ

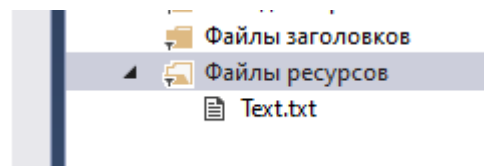


Рисунок 3 Создание текстового файла

Приступаем к выполнению 1 пункта задания.

Создание новой записи о студенте.

```

Введите количество студентов.
1
Введите номер группы.
<ios2
<iomВведите номер в списке группы.
<str
<win1
<fstВведите оценки за прошедшую сессию.
<sm (Физика История Алгебра Программирование МатАнализ Информатика Философия Экономика.
iespa(Форма ввода 5 4 4 3 5 4 5 2).

uden1

um_g Проверьте правильность данных <Оценки в диапазоне от 5-2>
um_l2
cenk5 5 5 5 5 5 5
g foВведите форму обучения(Дневная, Вечерняя, Заочная (Соблюдайте регистр!)).
g fi
sex;Заочная
ateВведите ФИО (Форма Фамилию Имя Отчество ).
[45]
int Караев Борис Борисович
int Введите пол(М,Ж).
()f
a(inM
d(inДата (Год) формат(хххх).
gr(12018
int
    Проверьте правильность данных
2021
Дата (Месяц) формат(хх).
e из: 09
Дата (День) формат(хх).
1
Дата (Час) формат(хх).
12
Дата (Мин) формат(хх).
34
Дата (Сек) формат(хх).
11_
Закл

```

Рисунок 4 Консольный интерфейс создания записи.

При создании новой записи файл дописывается.

Внесение изменений в уже имеющуюся запись.

```
2. Внесение изменений в уже имеющуюся запись.
#####!!!$$$$____%%^^^&&&***~~~~
Техданные :26
Загрузка файла Text.txt
Всё работает. Файл открыт!
1. Поиск по фамилии студента.

Введите фамилию студента соблюдая регистр по <форме Фамилия >
Караев

Вы искали: Караев Борис Борисович?
1. Да
2. Нет
1

В данный момент запись имеет данные

Номер группы: 2Номер в списке (порядковый номер) 1 ФИО: Караев Борис Борисович
Форма обучения: Заочная
ОЦЕНКИ.
Физика. 2
История. 5
Алгебра. 5
Программирование. 5
МатАнализ. 5
Информатика. 5
Философия. 5
Экономика. 5
Дата формата (год месяц день час мин сек).
Год: 2021 Месяц: 9 День: 1 Час: 12 Мин: 34 Сек: 11
Что требуется изменить? выберите варианты.
1. Номер группы:
2. ФИО:
3. Пол:
4. Номер в списке:
5. Оценки за прошедшую сессию:
6. Форму обучения:

```

Рисунок 5 Консольный интерфейс внесения изменений в существующую запись.

Вывод всех данных о студентах.

Приведена только один пример вывода данных т.к. вывод данных о всех студентах займет несколько страниц. Сам интерфейс однотипен для каждой записи.

```

ОЦЕНКИ.
Физика. 5
История. 5
Алгебра. 3
Программирование. 5
МатАнализ. 5
Информатика. 5
Философия. 5
Экономика. 5
Дата формата (год месяц день час мин сек).
Год: 2020 Месяц: 8 День: 31 Час: 9 Мин: 14 Сек: 33
Номер группы 1
Номер в списке (порядковый номер) 1
ФИО Тестов Тест Русский
Пол М
Форма обучения Дневная
ОЦЕНКИ.
Физика. 5
История. 5
Алгебра. 5
Программирование. 5
МатАнализ. 5
Информатика. 5
Философия. 5
Экономика. 5
Дата формата (год месяц день час мин сек).
Год: 2020 Месяц: 9 День: 1 Час: 9 Мин: 47 Сек: 22
Номер группы 2
Номер в списке (порядковый номер) 1
ФИО Караев Борис Борисович
Пол М
Форма обучения Дневная
ОЦЕНКИ.
Физика. 2

```

Рисунок 6 Консольный интерфейс вывода.

Вывод информации обо всех студентах группы  $N$ .  $N$  – инициализируется пользователем.



```
4. Вывод информации обо всех студентах группы N.N - инициализация
#####@!!!$$$$____%%^^^&&&***nnnn
Техданные :26
Загрузка файла Text.txt
Всё работает. Файл открыт!

Введите номер группы формат(xxxx).

324

Номер группы 324
Номер в списке (порядковый номер) 1
ФИО Балашова Мария Ивановна
Пол Ж
Форма обучения Вечерняя
ОЦЕНКИ.
Физика. 5
История. 5
Алгебра. 5
Программирование. 5
МатАнализ. 5
Информатика. 5
Философия. 5
Экономика. 5
Дата формата (год месяц день час мин сек).
Год: 2020 Месяц: 8 День: 30 Час: 11 Мин: 12 Сек: 55

Номер группы 324
Номер в списке (порядковый номер) 2
ФИО Городниченко Тимофей Кирилович
Пол М
Форма обучения Вечерняя
ОЦЕНКИ.
Физика. 4
```

Рисунок 7 Консольный интерфейс вывода.

Вывод топа самых успешных студентов с наивысшим по рейтингу средним баллом за прошедшую сессию.

```
Загрузка файла Text.txt
Всё работает. Файл открыт!

Номер группы: 321 ФИО: Агранов Алексей Анатольевич Средний балл: 5
Номер группы: 321 ФИО: Антрошенко Мария Анатольевна Средний балл: 5
Номер группы: 321 ФИО: Иванова Екатерина Ивановна Средний балл: 5
Номер группы: 321 ФИО: Феллер Владимир Иванович Средний балл: 5
Номер группы: 323 ФИО: Атлас Мария Ивановна Средний балл: 5
Номер группы: 323 ФИО: Балашевич Константин Константинович Средний балл: 5
Номер группы: 324 ФИО: Балашова Мария Ивановна Средний балл: 5
Номер группы: 324 ФИО: Жигалова Дарья Сергеевна Средний балл: 5
Номер группы: 333 ФИО: Титов Иван Иванович Средний балл: 5
Номер группы: 1 ФИО: Тестов Тест Русский Средний балл: 5
#####@!!!$$$$____%%^^^&&&***nnnn#####
ОСНОВНОЕ МЕНЮ ВВОД ДАННЫХ С КЛАВИАТУРЫ.
Выберите необходимые действия:
1. Создание новой записи о студенте.
2. Внесение изменений в уже имеющуюся запись.
3. Вывод всех данных о студентах.
```

Рисунок 8 Консольный интерфейс вывода.

Вывод количества студентов мужского и женского пола.

```
б6. Вывод количества студентов мужского и женского пола.
#####@@@!!!!$$$$____%%^^^&&&&***~
Техданные :26
Загрузка файла Text.txt
Всё работает. Файл открыт!

lude В списках групп всего : 16 студентов мужского пола
lude В списках групп всего : 10 студентов женского пола
lude #####*****@@@*****#####
lude
lude ОСНОВНОЕ МЕНЮ ВВОД ДАННЫХ С КЛАВИАТУРЫ.
lude Выберите необходимые действия:
g nam1. Создание новой записи о студенте.
kol; 2. Внесение изменений в уже имеющуюся запись.
ct st3. Вывод всех данных о студентах.
4. Вывод информации обо всех студентах группы N.N - инициализируется пользователем.
int n5. Вывод топа самых успешных студентов с наивысшим по рейтингу средним балом за прошедш
int n6. Вывод количества студентов мужского и женского пола.
int o7. Определение количества студентов, которые будут получать стипендию(стипендия начисля
string8. Вывод данных о студентах, которые не получают стипендию; учатся только на `хорошо` и
char9. Вывод данных о студентах, имеющих номер в списке - k.
int d10. Вывод всех записей, сделанных в день, который введёт пользователь.
udent11. EXIT...
info(12. Загрузить данные из файла.
shet(
vvod
izmen
```

Рисунок 9 Консольный интерфейс вывода.

Определение количества студентов, которые будут получать стипендию  
(стипендия начисляется, если у студента нет троек и очная форма обучения).

```
7. Определение количества студентов, которые будут получать стипендию(стипендия начисляется, если у студента нет троек и очная форма обучения).
#####@@@!!!!$$$$____%%^^^&&&&***~
Техданные :26
Загрузка файла Text.txt
Всё работает. Файл открыт!

Номер группы: 333 ФИО: Титов Иван Иванович Стипендия начисляется
Номер группы: 333 ФИО: Смирнова Анна Дмитриевна Стипендия начисляется
Номер группы: 1 ФИО: Тестов Тест Русский Стипендия начисляется
Общее количество студентов получающих стипендию = 3

#####*****@@@*****#####

ОСНОВНОЕ МЕНЮ ВВОД ДАННЫХ С КЛАВИАТУРЫ.
Выберите необходимые действия:
1. Создание новой записи о студенте.
2. Внесение изменений в уже имеющуюся запись.
```

Рисунок 10 Консольный интерфейс вывода.

```

номер группы: 324 ФИО: Чиж Илья Владимирович Форма обучения: Вечерняя
номер группы: 324 ФИО: Сотина Елизавета Петровна Форма обучения: Вечерняя
номер группы: 333 ФИО: Смирнова Анна Дмитриевна Форма обучения: Дневная
Студенты которые не получают стипендию <Есть тройки или ниже>
номер группы: 321 ФИО: Фурзиков Борис Фёдорович Форма обучения: Тест
номер группы: 321 ФИО: Куренков Георгий Петрович Форма обучения: Вечерняя
номер группы: 323 ФИО: Землянский Дмитрий Иванович Форма обучения: Вечерняя
номер группы: 323 ФИО: Мартынов Александр Александрович Форма обучения: Вечерняя
номер группы: 323 ФИО: Пикинер Денис Харитонович Форма обучения: Вечерняя
номер группы: 323 ФИО: Балякина Вероника Ивановна Форма обучения: Вечерняя
номер группы: 324 ФИО: Городниченко Тимофей Кирилович Форма обучения: Вечерняя
номер группы: 324 ФИО: Рысиков Роман Романович Форма обучения: Вечерняя
номер группы: 333 ФИО: Петров Петр Петрович Форма обучения: Дневная
номер группы: 2 ФИО: Караев Борис Борисович Форма обучения: Дневная

Списочный состав студентов: 26 Учатся тоько на отлично: 10 Учатся на <хорошо> и <отлично>: 6 Остальные не получают стипендию: 10
#####@@@@#####
ОСНОВНОЕ МЕНЮ ВВОД ДАННЫХ С КЛАВИАТУРЫ.

```

Вывод данных о студентах, имеющих номер в списке –  $k$ .

Рисунок 12 Консольный интерфейс вывода.

Вывод всех записей, сделанных в день, который введет пользователь. Вывод всех записей, сделанных после полудня. Вывод всех записей, сделанных до полудня.

```
10. Вывод всех записей, сделанных в день, который введёт пользователь.
#####!!!$$$$____%%^^^&&&&***~
Техданные :26
Загрузка файла Text.txt
Всё работает. Файл открыт!
Введите дату по которой начнем поиск данных <Дата (День) формат(хх)>.
31
Данные о студентах внесённые 31 числа до полудня
Номер группы: 321Номер в списке (порядковый номер) 1 ФИО: Фурзиков Борис Фёдорович
Форма обучения: Тест
ОЦЕНКИ.
Физика. 5
История. 4
Алгебра. 5
Программирование. 3
МатАнализ. 5
Информатика. 3
Философия. 3
Экономика. 4
Пол М
Дата формата (год месяц день час мин сек).
Год: 2020 Месяц: 8 День: 31 Час: 9 Мин: 13 Сек: 15
Данные о студентах внесённые 31 числа после полудня
(Номер группы: 321Номер в списке (порядковый номер) 2 ФИО: Агранов Алексей Анатольевич
Форма обучения: Вечерняя
ОЦЕНКИ.
Физика. 5
История. 5
Алгебра. 5
Программирование. 5
МатАнализ. 5
Информатика. 5
```

Рисунок 13 Консольный интерфейс вывода.

Пример тестовых данных:

В подключённом к проекту текстовом файле данные о каждом студенте имеют формат.

324	номер группы
5	порядковый номер в списке группы
5 5 5 5 5 5 4	оценки по 8 предметам
Вечерняя	форма обучения
Чиж Илья Владимирович	Фамилия Имя Отчество
М	Пол студента
2020 8 31 17 35 22	Дата формата <Годxxx> <Месяц> <День> <Час> <Мин> <Сек>

## Выводы.

Структуры полезны, когда нам надо объединить несколько переменных с разными типами под одним именем. Это делает программу более

компактной и более гибкой для внесения изменений. Также структуры незаменимы, когда необходимо сгруппировать некоторые данные, например, запись из базы данных или контакт из книги адресов. В последнем случае структура будет содержать такие данные контакта как имя, адрес, телефон и т.п.

## ПРИЛОЖЕНИЕ А

### ПОЛНЫЙ КОД ПРОГРАММЫ

```

                                                                 #include <iostream> //
стандартный
#include <iomanip> // для выравнивания setw и т.д
#include <string> // для обработки текста
#include <windows.h> // хз препод рассказал 1 курс 1 семестр 3 лаба
#include <fstream> // читать и писать файлы
#include <cmath> // для округления работет с round
using namespace std;
int kol;
struct student //создание структуры студент
{
    группы
    обучения
    день час мин сек
}mstudent[45];
int info(int a) //основное меню
{
    GetStdHandle(STD_OUTPUT_HANDLE);
    FOREGROUND_RED | FOREGROUND_INTENSITY);
    "\n#####*****@@@@@*****#####\n";
    МЕНЮ ВВОД ДАННЫХ С КЛАВИАТУРЫ.\n";

    int num_gr; //номер группы
    int num_list; //номер в списке

    int ocenki[8]; // оценки
    string forma; // форма

    string fio[20]; //ФИО
    char sex; // пол
    int date[40]; //дата год месяц

    int b = 0;
    HANDLE O =
    SetConsoleTextAttribute(O,
    cout <<
    cout << "\n ОСНОВНОЕ
```

```
    FOREGROUND_GREEN | FOREGROUND_INTENSITY);
```

```
    необходимые действия:" << "\n";
```

```
    Создание новой записи о студенте.\n";
```

```
    Внесение изменений в уже имеющуюся запись.\n";
```

```
    всех данных о студентах.\n";
```

```
    информации обо всех студентах группы N.N - инициализируется пользователем.\n";
```

```
    топа самых успешных студентов с наивысшим по рейтингу средним балом за прошедшую сессию.\n";
```

```
    количества студентов мужского и женского пола.\n";
```

```
    Определение количества студентов, которые будут получать стипендию(стипендия начисляется,если у студента нет троек и очная форма обучения).\n";
```

```
    данных о студентах, которые не получают стипендию; учатся только на `хорошо` и `отлично`; учатся только на `отлично`.\n";
```

```
    данных о студентах, имеющих номер в списке - k.\n";
```

```
    всех записей, сделанных в день, который введёт пользователь.\n";
```

```
    EXIT...\n";
```

```
    Загрузить данные из файла.\n";
```

```
}
```

```
int shet(int ras) //функция подсчёта строк в файле Text.txt
```

```
{
```

```
    SetConsoleTextAttribute(O,
```

```
    cout << setw(4) << "Выберите
```

```
    cout << setw(4) << "1.
```

```
    cout << setw(4) << "2.
```

```
    cout << setw(4) << "3. Вывод
```

```
    cout << setw(4) << "4. Вывод
```

```
    cout << setw(4) << "5. Вывод
```

```
    cout << setw(4) << "6. Вывод
```

```
    cout << setw(4) << "7.
```

```
    cout << setw(4) << "8. Вывод
```

```
    cout << setw(4) << "9. Вывод
```

```
    cout << setw(4) << "10. Вывод
```

```
    cout << setw(4) << "11.
```

```
    cout << setw(4) << "12.
```

```
    cin >> b;
```

```
    cin.sync();
```

```
    system("cls");
```

```
    return b;
```

```
    ifstream file("Text.txt");
```

```
    if (!file)
```

```
    {
```

```
        cout << " Error!!! \n";
```

```
    }
```

```
    else
```

```
    {
```

```
        int ras = 1;
```

```
        while (true)
```

```
        {
```

```
            string v;
```

```
            getline(file, v);
```

```
            if (!file.eof())
```

```
                ras++;
```

```

        else
            break;
    }
    kol = ras / 7;
    cin.get();
}
file.close();
cout << "\n Техданные : " <<

kol;

return kol;
}

void vvod() //создание новой записи о студенте. Геморойная функция замучился
допиливать
{
    SetConsoleCP(1251); //
    SetConsoleOutputCP(1251);
    // это тоже для русских букв тестил на винде без русского это не работает
    cout << "\nВведите
    количество студентов.\n" << endl;

    /*int nu = dva + 1;*/
    int za;
    cin >> za;
    /*da = nu + za;*/
    for (int i = 0; i < za; i++) {
        cout << "Введите номер
        группы.\n" << endl;

        cin >> mstudent[i].num_gr;
        cout << "Введите номер в
        списке группы.\n" << endl;

        cin >> mstudent[i].num_list;
        cout << "Введите оценки за
        прошедшую сессию.\n (Физика История Алгебра Программирование МатАнализ
        Информатика Философия Экономика.\n" << "(Форма ввода 5 4 4 3 5 4 5 2).\n" << endl;
        int raz;
        cin >> raz;
        cin.sync();
        while (raz > 5 || raz < 2)
        {
            cout << "\n Проверьте
            правильность данных <Оценки в диапазоне от 5-2> \n";

            cin >> raz;
            cin.sync();
            continue;
        }
        mstudent[i].ocenki[0] = raz;
        int dva;
        cin >> dva;
        cin.sync();
        while (dva > 5 || dva < 2)
        {

```



правильность данных <Оценки в диапазоне от 5-2> \n";

правильность данных <Оценки в диапазоне от 5-2> \n";

правильность данных <Оценки в диапазоне от 5-2> \n";

правильность данных <Оценки в диапазоне от 5-2> \n";

правильность данных <Оценки в диапазоне от 5-2> \n";

```
cout << "\n Проверьте

cin >> dva;
cin.sync();
continue;
}
mstudent[i].ocenki[1] = dva;
int tri;
cin >> tri;
cin.sync();
while (tri > 5 || tri < 2)
{
    cout << "\n Проверьте

    cin >> tri;
    cin.sync();
    continue;
}
mstudent[i].ocenki[2] = tri;
int che;
cin >> che;
cin.sync();
while (che > 5 || che < 2)
{
    cout << "\n Проверьте

    cin >> che;
    cin.sync();
    continue;
}
mstudent[i].ocenki[3] = che;
int piy;
cin >> piy;
cin.sync();
while (piy > 5 || piy < 2)
{
    cout << "\n Проверьте

    cin >> piy;
    cin.sync();
    continue;
}
mstudent[i].ocenki[4] = piy;
int six;
cin >> six;
cin.sync();
while (six > 5 || six < 2)
{
    cout << "\n Проверьте

    cin >> six;
    cin.sync();
```



правильность данных <Оценки в диапазоне от 5-2> \n";

правильность данных <Оценки в диапазоне от 5-2> \n";

обучения(Дневная, Вечерняя, Заочная (Соблюдайте регистр!)).\n" << endl;

(Форма Фамилию Имя Отчество ).\n" << endl;

mstudent[i].fio[j];

пол(М,Ж).\n" << endl;

формат(xxxx).\n";

2021 || mstudent[i].date[0] < 2020)

правильность данных \n";

mstudent[i].date[0];

```
        continue;
    }
    mstudent[i].ocenki[5] = six;
    int sem;
    cin >> sem;
    cin.sync();
    while (sem > 5 || sem < 2)
    {
        cout << "\n Проверьте

        cin >> sem;
        cin.sync();
        continue;
    }
    mstudent[i].ocenki[6] = sem;
    int eil;
    cin >> eil;
    cin.sync();
    while (eil > 5 || eil < 2)
    {
        cout << "\n Проверьте

        cin >> eil;
        cin.sync();
        continue;
    }
    mstudent[i].ocenki[7] = eil;
```

```
cout << "Введите форму
обучения(Дневная, Вечерняя, Заочная (Соблюдайте регистр!)).\n" << endl;
```

```
cin >> mstudent[i].forma;
cout << "Введите ФИО
```

```
for (int j = 0; j < 3; ++j) {
    cin >>
```

```
}
cout << "Введите
```

```
cin >> mstudent[i].sex;
cout << "Дата (Год)
```

```
cin >> mstudent[i].date[0];
cin.sync();
while (mstudent[i].date[0] >
```

```
{
    cout << "\n Проверьте

    cin >>
```

формат(xx).\n";

12 || mstudent[i].date[1] < 1)

правильность данных \n";

mstudent[i].date[1];

формат(xx).\n";

31 || mstudent[i].date[2] < 0)

правильность данных \n";

mstudent[i].date[2];

формат(xx).\n";

23 || mstudent[i].date[3] < 0)

правильность данных \n";

mstudent[i].date[3];

формат(xx).\n";

59 || mstudent[i].date[4] < 0)

```
cin.sync();
continue;
}
cout << "Дата (Месяц)

cin >> mstudent[i].date[1];
cin.sync();
while (mstudent[i].date[1] >

{
    cout << "\n Проверьте

    cin >>

    cin.sync();
    continue;
}
cout << "Дата (День)

cin >> mstudent[i].date[2];
cin.sync();
while (mstudent[i].date[2] >

{
    cout << "\n Проверьте

    cin >>

    cin.sync();
    continue;
}
cout << "Дата (Час)

cin >> mstudent[i].date[3];
cin.sync();
while (mstudent[i].date[3] >

{
    cout << "\n Проверьте

    cin >>

    cin.sync();
    continue;
}
cout << "Дата (Мин)

cin >> mstudent[i].date[4];
cin.sync();
while (mstudent[i].date[4] >

{
```

правильность данных \n";

mstudent[i].date[4];

формат(xx).\n";

59 || mstudent[i].date[5] < 0)

правильность данных \n";

mstudent[i].date[5];

открыт!!\n";

открыт в режиме дозаписи!\n";

mstudent[i].ocenki[j] << " ";

<< " ";

cout << "\n Проверьте

cin >>

cin.sync();

continue;

}

cout << "Дата (Сек)

cin >> mstudent[i].date[5];

cin.sync();

while (mstudent[i].date[5] >

{

cout << "\n Проверьте

cin >>

cin.sync();

continue;

}

}

ofstream d;

d.open("Text.txt", ios::app);

if (!d)

{

cout << "Файл не

}

else

{

cout << "Всё работает. Файл

}

for (int i = 0; i < za; i++) {

d << "\n";

d << mstudent[i].num\_gr;

d << "\n";

d << mstudent[i].num\_list;

d << "\n";

for (int j = 0; j < 8; ++j) {

d <<

}

d << "\n";

d << mstudent[i].forma;

d << "\n";

for (int j = 0; j < 3; ++j) {

d << mstudent[i].fio[j]

}

d << "\n";

```

mstudent[i].date[j] << " ";

}
int izmena(int kol) // изменение в имеющуюся запись
{
    cout << " 1. Поиск по
    фамилии студента." << endl;

    SetConsoleCP(1251); //
    SetConsoleOutputCP(1251);
    // это тоже для русских букв тестил на винде без русского это не работает
    X2:
    cout << "\n Введите фамилию
    студента соблюдая регистр по <форме Фамилия >" << endl;
    string x;
    cin >> x;
    cin.sync();
    int uff = 0;
    for (int i = 0; i < kol; i++) {
        if (mstudent[i].fio[0] == x) {
            cout << "\nВы искали:
            " << mstudent[i].fio[0] << " " << mstudent[i].fio[1] << " " << mstudent[i].fio[2] << "?" << endl;
            cout << "1. Да" <<
            endl;
            cout << "2. Нет" <<
            endl;

            int nem;
            cin >> nem;
            cin.sync();
            while (nem > 2 || nem
            < 1)
            {
                cout << "\n
                Проверьте правильность данных <Возможен ответ 1.да или 2.нет \n";
                cin >> nem;
                cin.sync();
                continue;
            }
            if (nem == 2) {
                goto X2;
            }
            if (nem == 1) {
                cout << "\nВ
                данный момент запись имеет данные" << endl;

```

```

cout <<
"\nНомер группы: " << mstudent[i].num_gr << "Номер в списке (порядковый номер) " <<
mstudent[i].num_list << " ФИО: " << mstudent[i].fio[0] << " " << mstudent[i].fio[1] << " " <<
mstudent[i].fio[2] << endl;

cout <<

"Форма обучения: " << mstudent[i].forma << endl;

cout << "
ОЦЕНКИ.\n" << "Физика. "
<< mstudent[i].ocenki[0] << "\n" << "История. " << mstudent[i].ocenki[1] << "\n" <<
"Алгебра. " << mstudent[i].ocenki[2] << endl;

cout <<
"Программирование. " << mstudent[i].ocenki[3] << "\n" << "МатАнализ. " <<
mstudent[i].ocenki[4] << "\n" << "Информатика. " << mstudent[i].ocenki[5] << endl;

cout <<
"Философия. " << mstudent[i].ocenki[6] << "\n" << "Экономика. " << mstudent[i].ocenki[7]
<< endl;

cout << "Дата
формата (год месяц день час мин сек).\n " << "Год: " << mstudent[i].date[0] << " Месяц: "
<< mstudent[i].date[1] << " День: " << mstudent[i].date[2];

cout << " Час: "
<< mstudent[i].date[3] << " Мин: " << mstudent[i].date[4] << " Сек: " << mstudent[i].date[5]
<< endl;

cout << "Что
требуется изменить? выберите варианты." << endl;

cout << "1.

Номер группы:" << endl;

cout << "2.

ФИО:" << endl;

cout << "3.

Пол:" << endl;

cout << "4.

Номер в списке:" << endl;

cout << "5.

Оценки за прошедшую сессию:" << endl;

cout << "6.

Форму обучения:" << endl;

int izi;
cin >> izi;
cin.sync();
while (izi > 6 ||

izi < 1)

{
cout <<

"\n Проверьте правильность данных <Ввидите цифру варианта 1-6> \n";

cin >>

cin.sync();

continue;

}
if (izi == 1) {

```

```

cout <<
"\nМеняем номер группы" << endl;

cout <<
"\nВведите номер группы формат(хххх).\n" << "Диапазон возможных групп от 1-9999" <<
endl;

int xzi =

0;

cin >>

xzi;

cin.sync();

while

(xzi > 9999 || xzi < 1)

{

cout << "\n Проверьте

правильность данных таких групп не существует \n";

cin >> xzi;

cin.sync();

continue;

}

mstudent[i].num_gr = xzi;
cout <<

"\nВнесены изменения теперь номер группы " << mstudent[i].num_gr << endl;

ofstream

d;

d.open("Text.txt", ios::out);
if (!d)
{

cout << "Файл не

открыт!!!\n";

}
else
{

cout << "Всё работает. Файл

открыт в режиме полной перезаписи данных!\n";

}

for (int i

= 0; i < kol; i++) {

d

<< "\n";

d

<< mstudent[i].num_gr;

d

<< "\n";

```

```

<< mstudent[i].num_list;
<< "\n";

for (int j = 0; j < 8; ++j) {
    d << mstudent[i].ocenki[j] << "
";
}
<< "\n";
<< mstudent[i].forma;
<< "\n";

for (int j = 0; j < 3; ++j) {
    d << mstudent[i].fio[j] << " ";
}
<< "\n";
<< mstudent[i].sex;
<< "\n";

for (int j = 0; j < 6; ++j) {
    d << mstudent[i].date[j] << " ";
}
d.close();
return 0;
}
if (izi == 2) {
    cout <<
"\nФамилию Имя Отчество" << endl;
    cout <<
"Введите ФИО (Форма Фамилию Имя Отчество ).\n" << endl;
    for (int j
= 0; j < 3; ++j) {
        cin >> mstudent[i].fio[j];
    }
    cout <<
"\nВнесены изменения теперь ФИО " << mstudent[i].fio[0] << " " << mstudent[i].fio[1] << " "
<< mstudent[i].fio[2] << endl;
ofstream
d;

```

```
открыт!!!\n";
```

```
открыт в режиме полной перезаписи данных!\n";
```

```
= 0; i < kol; i++) {
```

```
<< "\n";
```

```
<< mstudent[i].num_gr;
```

```
<< "\n";
```

```
<< mstudent[i].num_list;
```

```
<< "\n";
```

```
";
```

```
<< "\n";
```

```
<< mstudent[i].forma;
```

```
<< "\n";
```

```
<< "\n";
```

```
<< mstudent[i].sex;
```

```
<< "\n";
```

```
d.open("Text.txt", ios::out);  
    if (!d)  
    {
```

```
cout << "Файл не
```

```
    }  
    else  
    {
```

```
cout << "Всё работает. Файл
```

```
    }  
    for (int i
```

```
        d
```

```
        d
```

```
        d
```

```
        d
```

```
        d
```

```
for (int j = 0; j < 8; ++j) {
```

```
d << mstudent[i].ocenki[j] << "
```

```
    }  
    d
```

```
    d
```

```
    d
```

```
for (int j = 0; j < 3; ++j) {
```

```
d << mstudent[i].fio[j] << " ";
```

```
    }  
    d
```

```
    d
```

```
    d
```

```
for (int j = 0; j < 6; ++j) {
```



```

d << mstudent[i].date[j] << " ";
    }
}

d.close();

return 0;
}
if (izi == 3) {
    cout <<

    cout <<

    char xzi;
    cin >>

    cin.sync();

    mstudent[i].sex = xzi;
    cout <<

    ofstream
d;

d.open("Text.txt", ios::out);
    if (!d)
    {

    cout << "Файл не

    }
    else
    {

    cout << "Всё работает. Файл

    }
    for (int i

= 0; i < kol; i++) {

d
    << "\n";

d
    << mstudent[i].num_gr;

d
    << "\n";

d
    << mstudent[i].num_list;

d
    << "\n";

    for (int j = 0; j < 8; ++j) {

```

```

d << mstudent[i].ocenki[j] << "
}
d
<< "\n";
d
<< mstudent[i].forma;
d
<< "\n";

for (int j = 0; j < 3; ++j) {
d << mstudent[i].fio[j] << " ";
}
d
<< "\n";
<< mstudent[i].sex;
d
<< "\n";

for (int j = 0; j < 6; ++j) {
d << mstudent[i].date[j] << " ";
}
}

d.close();
return 0;
}
if (izi == 4) {
cout <<
"Введите номер в списке группы. <формат(хх) в диапазоне от 1 до 7>\n" << endl;
int uff;
cin >>
uff;

cin.sync();
while
(uff > 7 || uff < 1)
{
cout << "\n Проверьте
правильность данных студента с таким номером в списке пока не существует \n";
cin >> uff;
cin.sync();
continue;
}

```

```

mstudent[i].num_list = uff;
cout <<
"\nВнесены изменения номер студента в списке группы " << mstudent[i].num_list << endl;
ofstream
d;

d.open("Text.txt", ios::out);
if (!d)
{

cout << "Файл не

открыт!!!\n";

}
else
{

cout << "Всё работает. Файл

открыт в режиме полной перезаписи данных!\n";

}
for (int i
= 0; i < kol; i++) {

d
<< "\n";

d
<< mstudent[i].num_gr;

d
<< "\n";

d
<< mstudent[i].num_list;

d
<< "\n";

for (int j = 0; j < 8; ++j) {

d << mstudent[i].ocenki[j] << "

}
d
<< "\n";

d
<< mstudent[i].forma;

d
<< "\n";

for (int j = 0; j < 3; ++j) {

d << mstudent[i].fio[j] << " ";

}
d
<< "\n";

```

```
<< mstudent[i].sex;
```

d

```
<< "\n";
```

d

```
for (int j = 0; j < 6; ++j) {
```

```
d << mstudent[i].date[j] << " ";  
    }  
}
```

```
d.close();
```

```
return 0;
```

```
    }  
    if (izi == 5) {  
        cout <<
```

```
"Введите оценки за прошедшую сессию.\n (Физика История Алгебра Программирование  
МатАнализ Информатика Философия Экономика.\n" << "(Форма ввода 5 4 4 3 5 4 5 2).\n"  
<< endl;
```

```
int raz;  
cin >>
```

```
raz;
```

```
cin.sync();
```

```
while
```

```
(raz > 5 || raz < 2)
```

```
{
```

```
правильность данных <Оценки в диапазоне от 5-2> \n";
```

```
cout << "\n Проверьте
```

```
cin >> raz;
```

```
cin.sync();
```

```
continue;
```

```
}
```

```
mstudent[i].ocenki[0] = raz;  
int dva;  
cin >>
```

```
dva;
```

```
cin.sync();
```

```
while
```

```
(dva > 5 || dva < 2)
```

```
{
```

```
правильность данных <Оценки в диапазоне от 5-2> \n";
```

```
cout << "\n Проверьте
```

```
cin >> dva;
```

tri;

(tri > 5 || tri < 2)

правильность данных <Оценки в диапазоне от 5-2> \n";

che;

(che > 5 || che < 2)

правильность данных <Оценки в диапазоне от 5-2> \n";

piy;

cin.sync();

continue;

}

mstudent[i].ocenki[1] = dva;  
int tri;  
cin >>

cin.sync();

while

{

cout << "\n Проверьте

cin >> tri;

cin.sync();

continue;

}

mstudent[i].ocenki[2] = tri;  
int che;  
cin >>

cin.sync();

while

{

cout << "\n Проверьте

cin >> che;

cin.sync();

continue;

}

mstudent[i].ocenki[3] = che;  
int piy;  
cin >>

cin.sync();

(piy > 5 || piy < 2)

правильность данных <Оценки в диапазоне от 5-2> \n";

six;

(six > 5 || six < 2)

правильность данных <Оценки в диапазоне от 5-2> \n";

sem;

(sem > 5 || sem < 2)

правильность данных <Оценки в диапазоне от 5-2> \n";

```
while
{
    cout << "\n Проверьте
    cin >> piy;
    cin.sync();
    continue;
}
mstudent[i].ocenki[4] = piy;
    int six;
    cin >>

    cin.sync();
        while
        {
            cout << "\n Проверьте
            cin >> six;
            cin.sync();
            continue;
        }
mstudent[i].ocenki[5] = six;
    int sem;
    cin >>

    cin.sync();
        while
        {
            cout << "\n Проверьте
            cin >> sem;
            cin.sync();
            continue;
```

```

    }

    mstudent[i].ocenki[6] = sem;
    int eil;
    cin >>

eil;

    cin.sync();

    while

    {

    cout << "\n Проверьте

правильность данных <Оценки в диапазоне от 5-2> \n";

    cin >> eil;

    cin.sync();

    continue;

    }

    mstudent[i].ocenki[7] = eil;
    cout <<
    "\nВнесены изменения у студента " << mstudent[i].fio[0] << " " << mstudent[i].fio[1] << " "
    << mstudent[i].fio[2] << endl;

    cout <<
    setw(10) << "                ОЦЕНКИ.\n" << "Физика. "
    << mstudent[i].ocenki[0] << "\n" << "История. " << mstudent[i].ocenki[1] << "\n" <<
    "Алгебра. " << mstudent[i].ocenki[2] << endl;

    cout <<
    setw(10) << "Программирование. " << mstudent[i].ocenki[3] << "\n" << "МатАнализ. " <<
    mstudent[i].ocenki[4] << "\n" << "Информатика. " << mstudent[i].ocenki[5] << endl;

    cout <<
    setw(10) << "Философия. " << mstudent[i].ocenki[6] << "\n" << "Экономика. " <<
    mstudent[i].ocenki[7] << endl;

    ofstream

d;

    d.open("Text.txt", ios::out);
    if (!d)
    {

    cout << "Файл не

открыт!!!\n";

    }
    else
    {

    cout << "Всё работает. Файл

открыт в режиме полной перезаписи данных!\n";

    }

```

```

= 0; i < kol; i++) {
    << "\n";
    << mstudent[i].num_gr;
    << "\n";
    << mstudent[i].num_list;
    << "\n";

    for (int j = 0; j < 8; ++j) {
        d << mstudent[i].ocenki[j] << "
    }
    << "\n";
    << mstudent[i].forma;
    << "\n";

    for (int j = 0; j < 3; ++j) {
        d << mstudent[i].fio[j] << " ";
    }
    << "\n";
    << mstudent[i].sex;
    << "\n";

    for (int j = 0; j < 6; ++j) {
        d << mstudent[i].date[j] << " ";
    }

    d.close();

    return 0;
}
if (izi == 6) {
    cout <<
    string
    string
    "Введите форму обучения(Дневная, Вечерняя, Заочная (Соблюдайте регистр!)).\n" << endl;
    xzi = "Дневная";
    xzy = "Вечерняя";

```



xzz = "Заочная";	string
six;	string
six;	cin >>
	cin.sync();
(six == xzi    six == xzy    six == xzz)	while
	{
правильность данных таких форм обучения не существует\n";	cout << "\n Проверьте
	cin >> six;
	cin.sync();
	continue;
	}
	mstudent[i].forma = six;
"\nВнесены изменения теперь студент " << mstudent[i].fio[0] << " " << mstudent[i].fio[1] <<	cout <<
" " << mstudent[i].fio[2] << endl;	
	cout <<
"Имеет форму обучения " << mstudent[i].forma << endl;	
d;	ofstream
	d.open("Text.txt", ios::out);
	if (!d)
	{
открыт!!!\n";	cout << "Файл не
	}
	else
	{
открыт в режиме полной перезаписи данных!\n";	cout << "Всё работает. Файл
	}
= 0; i < kol; i++) {	for (int i
<< "\n";	d
<< mstudent[i].num_gr;	d
<< "\n";	d

<< mstudent[i].num_list;	d
<< "\n";	d
"; << "\n"; << mstudent[i].forma; << "\n";	for (int j = 0; j < 8; ++j) { d << mstudent[i].ocenki[j] << " } d d d
<< "\n"; << mstudent[i].sex; << "\n";	for (int j = 0; j < 3; ++j) { d << mstudent[i].fio[j] << " "; } d d d
	for (int j = 0; j < 6; ++j) { d << mstudent[i].date[j] << " "; } } d.close(); return 0;
сожалению студента " << x << " нет в списках " << endl; } void vivod(int kol) //функция вывода         {	} } else { uff++; } if (uff == kol) { cout << "\nK } } return 0; for (int i = 0; i < kol; i++)

```

<< mstudent[i].num_gr << endl;

(порядковый номер) " << mstudent[i].num_list << endl;

mstudent[i].fio[0] << " " << mstudent[i].fio[1] << " " << mstudent[i].fio[2] << endl;
mstudent[i].sex << endl;

<< mstudent[i].forma << endl;

<< mstudent[i].ocenki[0] << "\n" << "История. " << mstudent[i].ocenki[1] << "\n" <<
"Алгебра. " << mstudent[i].ocenki[2] << endl;

"Программирование. " << mstudent[i].ocenki[3] << "\n" << "МатАнализ. " <<
mstudent[i].ocenki[4] << "\n" << "Информатика. " << mstudent[i].ocenki[5] << endl;

"Философия. " << mstudent[i].ocenki[6] << "\n" << "Экономика. " << mstudent[i].ocenki[7]
<< endl;

формата (год месяц день час мин сек).\n " << "Год: " << mstudent[i].date[0] << " Месяц: "
<< mstudent[i].date[1] << " День: " << mstudent[i].date[2];

mstudent[i].date[3] << " Мин: " << mstudent[i].date[4] << " Сек: " << mstudent[i].date[5];

}

void num_gr(int kol) //поиск по группе
{
    группы формат(хххх).\n" << endl;

    правильность данных таких групп не существует \n";

{
    {
        cout << "\nНомер группы "

        cout << "Номер в списке

        cout << "ФИО " <<

        cout << "Пол " <<

        cout << "Форма обучения "

        cout << setw(10) << "
        ОЦЕНКИ.\n" << "Физика. "

        cout << setw(10) <<

        "Программирование. " << mstudent[i].ocenki[3] << "\n" << "МатАнализ. " <<
        mstudent[i].ocenki[4] << "\n" << "Информатика. " << mstudent[i].ocenki[5] << endl;
        cout << setw(10) <<

        "Философия. " << mstudent[i].ocenki[6] << "\n" << "Экономика. " << mstudent[i].ocenki[7]
        << endl;

        cout << setw(10) << "Дата
        формата (год месяц день час мин сек).\n " << "Год: " << mstudent[i].date[0] << " Месяц: "
        << mstudent[i].date[1] << " День: " << mstudent[i].date[2];

        cout << " Час: " <<

        mstudent[i].date[3] << " Мин: " << mstudent[i].date[4] << " Сек: " << mstudent[i].date[5];
        }

        cout << "\nВведите номер

        int xz = 0;
        cin >> xz;
        cin.sync();
        while (xz > 9999 || xz < 1)
        {
            cout << "\n Проверьте

            cin >> xz;
            cin.sync();
            continue;
        }
        int q = 0;
        for (int i = 0; i < kol; i++)
        {
            if (mstudent[i].num_gr != xz)

                q++;
        }
        else
        {

```

```

        cout << "\nНомер
группы " << mstudent[i].num_gr << endl;
        cout << "Номер в
списке (порядковый номер) " << mstudent[i].num_list << endl;
        cout << "ФИО " <<
mstudent[i].fio[0] << " " << mstudent[i].fio[1] << " " << mstudent[i].fio[2] << endl;
        cout << "Пол " <<
mstudent[i].sex << endl;
        cout << "Форма
обучения " << mstudent[i].forma << endl;
        cout << setw(10) << "
ОЦЕНКИ.\n" << "Физика. "
<< mstudent[i].ocenki[0] << "\n" << "История. " << mstudent[i].ocenki[1] << "\n" <<
"Алгебра. " << mstudent[i].ocenki[2] << endl;
        cout << setw(10) <<
"Программирование. " << mstudent[i].ocenki[3] << "\n" << "МатАнализ. " <<
mstudent[i].ocenki[4] << "\n" << "Информатика. " << mstudent[i].ocenki[5] << endl;
        cout << setw(10) <<
"Философия. " << mstudent[i].ocenki[6] << "\n" << "Экономика. " << mstudent[i].ocenki[7]
<< endl;
        cout << setw(10) <<
"Дата формата (год месяц день час мин сек).\n " << "Год: " << mstudent[i].date[0] << "
Месяц: " << mstudent[i].date[1] << " День: " << mstudent[i].date[2];
        cout << " Час: " <<
mstudent[i].date[3] << " Мин: " << mstudent[i].date[4] << " Сек: " << mstudent[i].date[5] <<
endl;
    }
}
if (kol == q) {
    cout << "Записей о таких
группах пока нет.\n";
}
}
void sex(int kol) // М&Ж
{
    int a = 0, b = 0;
    for (int i = 0; i < kol; i++) {
        if (mstudent[i].sex ==
            a++;
        }
        else {
            b++;
        }
    }
    cout << "\n В списках групп
    cout << "\n В списках групп
всего : " << a << " студентов мужского пола";
всего : " << b << " студентов женского пола";
}
void topchik(int kol) //топ студентов с Наивысшим по рейтингу среднему баллом

```

```

{
    float Xmax[360];
    for (int i = 0; i < kol; i++) {
        Xmax[i] =
((mstudent[i].ocenki[0] + mstudent[i].ocenki[1] + mstudent[i].ocenki[2] + mstudent[i].ocenki[3]
+ mstudent[i].ocenki[4] + mstudent[i].ocenki[5] + mstudent[i].ocenki[6] +
mstudent[i].ocenki[7]) / 8);
    }
    float Ymax = Xmax[0];
    for (int i = 0; i < kol; i++) {
        if (Xmax[i] > Ymax) {
            Ymax = Xmax[i];
        }
    }
    for (int i = 0; i < kol; i++) {
        if (Xmax[i] == Ymax) {
            cout << "\nНомер
группы: " << mstudent[i].num_gr << " ФИО: " << mstudent[i].fio[0] << " " <<
mstudent[i].fio[1] << " " << mstudent[i].fio[2] << " Средний балл: " << round(Xmax[i] * 100) /
100 << endl;
        }
    }
}

void stipuxa(int kol) // стипендия 1 нет троек 2 очная форма (у меня Дневная)
{
    string x = "Дневная";
    int qwer = 0;
    for (int i = 0; i < kol; i++) {
        if (mstudent[i].forma == x) {
            int x = 3, y = 0;
            for (int j = 0; j < 8;
++j) {
                if
(mstudent[i].ocenki[j] > x) {
                    y++;
                }
            }
            if (y == 8) {
                qwer++;
            }
        }
    }
    HANDLE Q =
GetStdHandle(STD_OUTPUT_HANDLE);
    SetConsoleTextAttribute(Q,
    FOREGROUND_BLUE | FOREGROUND_INTENSITY);
    cout <<
"\nНомер группы: " << mstudent[i].num_gr << " ФИО: " << mstudent[i].fio[0] << " " <<
mstudent[i].fio[1] << " " << mstudent[i].fio[2] << " Стипендия начисляется " << endl;
    }
}
}
}
}

```

```

студентов получающих стипендию = " << qwer << endl;
}
void netu(int kol) //8 пункт нечего сложного но повозится пришлось
{
    HANDLE J =
    SetConsoleTextAttribute(J,
    FOREGROUND_INTENSITY | FOREGROUND_INTENSITY);
    cout << "Учатся только на
    ОТЛИЧНО" << endl;

    float Xmax[360];
    for (int i = 0; i < kol; i++) {
        Xmax[i] =
        ((mstudent[i].ocenki[0] + mstudent[i].ocenki[1] + mstudent[i].ocenki[2] + mstudent[i].ocenki[3]
        + mstudent[i].ocenki[4] + mstudent[i].ocenki[5] + mstudent[i].ocenki[6] +
        mstudent[i].ocenki[7]) / 8);
    }
    float Ymax = Xmax[0];
    for (int i = 0; i < kol; i++) {
        if (Xmax[i] > Ymax) {
            Ymax = Xmax[i];
        }
    }
    int otl = 0;
    for (int i = 0; i < kol; i++) {
        if (Xmax[i] == Ymax) {
            otl++;
            cout << "\nНомер
            группы: " << mstudent[i].num_gr << " ФИО: " << mstudent[i].fio[0] << " " <<
            mstudent[i].fio[1] << " " << mstudent[i].fio[2] << " Форма обучения: " << mstudent[i].forma
            << endl;
        }
    }
    cout << "Учатся на <хорошо>
    и <отлично>" << endl; // запутался в циклах =(

    int x = 3, y = 5, v = 0;
    for (int i = 0; i < kol; i++) {
        int z = 0, w = 0;
        for (int j = 0; j < 8; ++j) {
            if
            (mstudent[i].ocenki[j] > x) {
                z++;
            }
            if (z == 8)
            {
                for (int j = 0; j <
                8; ++j) {
                    if
                    (mstudent[i].ocenki[j] != y) {

```

```

w++;
    }
}
}
}
if (w >= 1) {
    v++;
    cout << "\nНомер
группы: " << mstudent[i].num_gr << " ФИО: " << mstudent[i].fio[0] << " " <<
mstudent[i].fio[1] << " " << mstudent[i].fio[2] << " Форма обучения: " << mstudent[i].forma
<< endl;

}
}
HANDLE A =
GetStdHandle(STD_OUTPUT_HANDLE);
SetConsoleTextAttribute(A,
FOREGROUND_GREEN | FOREGROUND_INTENSITY);
cout << "\n Студенты которые
не получают стипендию <Есть тройки или ниже>" << endl;

int tri = 0;
for (int i = 0; i < kol; i++) {
    int x = 3, y = 0;
    for (int j = 0; j < 8; ++j) {
        if
(mstudent[i].ocenki[j] <= x) {

            y++;
        }
    }
    if (y > 0) {
        tri++;
        cout << "\nНомер
группы: " << mstudent[i].num_gr << " ФИО: " << mstudent[i].fio[0] << " " <<
mstudent[i].fio[1] << " " << mstudent[i].fio[2] << " Форма обучения: " << mstudent[i].forma
<< endl;

    }
}
cout << "\n Списочный состав
студентов: " << kol << " Учатся тоько на отлично: " << otl << " Учатся на <хорошо> и
<отлично>: " << v << " Остальные не получают стипендию: " << tri << endl;
}
void list(int kol) // поиск по номеру <у меня номер в списке не сквозной это усложняет
задачу>
{
    cout << "\nВведите номер
группы формат(хххх).\n" << endl;

    int xz = 0;
    cin >> xz;
    cin.sync();
    while (xz > 9999 || xz < 0)
    {

```

правильность данных таких групп не существует \n";

{

группы: " << xz << endl;

номер в списке группы. <формат(хх) в диапазоне от 1 до 7>\n" << endl;

1)

Проверьте правильность данных студента с таким номером в списке пока не существует \n";

(mstudent[i].num\_list == uff) {

cout << "\nНомер группы: " << mstudent[i].num\_gr << "Номер в списке (порядковый номер) " << mstudent[i].num\_list << " ФИО: " << mstudent[i].fio[0] << " " << mstudent[i].fio[1] << " " << mstudent[i].fio[2] << endl;

cout << "Форма обучения: " << mstudent[i].forma << endl;

cout << "ОЦЕНКИ.\n" << "Физика. " << mstudent[i].ocenki[0] << "\n" << "История. " << mstudent[i].ocenki[1] << "\n" << "Алгебра. " << mstudent[i].ocenki[2] << endl;

cout << "Программирование. " << mstudent[i].ocenki[3] << "\n" << "МатАнализ. " << mstudent[i].ocenki[4] << "\n" << "Информатика. " << mstudent[i].ocenki[5] << endl;

cout << "Философия. " << mstudent[i].ocenki[6] << "\n" << "Экономика. " << mstudent[i].ocenki[7] << endl;

cout << "Дата формата (год месяц день час мин сек).\n " << "Год: " << mstudent[i].date[0] << " Месяц: " << mstudent[i].date[1] << " День: " << mstudent[i].date[2];

cout << " Час: " << mstudent[i].date[3] << " Мин: " << mstudent[i].date[4] << " Сек: " << mstudent[i].date[5];  
goto kostili;

cout << "\n Проверьте

cin >> xz;  
cin.sync();  
continue;

}  
int q = 0, kostili = 0;  
for (int i = 0; i < kol; i++)  
{  
if (mstudent[i].num\_gr == xz)

cout << "\n Номер

cout << "Ввидите

int uff;  
cin >> uff;  
cin.sync();  
while (uff > 7 || uff <

{

cout << "\n

cin >> uff;  
cin.sync();  
continue;

}

if



```

    }
    else {
        kostili = i;
    }
    for (kostili; kostili <
kostili + 7, kostili <= kol; kostili++)

    {
        if
(mstudent[kostili].num_list == uff) {
cout <<
"\nНомер группы: " << mstudent[kostili].num_gr << "Номер в списке (порядковый номер) "
<< mstudent[kostili].num_list << " ФИО: " << mstudent[kostili].fio[0] << " " <<
mstudent[kostili].fio[1] << " " << mstudent[kostili].fio[2] << endl;
cout <<
"Форма обучения: " << mstudent[kostili].forma << endl;
cout <<
"
ОЦЕНКИ.\n" << "Физика. "
<< mstudent[kostili].ocenki[0] << "\n" << "История. " << mstudent[kostili].ocenki[1] << "\n"
<< "Алгебра. " << mstudent[kostili].ocenki[2] << endl;
cout <<
"Программирование. " << mstudent[kostili].ocenki[3] << "\n" << "МатАнализ. " <<
mstudent[kostili].ocenki[4] << "\n" << "Информатика. " << mstudent[kostili].ocenki[5] <<
endl;
cout <<
"Философия. " << mstudent[kostili].ocenki[6] << "\n" << "Экономика. " <<
mstudent[kostili].ocenki[7] << endl;
cout <<
"Дата формата (год месяц день час мин сек).\n " << "Год: " << mstudent[kostili].date[0] << "
Месяц: " << mstudent[kostili].date[1] << " День: " << mstudent[kostili].date[2];
cout <<
" Час: " << mstudent[kostili].date[3] << " Мин: " << mstudent[kostili].date[4] << " Сек: " <<
mstudent[kostili].date[5];
goto
kostili;
}
}
cout << "Записей о
студентах с номером в списке " << uff << " пока нет.\n";
goto kostili;
}
else {
    q++;
}
}
if (kol == q) {
    cout << "Записей о таких
группах пока нет.\n";
}
kostili:
cout << "\nПоиск
закончен...\n";
}

```

```
void PM(int kol) // поиск по дате
{
```

```
    которой начнем поиск данных <Дата (День) формат(хх)>." << endl;
```

```
    правильность данных \n";
```

```
    mstudent[i].num_gr;
```

```
    mstudent[i].num_list;
```

```
    mstudent[i].ocenki[0];
```

```
    mstudent[i].ocenki[1];
```

```
    mstudent[i].ocenki[2];
```

```
    mstudent[i].ocenki[3];
```

```
    mstudent[i].ocenki[4];
```

```
    mstudent[i].ocenki[5];
```

```
    mstudent[i].ocenki[6];
```

```
    mstudent[i].ocenki[7];
```

```
    cout << "Введите дату по
```

```
    int I;
```

```
    cin >> I;
```

```
    cin.sync();
```

```
    while (I > 31 || I < 0)
```

```
    {
```

```
        cout << "\n Проверьте
```

```
        cin >> I;
```

```
        cin.sync();
```

```
        continue;
```

```
    }
```

```
    int raz = 0, p = 0;
```

```
    struct MyStruct
```

```
    {
```

```
        int gr;
```

```
        int num;
```

```
        int oce[8];
```

```
        string form;
```

```
        string name[20];
```

```
        char s;
```

```
        int d[40];
```

```
    }uffMy[50];
```

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

```
    {
```

```
        if (mstudent[i].date[2] == I)
```

```
        {
```

```
            p++;
```

```
            uffMy[p].gr =
```

```
            uffMy[p].num =
```

```
            uffMy[p].oce[0] =
```

```
            uffMy[p].oce[1] =
```

```
            uffMy[p].oce[2] =
```

```
            uffMy[p].oce[3] =
```

```
            uffMy[p].oce[4] =
```

```
            uffMy[p].oce[5] =
```

```
            uffMy[p].oce[6] =
```

```
            uffMy[p].oce[7] =
```

```

mstudent[i].forma;
mstudent[i].fio[0];
mstudent[i].fio[1];
mstudent[i].fio[2];
mstudent[i].sex;
mstudent[i].date[0];
mstudent[i].date[1];
mstudent[i].date[2];
mstudent[i].date[3];
mstudent[i].date[4];
mstudent[i].date[5];

```

```

<< " числа записи о студентах не вносились" << endl;

```

```

GetStdHandle(STD_OUTPUT_HANDLE);

```

```

FOREGROUND_INTENSITY | FOREGROUND_INTENSITY);

```

```

студентах внесённые " << I << " числа до полудня" << endl;

```

```

группы: " << uffMy[i].gr << "Номер в списке (порядковый номер) " << uffMy[i].num << "
ФИО: " << uffMy[i].name[0] << " " << uffMy[i].name[1] << " " << uffMy[i].name[2] << endl;
обучения: " << uffMy[i].form << endl;

```

```

<< uffMy[i].oce[0] << "\n" << "История. " << uffMy[i].oce[1] << "\n" << "Алгебра. " <<
uffMy[i].oce[2] << endl;

```

```

uffMy[p].form =
uffMy[p].name[0] =
uffMy[p].name[1] =
uffMy[p].name[2] =
uffMy[p].s =
uffMy[p].d[0] =
uffMy[p].d[1] =
uffMy[p].d[2] =
uffMy[p].d[3] =
uffMy[p].d[4] =
uffMy[p].d[5] =

```

```

}
else {
    raz++;
}
}
if (raz == kol) {
    cout << " К сожалению " << I
}
for (int i = 1; i <= p; i++)
{
    if (uffMy[i].d[3] <= 11)
    {
        HANDLE R =

```

```

SetConsoleTextAttribute(R,

```

```

cout << "Данные о

```

```

cout << "\nНомер

```

```

cout << "Форма

```

```

cout << "

```

```

ОЦЕНКИ.\n" << "Физика. "

```

```

cout <<
"Программирование. " << uffMy[i].oce[3] << "\n" << "МатАнализ. " << uffMy[i].oce[4] <<
"\n" << "Информатика. " << uffMy[i].oce[5] << endl;
cout << "Философия.
" << uffMy[i].oce[6] << "\n" << "Экономика. " << uffMy[i].oce[7] << endl;
cout << "Пол " <<
uffMy[i].s << endl;
cout << "Дата
формата (год месяц день час мин сек).\n " << "Год: " << uffMy[i].d[0] << " Месяц: " <<
uffMy[i].d[1] << " День: " << uffMy[i].d[2];
cout << " Час: " <<
uffMy[i].d[3] << " Мин: " << uffMy[i].d[4] << " Сек: " << uffMy[i].d[5] << endl;
}
if (uffMy[i].d[3] > 12)
{
HANDLE my =
GetStdHandle(STD_OUTPUT_HANDLE);

SetConsoleTextAttribute(my,
FOREGROUND_GREEN | FOREGROUND_INTENSITY);
cout << "Данные о
студентах внесённые " << i << " числа после полудня" << endl;
cout << "\nНомер
группы: " << uffMy[i].gr << "Номер в списке (порядковый номер) " << uffMy[i].num << "
ФИО: " << uffMy[i].name[0] << " " << uffMy[i].name[1] << " " << uffMy[i].name[2] << endl;
cout << "Форма
обучения: " << uffMy[i].form << endl;
cout << "
ОЦЕНКИ.\n" << "Физика. "
<< uffMy[i].oce[0] << "\n" << "История. " << uffMy[i].oce[1] << "\n" << "Алгебра. " <<
uffMy[i].oce[2] << endl;
cout <<
"Программирование. " << uffMy[i].oce[3] << "\n" << "МатАнализ. " << uffMy[i].oce[4] <<
"\n" << "Информатика. " << uffMy[i].oce[5] << endl;
cout << "Философия.
" << uffMy[i].oce[6] << "\n" << "Экономика. " << uffMy[i].oce[7] << endl;
cout << "Пол " <<
uffMy[i].s << endl;
cout << "Дата
формата (год месяц день час мин сек).\n " << "Год: " << uffMy[i].d[0] << " Месяц: " <<
uffMy[i].d[1] << " День: " << uffMy[i].d[2];
cout << " Час: " <<
uffMy[i].d[3] << " Мин: " << uffMy[i].d[4] << " Сек: " << uffMy[i].d[5] << endl;
}
}
/*cout << "тест подсчёта " <<
p << endl;*/
}
void ReadFile(int kol) //работа с файлом Text.txt я знаю что по заданию название students
{
cout << "\n Загрузка файла
Text.txt \n";

```

открыт!!\n";

открыт!\n";

mstudent[i].ocenki[j];

mstudent[i].fio[j];

mstudent[i].date[j];

```
}
int main()
{
M1:
```

Создание новой записи о студенте.\n";

"#####@@@@@!!!!\$\$\$\$\$\_\_\_\_%%^^^&&&&\*\*\*\*~~~~~";

Внесение изменений в уже имеющуюся запись.\n";

```
ifstream file("Text.txt");
if (!file)
{
    cout << "Файл не

}
else
{
    cout << "Всё работает. Файл

}
for (int i = 0; i < kol; i++)
{
    file >> mstudent[i].num_gr;
    file >> mstudent[i].num_list;
    for (int j = 0; j < 8; ++j) {
        file >>

    }
    file >> mstudent[i].forma;
    for (int j = 0; j < 3; ++j) {
        file >>

    }
    file >> mstudent[i].sex;
    for (int j = 0; j < 6; ++j) {
        file >>

    }
    file.get();
}
file.close();
```

```
int xz = 0;
system("chcp 1251 > nul");
setlocale(LC_ALL, "Russian");
int lab = info(xz);
if (lab == 1) {
    cout << setw(4) << "1.

    cout <<

    shet(kol);
    vvod();
    goto M1;
}
if (lab == 2) {
    cout << setw(4) << "2.
```

"####@@@@!!!!\$\$\$\$\_\_\_\_%%^^^&&&\*\*\*\*~";

Вывод всех данных о студентах.\n";

"####@@@@!!!!\$\$\$\$\_\_\_\_%%^^^&&&\*\*\*\*~";

Вывод информации обо всех студентах группы N.N - инициализируется пользователем.\n";

"####@@@@!!!!\$\$\$\$\_\_\_\_%%^^^&&&\*\*\*\*~";

Вывод топа самых успешных студентов с наивысшим по рейтингу средним балом за прошедшую сессию.\n";

"####@@@@!!!!\$\$\$\$\_\_\_\_%%^^^&&&\*\*\*\*~";

Вывод количества студентов мужского и женского пола.\n";

"####@@@@!!!!\$\$\$\$\_\_\_\_%%^^^&&&\*\*\*\*~";

Определение количества студентов, которые будут получать стипендию(стипендия начисляется,если у студента нет троек и очная форма обучения).\n";

cout <<

```
shet(kol);
ReadFile(kol);
izmena(kol);
goto M1;
}
if (lab == 3) {
    cout << setw(4) << "3.
```

cout <<

```
shet(kol);
ReadFile(kol);
vivod(kol);
goto M1;
}
if (lab == 4) {
    cout << setw(4) << "4.
```

cout <<

```
shet(kol);
ReadFile(kol);
num_gr(kol);
goto M1;
}
if (lab == 5) {
    cout << setw(4) << "5.
```

cout <<

```
shet(kol);
ReadFile(kol);
topchik(kol);
goto M1;
}
if (lab == 6) {
    cout << setw(4) << "6.
```

cout <<

```
shet(kol);
ReadFile(kol);
sex(kol);
goto M1;
}
if (lab == 7) {
    cout << setw(4) << "7.
```

"####@@@@!!!!\$\$\$\$\_\_\_\_%%^^^&&&\*\*\*\*~";

cout <<

```
shet(kol);
ReadFile(kol);
stipuxa(kol);
goto M1;
}
if (lab == 8) {
    cout << setw(4) << "8.
```

Вывод данных о студентах, которые не получают стипендию; учатся только на `хорошо` и `отлично`; учатся только на `отлично`. \n";

cout <<

"####@@@@!!!!\$\$\$\$\_\_\_\_%%^^^&&&\*\*\*\*~";

```
shet(kol);
ReadFile(kol);
netu(kol);
goto M1;
}
if (lab == 9) {
    cout << setw(4) << "9.
```

Вывод данных о студентах, имеющих номер в списке - k. \n";

cout <<

"####@@@@!!!!\$\$\$\$\_\_\_\_%%^^^&&&\*\*\*\*~";

```
shet(kol);
ReadFile(kol);
list(kol);
goto M1;
}
if (lab == 10) {
    cout << setw(4) << "10.
```

Вывод всех записей, сделанных в день, который введёт пользователь. \n";

cout <<

"####@@@@!!!!\$\$\$\$\_\_\_\_%%^^^&&&\*\*\*\*~";

```
shet(kol);
ReadFile(kol);
PM(kol);
goto M1;
}
if (lab == 11) {
    cout << "\n GEME OVER \n";
    return 0;
}
if (lab == 12) {
    cout << setw(4) << "12.
```

Загрузить данные из файла. \n";

cout <<

"####@@@@!!!!\$\$\$\$\_\_\_\_%%^^^&&&\*\*\*\*~";

```
shet(kol);
ReadFile(kol);
goto M1;
}
return 0;}
```