**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.

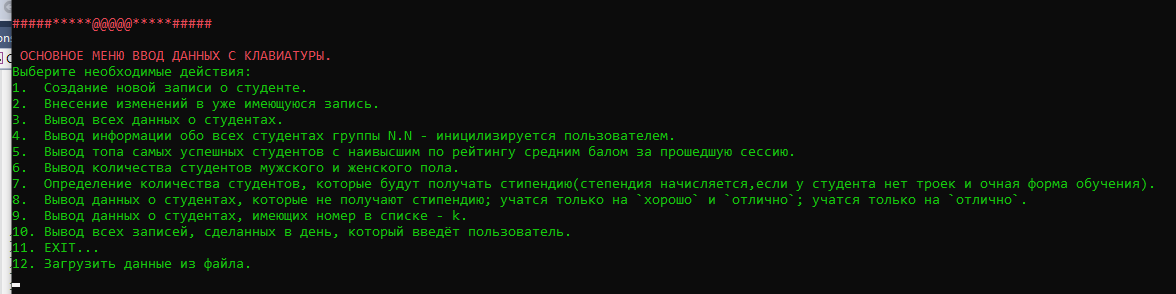


Рисунок 1

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

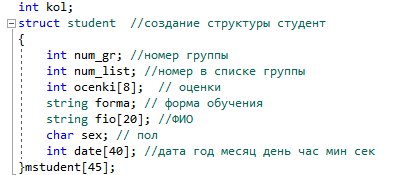


Рисунок 2 Создание структуры с требуемыми полями.

Теперь создадим текстовый документ



Рисунок 3 Создание текстового файла

Приступаем к выполнению 1 пункта задания.

Создание новой записи о студенте.

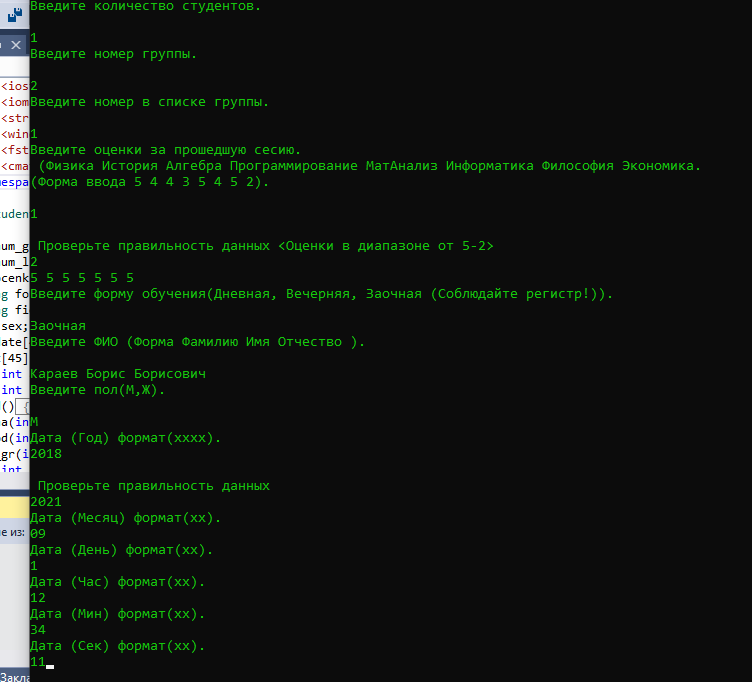


Рисунок 4 Консольный интерфейс создания записи.

При создании новой записи файл дописывается.

Внесение изменений в уже имеющуюся запись.

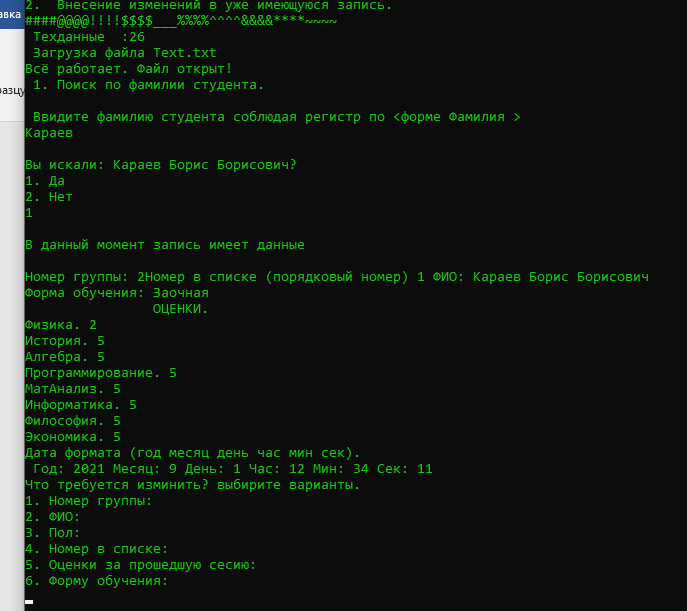


Рисунок 5 Консольный интерфейс внесения изменений в существующую запись.

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.

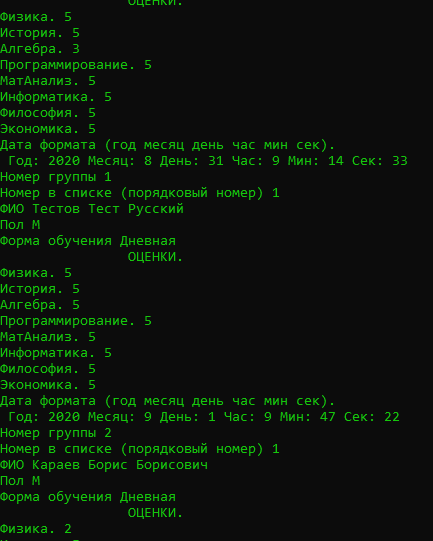


Рисунок 6 Консольный интерфейс вывода.

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

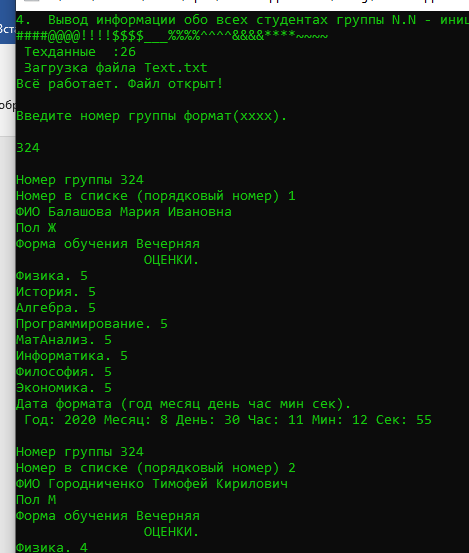


Рисунок 7 Консольный интерфейс вывода.

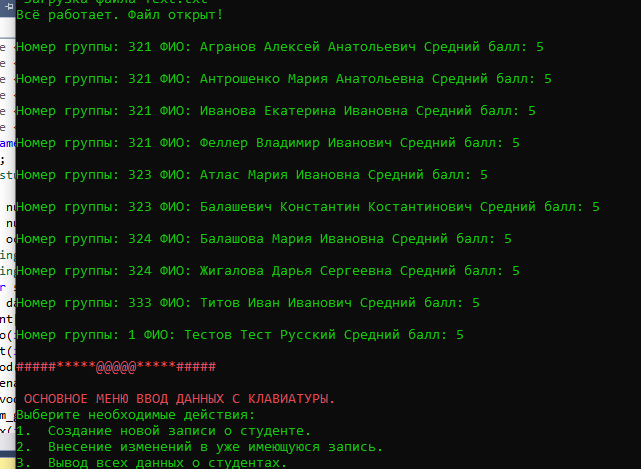
Вывод топа самых успешных студентов с наивысшим по рейтингу средним баллом за прошедшую сессию. 

Рисунок 8 Консольный интерфейс вывода.

Вывод количества студентов мужского и женского пола.

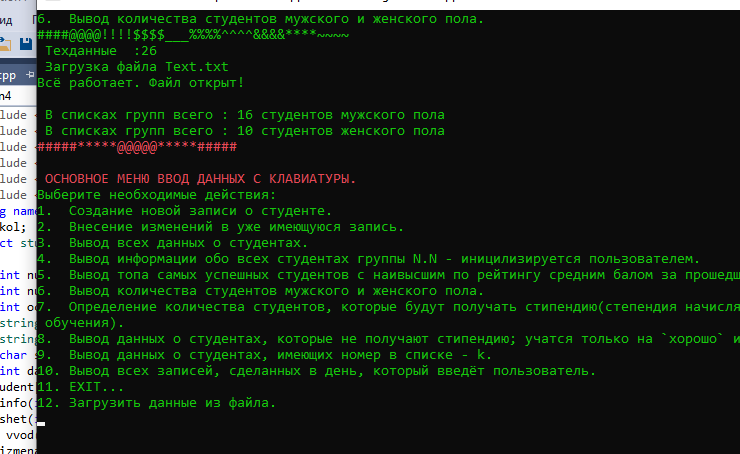


Рисунок 9 Консольный интерфейс вывода.

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

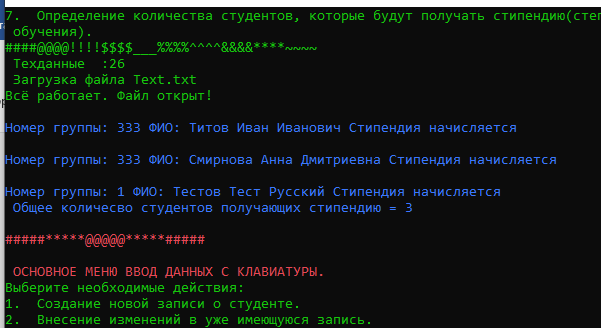


Рисунок 10 Консольный интерфейс вывода.

Вывод данных о студентах, которые не получают стипендию; учатся только на «хорошо» и «отлично»; учатся только на «отлично»;

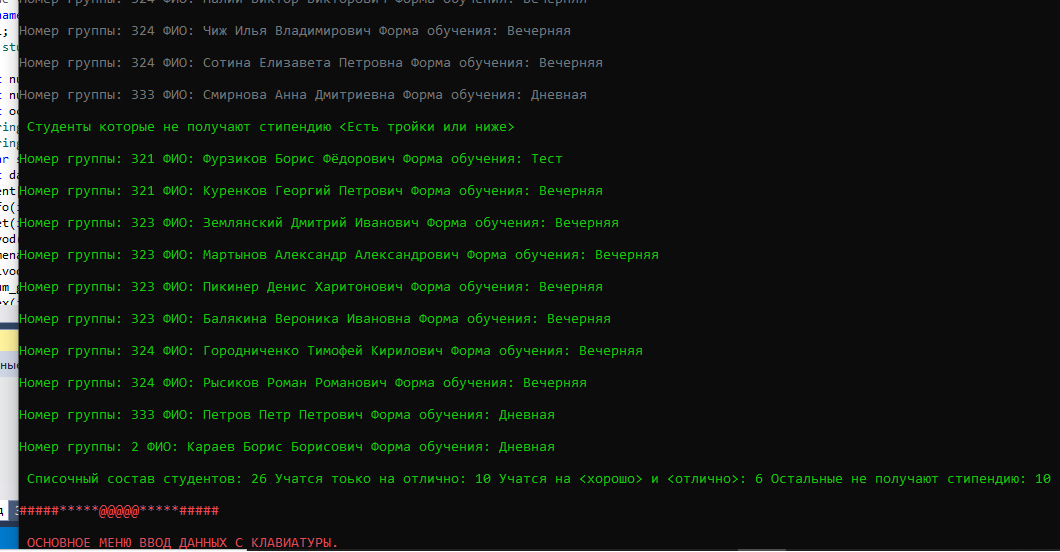


Рисунок 11 Консольный интерфейс вывода.

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

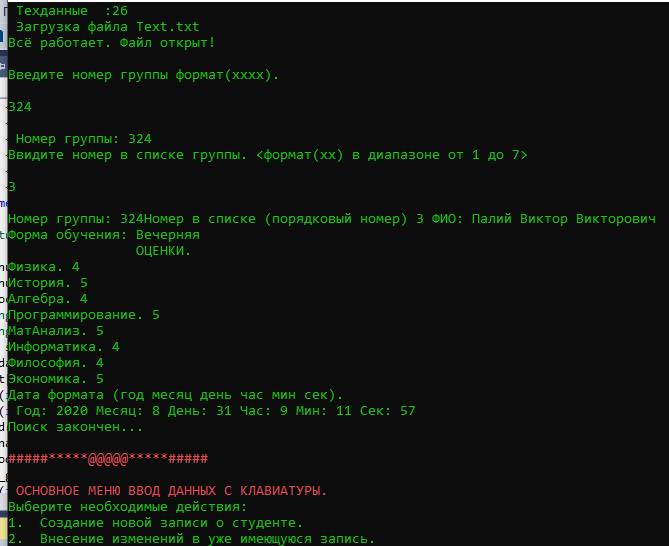


Рисунок 12 Консольный интерфейс вывода.

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

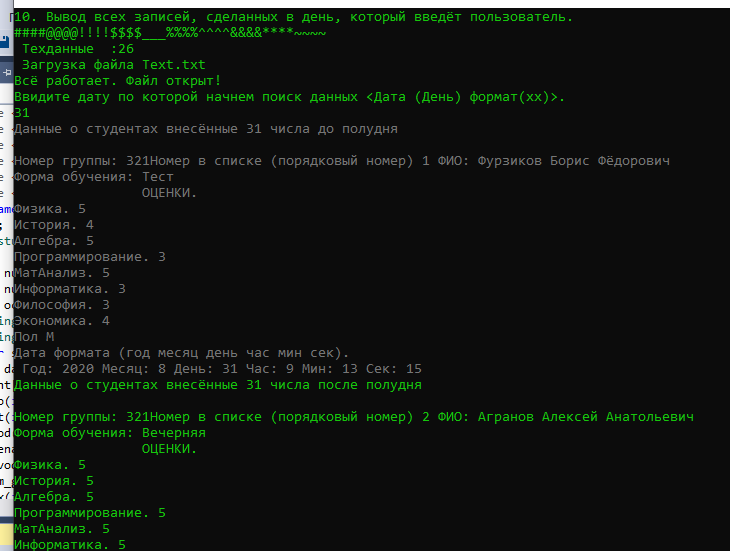


Рисунок 13 Консольный интерфейс вывода.

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 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

{

int num\_gr; //номер группы

int num\_list; //номер в списке группы

int ocenki[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];

int info(int a) //основное меню

{

int b = 0;

HANDLE O = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(O, FOREGROUND\_RED | FOREGROUND\_INTENSITY);

cout << "\n#####\*\*\*\*\*@@@@@\*\*\*\*\*#####\n";

cout << "\n ОСНОВНОЕ МЕНЮ ВВОД ДАННЫХ С КЛАВИАТУРЫ.\n";

SetConsoleTextAttribute(O, FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << setw(4) << "Выберите необходимые действия:" << "\n";

cout << setw(4) << "1. Создание новой записи о студенте.\n";

cout << setw(4) << "2. Внесение изменений в уже имеющуюся запись.\n";

cout << setw(4) << "3. Вывод всех данных о студентах.\n";

cout << setw(4) << "4. Вывод информации обо всех студентах группы N.N - иницилизируется пользователем.\n";

cout << setw(4) << "5. Вывод топа самых успешных студентов с наивысшим по рейтингу средним балом за прошедшую сессию.\n";

cout << setw(4) << "6. Вывод количества студентов мужского и женского пола.\n";

cout << setw(4) << "7. Определение количества студентов, которые будут получать стипендию(степендия начисляется,если у студента нет троек и очная форма обучения).\n";

cout << setw(4) << "8. Вывод данных о студентах, которые не получают стипендию; учатся только на `хорошо` и `отлично`; учатся только на `отлично`.\n";

cout << setw(4) << "9. Вывод данных о студентах, имеющих номер в списке - k.\n";

cout << setw(4) << "10. Вывод всех записей, сделанных в день, который введёт пользователь.\n";

cout << setw(4) << "11. EXIT...\n";

cout << setw(4) << "12. Загрузить данные из файла.\n";

cin >> b;

cin.sync();

system("cls");

return b;

}

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

{

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); // незнаю как работает это русский

SetConsoleOutputCP(1251); // это тоже для русских букв тестил на винде без руссого 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 сесию.\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)

{

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

cin >> dva;

cin.sync();

continue;

}

mstudent[i].ocenki[1] = dva;

int tri;

cin >> tri;

cin.sync();

while (tri > 5 || tri < 2)

{

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

cin >> tri;

cin.sync();

continue;

}

mstudent[i].ocenki[2] = tri;

int che;

cin >> che;

cin.sync();

while (che > 5 || che < 2)

{

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

cin >> che;

cin.sync();

continue;

}

mstudent[i].ocenki[3] = che;

int piy;

cin >> piy;

cin.sync();

while (piy > 5 || piy < 2)

{

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

cin >> piy;

cin.sync();

continue;

}

mstudent[i].ocenki[4] = piy;

int six;

cin >> six;

cin.sync();

while (six > 5 || six < 2)

{

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

cin >> six;

cin.sync();

continue;

}

mstudent[i].ocenki[5] = six;

int sem;

cin >> sem;

cin.sync();

while (sem > 5 || sem < 2)

{

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

cin >> sem;

cin.sync();

continue;

}

mstudent[i].ocenki[6] = sem;

int eil;

cin >> eil;

cin.sync();

while (eil > 5 || eil < 2)

{

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

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

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

cin >> mstudent[i].fio[j];

}

cout << "Введите пол(M,Ж).\n" << endl;

cin >> mstudent[i].sex;

cout << "Дата (Год) формат(xxxx).\n";

cin >> mstudent[i].date[0];

cin.sync();

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

{

cout << "\n Проверьте правильность данных \n";

cin >> mstudent[i].date[0];

cin.sync();

continue;

}

cout << "Дата (Месяц) формат(xx).\n";

cin >> mstudent[i].date[1];

cin.sync();

while (mstudent[i].date[1] > 12 || mstudent[i].date[1] < 1)

{

cout << "\n Проверьте правильность данных \n";

cin >> mstudent[i].date[1];

cin.sync();

continue;

}

cout << "Дата (День) формат(xx).\n";

cin >> mstudent[i].date[2];

cin.sync();

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

{

cout << "\n Проверьте правильность данных \n";

cin >> mstudent[i].date[2];

cin.sync();

continue;

}

cout << "Дата (Час) формат(xx).\n";

cin >> mstudent[i].date[3];

cin.sync();

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

{

cout << "\n Проверьте правильность данных \n";

cin >> mstudent[i].date[3];

cin.sync();

continue;

}

cout << "Дата (Мин) формат(xx).\n";

cin >> mstudent[i].date[4];

cin.sync();

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

{

cout << "\n Проверьте правильность данных \n";

cin >> mstudent[i].date[4];

cin.sync();

continue;

}

cout << "Дата (Сек) формат(xx).\n";

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

cin.sync();

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

{

cout << "\n Проверьте правильность данных \n";

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

cin.sync();

continue;

}

}

ofstream d;

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

if (!d)

{

cout << "Файл не открыт!!!\n";

}

else

{

cout << "Всё работает. Файл открыт в режиме дозаписи!\n";

}

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 << 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";

d << mstudent[i].sex;

d << "\n";

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

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

}

}

d.close();

}

int izmena(int kol) // изменение в имеющуюся запись

{

cout << " 1. Поиск по фамилии студента." << endl;

SetConsoleCP(1251); // незнаю как работает это русский

SetConsoleOutputCP(1251); // это тоже для русских букв тестил на винде без руссого это не работает

X2:

cout << "\n Ввидите фамилию студента cоблюдая 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 изминить? выбирите 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 сесию:" << 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 >> izi;

cin.sync();

continue;

}

if (izi == 1) {

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

cout << "\nВведите номер группы формат(xxxx).\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";

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";

d << mstudent[i].sex;

d << "\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;

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";

d << mstudent[i].sex;

d << "\n";

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

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

}

}

d.close();

return 0;

}

if (izi == 3) {

cout << "\nМеняем пол" << endl;

cout << "\nВведите пол студента формат <M или Ж>.\n";

char xzi;

cin >> xzi;

cin.sync();

mstudent[i].sex = xzi;

cout << "\nВнесены изменения пол студента " << mstudent[i].sex << 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";

d << 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 << "Ввидите номер в списке группы. <формат(xx) 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";

d << mstudent[i].sex;

d << "\n";

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

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

}

}

d.close();

return 0;

}

if (izi == 5) {

cout << "Введите оценки за прошедшую сесию.\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)

{

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

cin >> dva;

cin.sync();

continue;

}

mstudent[i].ocenki[1] = dva;

int tri;

cin >> tri;

cin.sync();

while (tri > 5 || tri < 2)

{

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

cin >> tri;

cin.sync();

continue;

}

mstudent[i].ocenki[2] = tri;

int che;

cin >> che;

cin.sync();

while (che > 5 || che < 2)

{

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

cin >> che;

cin.sync();

continue;

}

mstudent[i].ocenki[3] = che;

int piy;

cin >> piy;

cin.sync();

while (piy > 5 || piy < 2)

{

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

cin >> piy;

cin.sync();

continue;

}

mstudent[i].ocenki[4] = piy;

int six;

cin >> six;

cin.sync();

while (six > 5 || six < 2)

{

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

cin >> six;

cin.sync();

continue;

}

mstudent[i].ocenki[5] = six;

int sem;

cin >> sem;

cin.sync();

while (sem > 5 || sem < 2)

{

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

cin >> sem;

cin.sync();

continue;

}

mstudent[i].ocenki[6] = sem;

int eil;

cin >> eil;

cin.sync();

while (eil > 5 || eil < 2)

{

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";

}

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";

d << mstudent[i].sex;

d << "\n";

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

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

}

}

d.close();

return 0;

}

if (izi == 6) {

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

string xzi = "Дневная";

string xzy = "Вечерняя";

string xzz = "Заочная";

string six;

cin >> six;

cin.sync();

while (six == xzi || six == xzy || six == xzz)

{

cout << "\n Проверьте правильность данных таких форм обучения не существует\n";

cin >> six;

cin.sync();

continue;

}

mstudent[i].forma = six;

cout << "\nВнесены изменения теперь студент " << mstudent[i].fio[0] << " " << mstudent[i].fio[1] << " " << mstudent[i].fio[2] << endl;

cout << "Имеет форму обучения " << mstudent[i].forma << 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";

d << mstudent[i].sex;

d << "\n";

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

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

}

}

d.close();

return 0;

}

}

}

else {

uff++;

}

if (uff == kol) {

cout << "\nК сожалению студента " << x << " нет в списках " << endl;

}

}

return 0;

}

void vivod(int kol) //функция вывода

{

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

{

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];

}

}

void num\_gr(int kol) //поиск по группе

{

cout << "\nВведите номер группы формат(xxxx).\n" << endl;

int xz = 0;

cin >> xz;

cin.sync();

while (xz > 9999 || xz < 1)

{

cout << "\n Проверьте правильность данных таких групп не существует \n";

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 == mstudent[0].sex) {

a++;

}

else {

b++;

}

}

cout << "\n В списках групп всего : " << a << " студентов мужского пола";

cout << "\n В списках групп всего : " << b << " студентов женского пола";

}

void topchik(int kol) //топ студентов с Наивысшим по рейтингу среднем 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) // степендия 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;

}

}

}

}

cout << " Общее количесво студентов получающих стипендию = " << qwer << endl;

/\*return qwer;\*/

}

void netu(int kol) //8 пукт нечего сложного но повозится пришлось

{

HANDLE J = GetStdHandle(STD\_OUTPUT\_HANDLE);

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(xxxx).\n" << endl;

int xz = 0;

cin >> xz;

cin.sync();

while (xz > 9999 || xz < 0)

{

cout << "\n Проверьте правильность данных таких групп не существует \n";

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 группы: " << xz << endl;

cout << "Ввидите номер в списке группы. <формат(xx) 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;

}

if (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;

}

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

{

cout << "Ввидите дату по которой начнем поиск данных <Дата (День) формат(xx)>." << endl;

int I;

cin >> I;

cin.sync();

while (I > 31 || I < 0)

{

cout << "\n Проверьте правильность данных \n";

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 = mstudent[i].num\_gr;

uffMy[p].num = mstudent[i].num\_list;

uffMy[p].oce[0] = mstudent[i].ocenki[0];

uffMy[p].oce[1] = mstudent[i].ocenki[1];

uffMy[p].oce[2] = mstudent[i].ocenki[2];

uffMy[p].oce[3] = mstudent[i].ocenki[3];

uffMy[p].oce[4] = mstudent[i].ocenki[4];

uffMy[p].oce[5] = mstudent[i].ocenki[5];

uffMy[p].oce[6] = mstudent[i].ocenki[6];

uffMy[p].oce[7] = mstudent[i].ocenki[7];

uffMy[p].form = mstudent[i].forma;

uffMy[p].name[0] = mstudent[i].fio[0];

uffMy[p].name[1] = mstudent[i].fio[1];

uffMy[p].name[2] = mstudent[i].fio[2];

uffMy[p].s = mstudent[i].sex;

uffMy[p].d[0] = mstudent[i].date[0];

uffMy[p].d[1] = mstudent[i].date[1];

uffMy[p].d[2] = mstudent[i].date[2];

uffMy[p].d[3] = mstudent[i].date[3];

uffMy[p].d[4] = mstudent[i].date[4];

uffMy[p].d[5] = mstudent[i].date[5];

}

else {

raz++;

}

}

if (raz == kol) {

cout << " К сожелению " << I << " числа записи о студентах не вносились" << endl;

}

for (int i = 1; i <= p; i++)

{

if (uffMy[i].d[3] <= 11)

{

HANDLE R = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(R, FOREGROUND\_INTENSITY | FOREGROUND\_INTENSITY);

cout << "Данные о студентах внесённые " << I << " числа до 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;

}

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";

ifstream file("Text.txt");

if (!file)

{

cout << "Файл не открыт!!!\n";

}

else

{

cout << "Всё работает. Файл открыт!\n";

}

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 >> mstudent[i].ocenki[j];

}

file >> mstudent[i].forma;

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

file >> mstudent[i].fio[j];

}

file >> mstudent[i].sex;

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

file >> mstudent[i].date[j];

}

file.get();

}

file.close();

}

int main()

{

M1:

int xz = 0;

system("chcp 1251 > nul");

setlocale(LC\_ALL, "Russian");

int lab = info(xz);

if (lab == 1) {

cout << setw(4) << "1. Создание новой записи о студенте.\n";

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

shet(kol);

vvod();

goto M1;

}

if (lab == 2) {

cout << setw(4) << "2. Внесение изменений в уже имеющуюся запись.\n";

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

shet(kol);

ReadFile(kol);

izmena(kol);

goto M1;

}

if (lab == 3) {

cout << setw(4) << "3. Вывод всех данных о студентах.\n";

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

shet(kol);

ReadFile(kol);

vivod(kol);

goto M1;

}

if (lab == 4) {

cout << setw(4) << "4. Вывод информации обо всех студентах группы N.N - иницилизируется пользователем.\n";

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

shet(kol);

ReadFile(kol);

num\_gr(kol);

goto M1;

}

if (lab == 5) {

cout << setw(4) << "5. Вывод топа самых успешных студентов с наивысшим по рейтингу средним балом за прошедшую сессию.\n";

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

shet(kol);

ReadFile(kol);

topchik(kol);

goto M1;

}

if (lab == 6) {

cout << setw(4) << "6. Вывод количества студентов мужского и женского пола.\n";

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

shet(kol);

ReadFile(kol);

sex(kol);

goto M1;

}

if (lab == 7) {

cout << setw(4) << "7. Определение количества студентов, которые будут получать стипендию(степендия начисляется,если у студента нет троек и очная форма обучения).\n";

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;}