Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа №5

По дисциплине «Основы алгоритмизации и программирования»

На тему «Объединения, перечисления, битовые поля»

Выполнил(а):

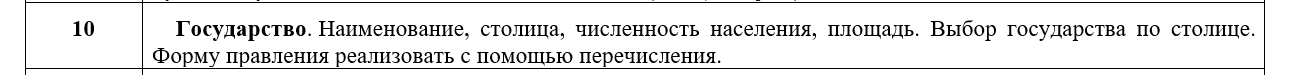
Студентка 1 курса 6 группы

Литвинчук Дарья

Преподаватель: асс. Андронова М.В.

Минск, 2024

**Вариант 10**



#include <iostream>

#include <string>

using namespace std;

const int SIZE = 30;

int current\_size = 0;

enum Form\_of\_government {

Democraty = 1, Tyrany, Kapitalistic, Monarhy

}forms\_of\_government;

struct Government {

string name;

string capital;

unsigned population : 6;

float area;

Form\_of\_government form\_of\_gov;

} governments[SIZE];

struct Government empty\_government;

void addNew();

void displayOneStructure(Government oneGovernment);

void display();

void deleteRecord();

void search();

void main()

{

short choose;

do{

cout << "Enter:\n1 - to enter new record;\n2 - to display information;\n3 - to delete the record;\n4 - to search\n5 - to exit.\n";

cin >> choose;

switch (choose) {

case 1: addNew(); break;

case 2: display(); break;

case 3: deleteRecord(); break;

case 4: search(); break;

}

} while (choose != 5);

}

void addNew() {

cout << "Information Entry" << endl;

cout << "Record number " << current\_size + 1 << endl;

cout << "Govenment name: ";

cin.get();

getline(cin, governments[current\_size].name);

cout << "Capital: ";

cin >> governments[current\_size].capital;

cout << "Population: ";

int temp;

cin >> temp;

governments[current\_size].population = temp;

cout << "Area (in thouthands sq km): ";

cin >> governments[current\_size].area;

cout << "Form of government(1 - Democraty, 2 - Tyrany, 3- Kapitalistic, 4 - Monarhy): ";

int form;

do {

cin >> form;

if (form < 1 or form > 4)

cout << "Incorrect Input";

} while (form < 1 or form > 4);

if (form == 1)

governments[current\_size].form\_of\_gov = Democraty;

if (form == 2)

governments[current\_size].form\_of\_gov = Tyrany;

if (form == 3)

governments[current\_size].form\_of\_gov = Kapitalistic;

if (form == 4)

governments[current\_size].form\_of\_gov = Monarhy;

current\_size++;

}

void displayOneStructure(Government oneGovernment) {

cout << "Government name: " << oneGovernment.name << endl;

cout << "Capital: " << oneGovernment.capital << endl;

cout << "Population: " << oneGovernment.population << endl;

cout << "Area (in thouthands sq km): " << oneGovernment.area << endl;

cout << "Form of fovernment: ";

if (oneGovernment.form\_of\_gov == Democraty)

cout << "Democraty" << endl;

if (oneGovernment.form\_of\_gov == Tyrany)

cout << "Tyrany" << endl;

if (oneGovernment.form\_of\_gov == Kapitalistic)

cout << "Kapitalistic" << endl;

if (oneGovernment.form\_of\_gov == Monarhy)

cout << "Monarhy" << endl;

}

void display() {

for (int i = 0; i < current\_size; i++) {

displayOneStructure(governments[i]);

cout << endl;

}

}

void deleteRecord() {

int rec\_number;

cout << "Enter teh number of record to delete" << endl;

cin >> rec\_number;

if (rec\_number != 0) {

for (int de = (rec\_number - 1); de < current\_size - 1; de++)

governments[de] = governments[de + 1];

current\_size -= 1;

}

if (rec\_number == 0)

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

governments[i] = empty\_government;

}

void search() {

int nedded\_area;

cout << "Enter the area: ";

cin >> nedded\_area;

cout << endl;

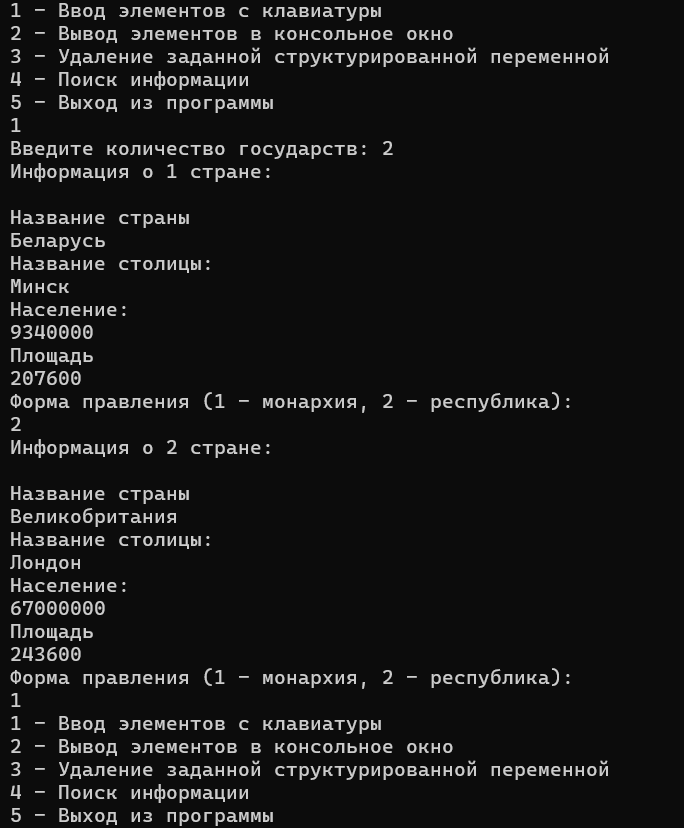
for (int i = 0; i < current\_size; i++) {

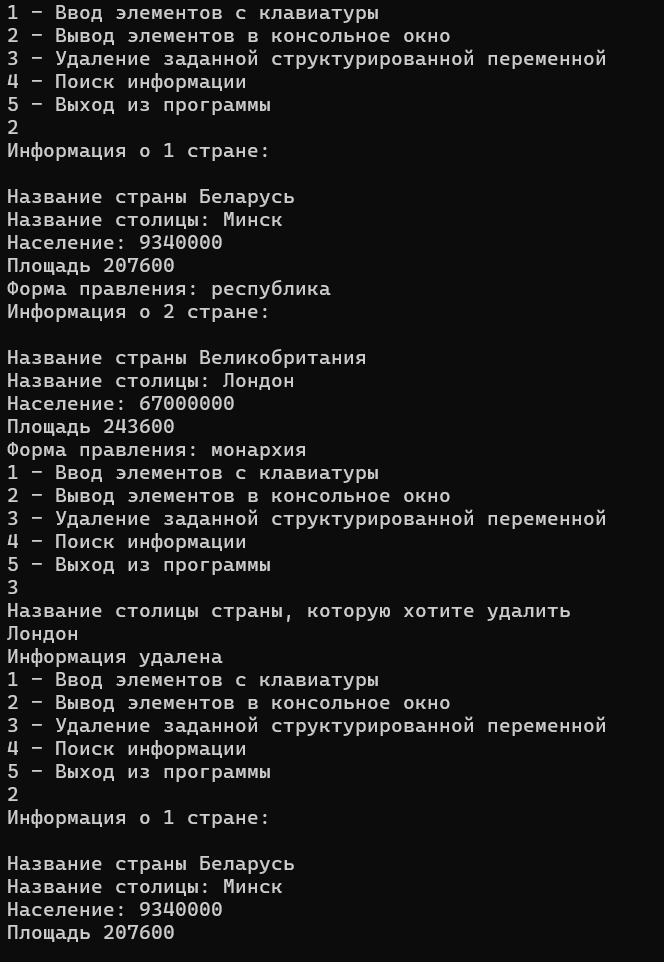
if (governments[i].area > nedded\_area)

displayOneStructure(governments[i]);

}

}







#include <iostream>

#include <fstream>

#include <string>

#include <windows.h>

#define size 10

using namespace std;

void input(); //прототип функции ввода

void output(); // прототип функции вывода

void inFile(); // прототип функции для записи в файл

void outFile(); // прототип функции для чтения из файла

void find(); // прототип функции поиска

union buf // объединение

{

// для понимания используем несколько переменных, хотя можно было использовать только одну

char fullname[50];

char birthday[50];

char addres[50];

char gender;

};

struct citizen // структура

{

buf full\_name;

buf birth;

buf addr;

buf gend;

};

citizen citizens[size];

int choice;

int current\_size;

int main()

{

setlocale(LC\_ALL, "Rus");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

//меню выбора

do

{

cout << "1 - Ввод элементов с клавиатуры" << endl;

cout << "2 - Вывод элементов в консольное окно" << endl;

cout << "3 - Запись информации в файл" << endl;

cout << "4 - Чтение информации из файла" << endl;

cout << "5 - Поиск информации по дате" << endl;

cout << "6 - Выход из программы" << endl;

cin >> choice;

// анализируем choice

switch (choice)

{

case 1:

input();

break;

case 2:

output();

break;

case 3:

inFile();

break;

case 4:

outFile();

break;

case 5:

find();

break;

}

} while (choice != 6);

return 0;

}

void input() // функция ввода

{

int numb; // для хранения количества горожан

int choice; // выбор из перечесления

cout << "Введите количество горожан: "; cin >> numb;

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

{

if (current\_size < size) // условие для проверки наличия места

{

// заполнение информации

cout << "Информация о " << i + 1 << " горожанине: " << endl;

cin.ignore();

cout << endl << "ФИО: " << endl;

gets\_s(citizens[current\_size].full\_name.fullname, 50);

cout << "Дата рождения: " << endl;

gets\_s(citizens[current\_size].birth.birthday, 50);

cout << "Адрес: " << endl;

gets\_s(citizens[current\_size].addr.addres, 50);

cout << "Пол: " << endl;

cin >> citizens[current\_size].gend.gender;

current\_size++;

}

}

}

void output() // функция вывода

{

// вывод информации

for (int i = 0; i < current\_size;i++)

{

cout << "Информация о " << i + 1 << " горожанине: " << endl;

cout << endl << "ФИО: " << citizens[i].full\_name.fullname << endl;

cout << "Дата рождения: " << citizens[i].birth.birthday<< endl;

cout << "Адрес: " << citizens[i].addr.addres << endl;

cout << "Пол: " << citizens[i].gend.gender << endl;

}

}

void inFile() // функция записи в файл

{

ofstream file1("FILE1.txt");

if (file1)

{

for (int i = 0; i < current\_size; i++)

{

file1 << "Информация о " << i + 1 << " горожанине: " << endl;

file1 << endl << "ФИО: " << citizens[i].full\_name.fullname << endl;

file1 << "Дата рождения: " << citizens[i].birth.birthday << endl;

file1 << "Адрес: " << citizens[i].addr.addres << endl;

file1 << "Пол: " << citizens[i].gend.gender << endl;

}

cout << "Информация записана в файл!"<<endl;

file1.close();

}

else

cout << "Не удалось открыть файл!" << endl;

}

void outFile() // функция чтения из файла

{

ifstream file1("FILE1.txt");

string line;

while (getline(file1, line))

{

cout << line<<endl;

}

file1.close();

}

void find() // функция поиска

{

string date;

bool f = true;

cout << "Введите дату рождения: ";

cin.ignore();

getline(cin, date);

for (int i = 0;i < current\_size;i++)

{

if (date == citizens[i].birth.birthday)

{

cout << "Информация о " << i + 1 << " горожанине: " << endl;

cout << endl << "ФИО: " << citizens[i].full\_name.fullname << endl;

cout << "Дата рождения: " << citizens[i].birth.birthday << endl;

cout << "Адрес: " << citizens[i].addr.addres << endl;

cout << "Пол: " << citizens[i].gend.gender << endl;

f = false;

}

}

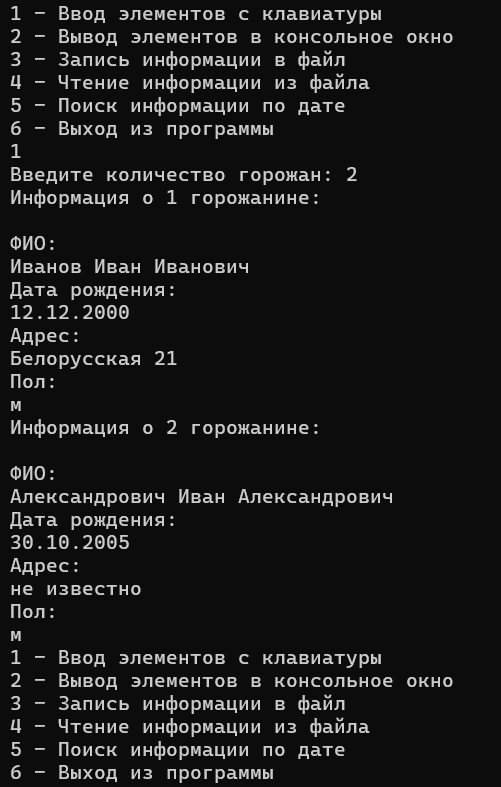
if (f)

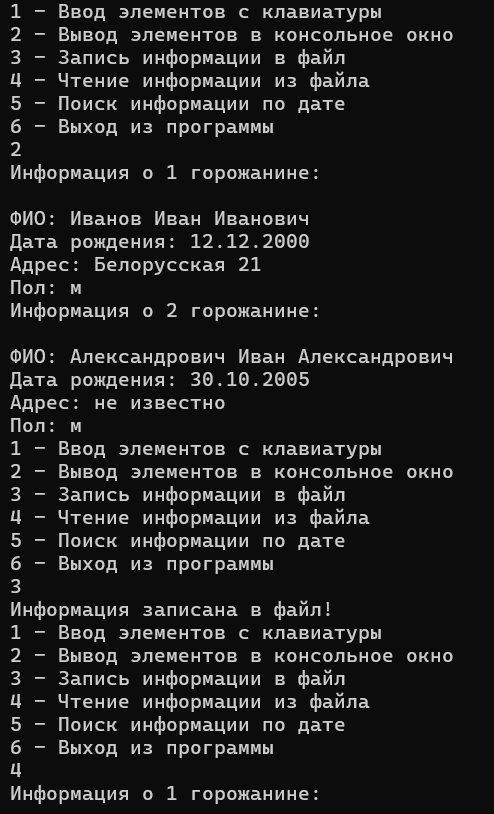
{

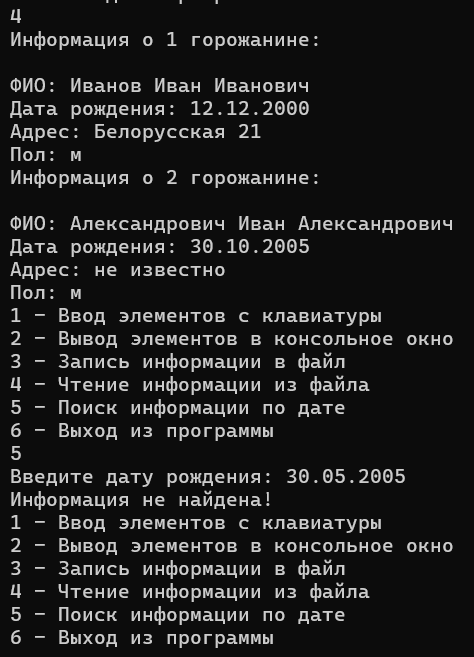
cout << "Информация не найдена!" << endl;

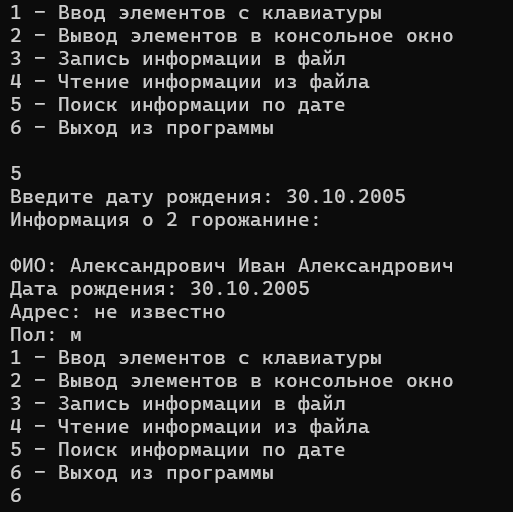
}

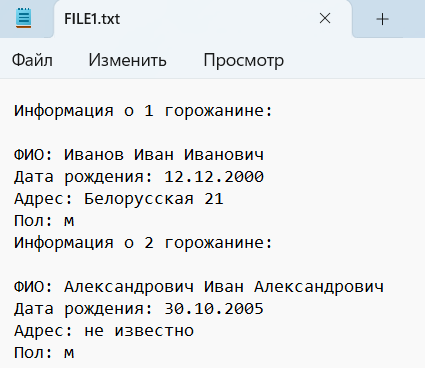
}











**Доп. Задачи**

**Вариант 12**



#include <iostream>

#include <string> // для работы со строками

#include <windows.h> // для корректного отображения символов

#define size 10 //подключение макроса для длины массива

using namespace std;

void input(); //прототип функции ввода

void output(); // прототип функции вывода

void del(); // прототип функции для удаления

void find(); // прототип функции поиска

struct notebook // структура

{

string name;

unsigned day : 5;

unsigned month : 4;

unsigned year : 14;

string addr;

string phonenumber;

};

notebook diary[size];

int choice;

int current\_size = 0;

int main()

{

setlocale(LC\_ALL, "Rus");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

//меню выбора

do

{

cout << "1 - Ввод элементов с клавиатуры" << endl;

cout << "2 - Вывод элементов в консольное окно" << endl;

cout << "3 - Удаление заданной структурированной переменной" << endl;

cout << "4 - Поиск информации " << endl;

cout << "5 - Выход из программы" << endl;

cin >> choice;

// анализируем choice

switch (choice)

{

case 1:

input();

break;

case 2:

output();

break;

case 3:

del();

break;

case 4:

find();

break;

}

} while (choice != 5);

return 0;

}

void input() // функция ввода

{

int day, month, year;

int numb; // для хранения количества

cout << "Введите количество записей: "; cin >> numb;

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

{

if (current\_size < size) // условие для проверки наличия места

{

// заполнение информации

cout << "Информация о " << i + 1 << " записи: " << endl;

cin.ignore();

cout << endl << "ФИО: " << endl;

getline(cin, diary[current\_size].name);

cout << "Адрес: " << endl;

getline(cin, diary[current\_size].addr);

cout << "Номер телефона: " << endl;

getline(cin, diary[current\_size].phonenumber);

cout << "Введите дату в формате ДД ММ ГГГГ: " << endl;

cin >> day >> month >> year;

diary[current\_size].day = day;

diary[current\_size].month = month;

diary[current\_size].year = year;

current\_size++;

}

}

}

void output() // функция вывода

{

// вывод информации

for (int i = 0; i < current\_size;i++)

{

cout << "Информация о " << i + 1 << " человеке в записной книжке: " << endl;

cout << endl << "ФИО: " << diary[i].name << endl;

cout << "Введите дату в формате ДД ММ ГГГГ: " << endl;

cout << "Дата рождения: " << diary[i].day << " " << diary[i].month << " " << diary[i].year << endl;

cout << "Адрес: " << diary[i].addr << endl;

cout << "Номер телефона: " << diary[i].phonenumber << endl;

}

}

void del() // функция для удаления

{

int d;

bool f = false;

string line;

cout << "Введите ФИО для удаления информации" << endl;

cin.ignore();

getline(cin, line);

for (int i = 0;i < current\_size;i++) // ищем подходящее

{

if (line==diary[i].name) // если нашли

{

d = i; // запоминаем

f = true;

}

}

if (f) // если нашли

{

for (int de1 = d; de1 < current\_size; de1++) // цикл для удаления

{

diary[de1] = diary[de1 + 1];

}

current\_size = current\_size - 1;

cout << "Информация удалена" << endl;

}

else

cout << "Информация не найдена!" << endl; // иначе выводим что не нашли

}

void find() // функция поиска

{

bool f = true;

string line;

cout << "Введите ФМО для удаления информации" << endl;

cin.ignore();

getline(cin, line);

for (int i = 0;i < current\_size;i++)

{

if (line == diary[i].name) // если нашли выводим

{

cout << "Информация о " << i + 1 << " записи: " << endl;

cout << endl << "ФИО: " << diary[i].name << endl;

cout << "Введите дату в формате ДД ММ ГГГГ: " << endl;

cout << "Дата рождения: " << diary[i].day << " " << diary[i].month << " " << diary[i].year << endl;

cout << "Адрес: " << diary[i].addr << endl;

cout << "Номер телефона: " << diary[i].phonenumber << endl;

f = false;

}

}

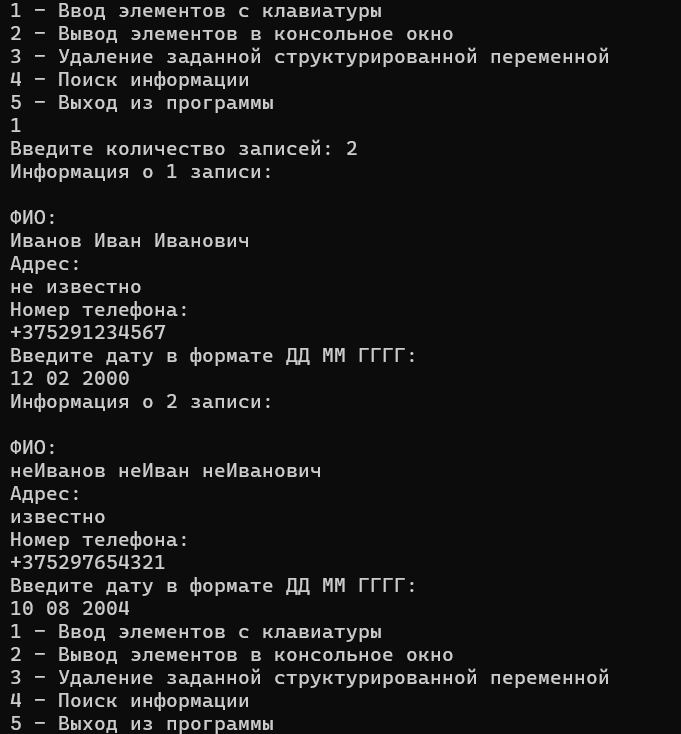
if (f)

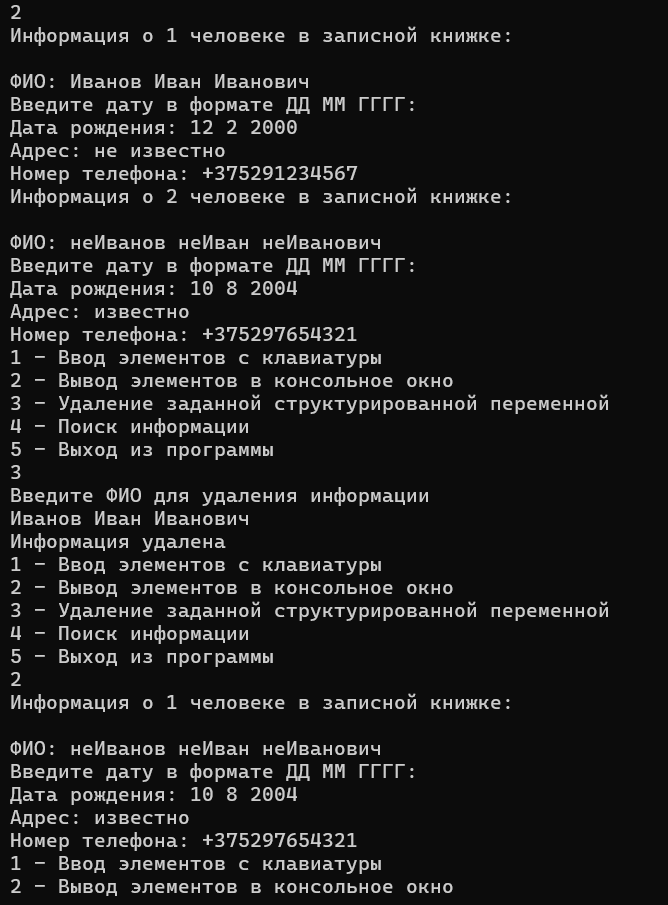
{

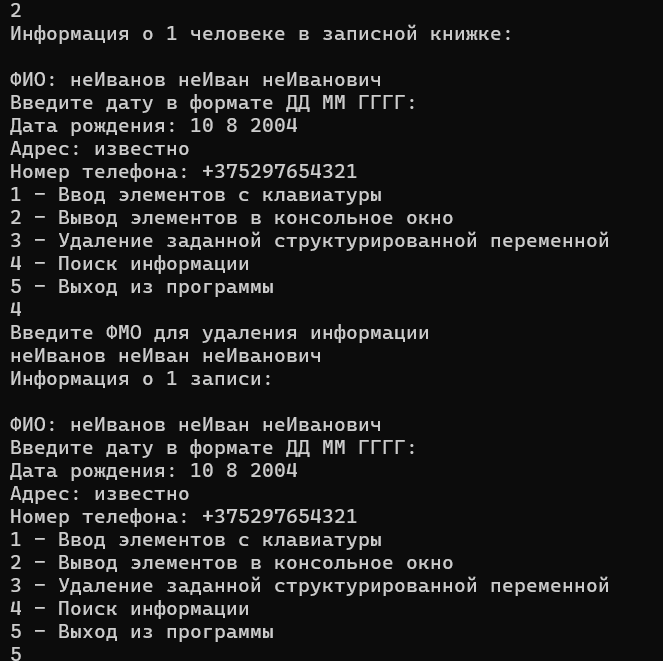
cout << "Информация не найдена!" << endl;

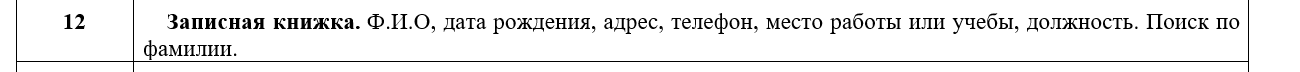
}

}

****

****

****

****

#include <iostream>

#include <fstream> // для работы с файлами

#include <string> // для работы со строками

#include <windows.h> // для корректного отображения символов

#define size 10 //подключение макроса для длины массива

using namespace std;

void input(); //прототип функции ввода

void output(); // прототип функции вывода

void inFile(); // прототип функции для записи в файл

void outFile(); // прототип функции для чтения из файла

void find(); // прототип функции поиска

union buf // объединение

{

// аналогичная ситуация

char name[50];

char date[50];

char addr[50];

char phonenumber[50];

char job[50];

char position[50];

};

struct notebook // структура

{

buf fullname;

buf birthday;

buf add;

buf phone;

buf pljob;

buf pos;

};

notebook diary[size];

int choice;

int current\_size;

int main()

{

setlocale(LC\_ALL, "Rus");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

//меню выбора

do

{

cout << "1 - Ввод элементов с клавиатуры" << endl;

cout << "2 - Вывод элементов в консольное окно" << endl;

cout << "3 - Запись информации в файл" << endl;

cout << "4 - Чтение информации из файла" << endl;

cout << "5 - Поиск информации" << endl;

cout << "6 - Выход из программы" << endl;

cin >> choice;

// анализируем choice

switch (choice)

{

case 1:

input();

break;

case 2:

output();

break;

case 3:

inFile();

break;

case 4:

outFile();

break;

case 5:

find();

break;

}

} while (choice != 6);

return 0;

}

void input() // функция ввода

{

int numb; // для хранения количества

int choice; // выбор из перечесления

cout << "Введите количество записей: "; cin >> numb;

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

{

if (current\_size < size) // условие для проверки наличия места

{

// заполнение информации

cin.ignore();

cout << "Информация о " << i + 1 << " человеке: " << endl;

cout << endl << "ФИО: " << endl;

gets\_s(diary[current\_size].fullname.name, 50);

cout << "Дата рождения: " << endl;

gets\_s(diary[current\_size].birthday.date, 50);

cout << "Адрес: " << endl;

gets\_s(diary[current\_size].add.addr, 50);

cout << "Телефон: " << endl;

gets\_s(diary[current\_size].phone.phonenumber, 50);

cout << "Место работы или учёбы: " << endl;

gets\_s(diary[current\_size].pljob.job, 50);

cout << "Должность: " << endl;

gets\_s(diary[current\_size].pos.position, 50);

current\_size++;

}

}

}

void output() // функция вывода

{

// вывод информации

for (int i = 0; i < current\_size;i++)

{

cout << "Информация о " << i + 1 << " человеке: " << endl;

cout << endl << "ФИО: " << diary[i].fullname.name << endl;

cout << "Дата рождения: " << diary[i].birthday.date << endl;

cout << "Адрес: " << diary[i].add.addr << endl;

cout << "Телефон: " << diary[i].phone.phonenumber << endl;

cout << "Место работы или учёбы: " << diary[i].pljob.job << endl;

cout << "Должность: " << diary[i].pos.position << endl;

}

}

void inFile() // функция записи в файл

{

ofstream file1("FILE1.txt");

if (file1)

{

for (int i = 0; i < current\_size; i++)

{

file1 << "Информация о " << i + 1 << " человеке: " << endl;

file1 << endl << "ФИО: " << diary[i].fullname.name << endl;

file1 << "Дата рождения: " << diary[i].birthday.date << endl;

file1 << "Адрес: " << diary[i].add.addr << endl;

file1 << "Телефон: " << diary[i].phone.phonenumber << endl;

file1 << "Место работы или учёбы: " << diary[i].pljob.job << endl;

file1 << "Должность: " << diary[i].pos.position << endl;

}

cout << "Информация записана в файл!" << endl;

file1.close();

}

else

cout << "Не удалось открыть файл!" << endl;

}

void outFile() // функция чтения из файла

{

ifstream file1("FILE1.txt");

string line;

while (getline(file1, line))

{

cout << line << endl;

}

file1.close();

}

void find() // функция поиска

{

string line;

bool f = true;

cout << "Введите ФИО: ";

cin.ignore();

getline(cin, line);

for (int i = 0;i < current\_size;i++)

{

if (line == diary[i].fullname.name)

{

cout << "Информация о " << i + 1 << " человеке: " << endl;

cout << endl << "ФИО: " << diary[i].fullname.name << endl;

cout << "Дата рождения: " << diary[i].birthday.date << endl;

cout << "Адрес: " << diary[i].add.addr << endl;

cout << "Телефон: " << diary[i].phone.phonenumber << endl;

cout << "Место работы или учёбы: " << diary[i].pljob.job << endl;

cout << "Должность: " << diary[i].pos.position << endl;

f = false;

}

}

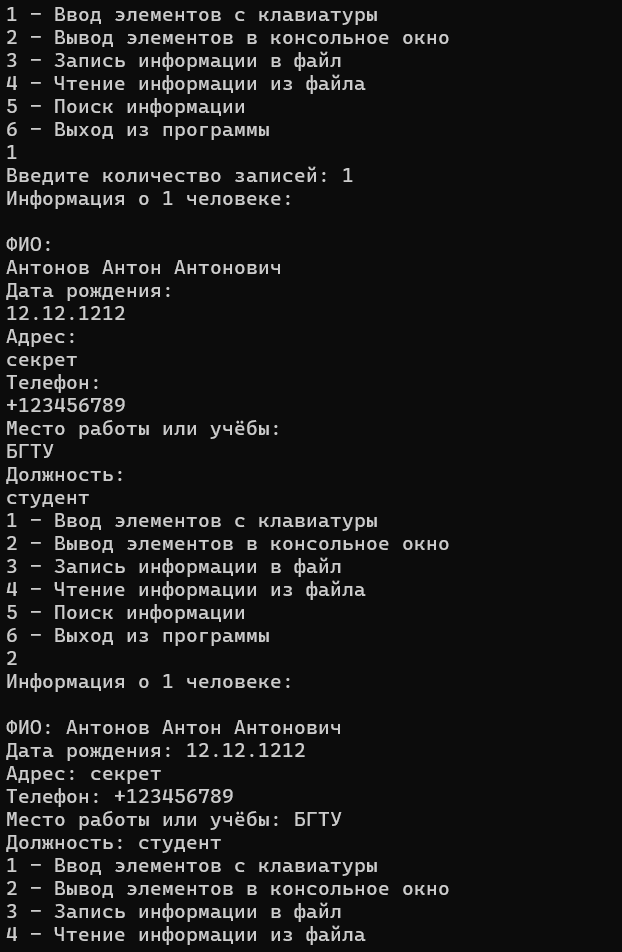
if (f)

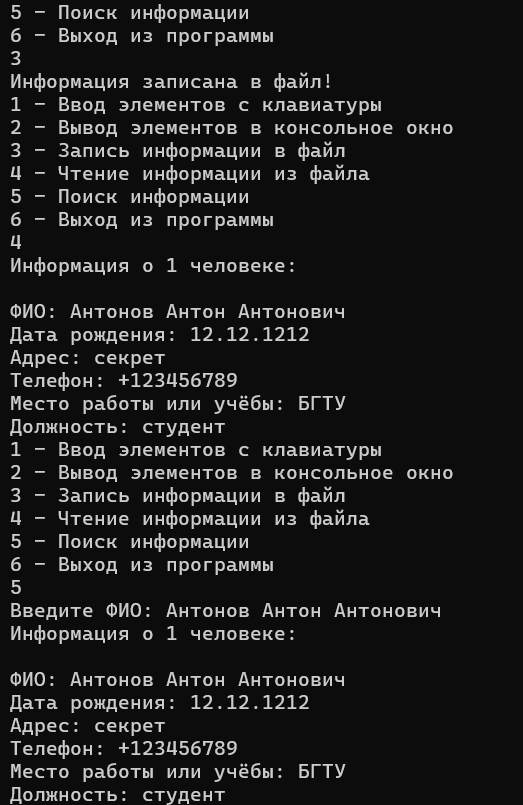
{

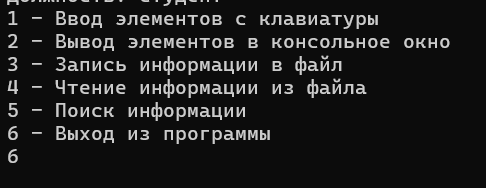
cout << "Информация не найдена!" << endl;

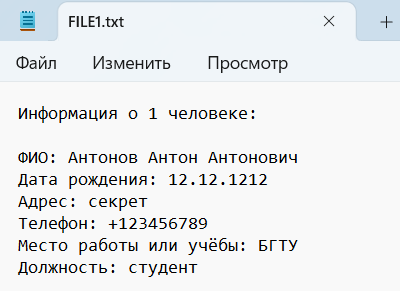
}

}

****

****

****

****

Вариант 13



#include <iostream>

#include <string> // для работы со строками

#include <windows.h> // для корректного отображения символов

#define size 10 //подключение макроса для длины массива

using namespace std;

void input(); //прототип функции ввода

void output(); // прототип функции вывода

void del(); // прототип функции для удаления

void find(); // прототип функции поиска

enum faculty // перечисление

{

ФИТ = 1, не\_ФИТ

};

struct student // структура

{

char name[50];

char speciality[50];

int group;

float average;

faculty fac;

unsigned day : 5;

unsigned month : 4;

unsigned year : 14;

};

student students[size];

int choice;

int current\_size = 0;

int main()

{

setlocale(LC\_ALL, "Rus");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

//меню выбора

do

{

cout << "1 - Ввод элементов с клавиатуры" << endl;

cout << "2 - Вывод элементов в консольное окно" << endl;

cout << "3 - Удаление заданной структурированной переменной" << endl;

cout << "4 - Поиск информации " << endl;

cout << "5 - Выход из программы" << endl;

cin >> choice;

// анализируем choice

switch (choice)

{

case 1:

input();

break;

case 2:

output();

break;

case 3:

del();

break;

case 4:

find();

break;

}

} while (choice != 5);

return 0;

}

void input() // функция ввода

{

int day, month, year;

int numb; // для хранения количества

cout << "Введите количество записей: "; cin >> numb;

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

{

if (current\_size < size) // условие для проверки наличия места

{

// заполнение информации

cout << "Информация о " << i + 1 << " студенте: " << endl;

cin.ignore();

cout << endl << "ФИО: " << endl;

gets\_s(students[current\_size].name,50);

cout << "Специальность: "<<endl;

gets\_s(students[current\_size].speciality, 50);

cout << "Введите дату поступления в формате ДД ММ ГГГГ: " << endl;

cin >> day >> month >> year;

students[current\_size].day = day;

students[current\_size].month = month;

students[current\_size].year = year;

cout << "Факультет (1 - ФИТ, 2 - не ФИТ): " << endl;

cin >> choice;

switch (choice)

{

case 1:

students[current\_size].fac = ФИТ;

break;

case 2:

students[current\_size].fac = не\_ФИТ;

break;

}

cout << "Группа: " << endl;

cin >> students[current\_size].group;

cout << "Средний балл: " << endl;

cin >> students[current\_size].average;

current\_size++;

}

}

}

void output() // функция вывода

{

// вывод информации

for (int i = 0; i < current\_size;i++)

{

cout << endl << "ФИО: " << students[i].name << endl;

cout << "Дата поступления: " << students[i].day << students[i].month << students[i].year <<endl;

cout << "Специальность: " << students[i].speciality<< endl;

cout << "Группа: " << students[i].group << endl;

cout << "Факультет: ";

if (students[i].fac == ФИТ)

{

cout << "ФИТ";

}

else

if (students[i].fac == не\_ФИТ)

{

cout << " не ФИТ";

}

cout << endl;

cout << "Средний балл: " << students[i].average << endl;

}

}

void del() // функция для удаления

{

int d;

bool f = false;

float av;

cout << "Введите средний балл для удаления информации" << endl;

cin >> av;

for (int i = 0;i < current\_size;i++) // ищем подходящее

{

if (av == students[i].average) // если нашли

{

d = i; // запоминаем

f = true;

}

}

if (f) // если нашли

{

for (int de1 = d; de1 < current\_size; de1++) // цикл для удаления

{

students[de1] = students[de1 + 1];

}

current\_size = current\_size - 1;

cout << "Информация удалена" << endl;

}

else

cout << "Информация не найдена!" << endl; // иначе выводим что не нашли

}

void find() // функция поиска

{

bool f = true;

float av;

cout << "Введите средний балл для поиска информации: " << endl;

cin >> av;

for (int i = 0;i < current\_size;i++)

{

if (av == students[i].average) // если нашли выводим

{

cout << endl << "ФИО: " << students[i].name << endl;

cout << "Дата поступления: " << students[i].day <<" " << students[i].month << " " << students[i].year << endl;

cout << "Специальность: " << students[i].speciality << endl;

cout << "Группа: " << students[i].group << endl;

cout << "Факультет: ";

if (students[i].fac == ФИТ)

{

cout << "ФИТ";

}

else

if (students[i].fac == не\_ФИТ)

{

cout << "не ФИТ";

}

cout << endl;

cout << "Средний балл: " << students[i].average << endl;

f = false;

}

}

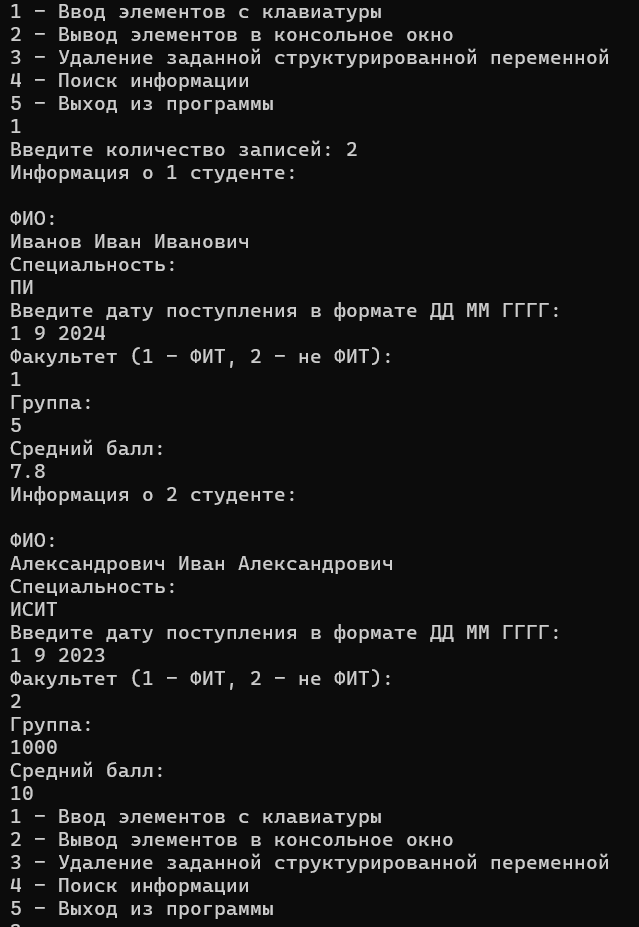
if (f)

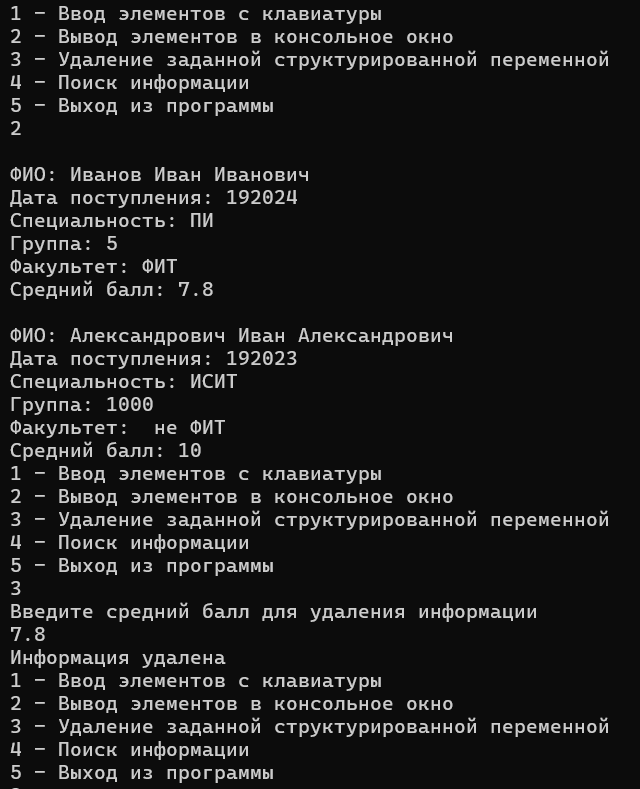
{

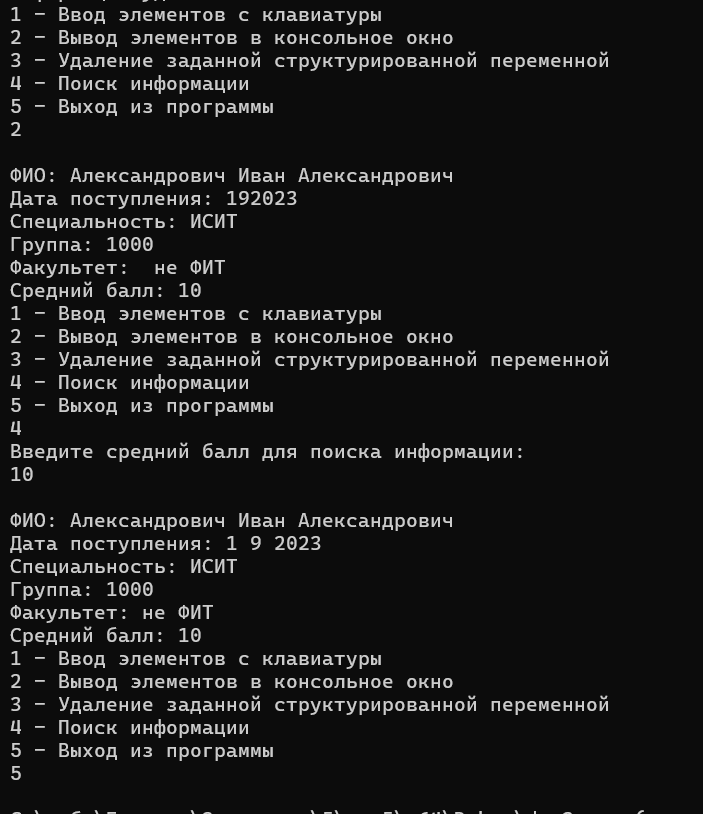
cout << "Информация не найдена!" << endl;

}

}









#include <iostream>

#include <fstream> // для работы с файлами

#include <string> // для работы со строками

#include <windows.h> // для корректного отображения символов

#define size 10 //подключение макроса для длины массива

using namespace std;

void input(); //прототип функции ввода

void output(); // прототип функции вывода

void inFile(); // прототип функции для записи в файл

void outFile(); // прототип функции для чтения из файла

void find(); // прототип функции поиска

union buf // объединение

{

char name[50];

char capital[50];

int people;

int square;

char prezident[50];

};

struct country // структура

{

buf nam;

buf cap;

buf peop;

buf sq;

buf pr;

};

country countries[size];

int choice;

int current\_size;

int main()

{

setlocale(LC\_ALL, "Rus");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

//меню выбора

do

{

cout << "1 - Ввод элементов с клавиатуры" << endl;

cout << "2 - Вывод элементов в консольное окно" << endl;

cout << "3 - Запись информации в файл" << endl;

cout << "4 - Чтение информации из файла" << endl;

cout << "5 - Поиск информации" << endl;

cout << "6 - Выход из программы" << endl;

cin >> choice;

// анализируем choice

switch (choice)

{

case 1:

input();

break;

case 2:

output();

break;

case 3:

inFile();

break;

case 4:

outFile();

break;

case 5:

find();

break;

}

} while (choice != 6);

return 0;

}

void input() // функция ввода

{

int numb; // для хранения количества

int choice; // выбор из перечесления

cout << "Введите количество записей: "; cin >> numb;

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

{

if (current\_size < size) // условие для проверки наличия места

{

// заполнение информации

cin.ignore();

cout << "Информация о " << i + 1 << " стране: " << endl;

cout << endl << "Название: " << endl;

gets\_s(countries[current\_size].nam.name, 50);

cout << "Столица: " << endl;

gets\_s(countries[current\_size].cap.capital, 50);

cout << "Фамилия президента: " << endl;

gets\_s(countries[current\_size].pr.prezident, 50);

cout << "Численность: " << endl;

cin >> countries[current\_size].peop.people;

cout << "Площадь: " << endl;

cin >> countries[current\_size].sq.square;

current\_size++;

}

}

}

void output() // функция вывода

{

// вывод информации

for (int i = 0; i < current\_size;i++)

{

cout << "Информация о " << i + 1 << " стране: " << endl;

cout << endl << "Название: " << countries[i].nam.name << endl;

cout << "Столица: " << countries[i].cap.capital << endl;

cout << "Численность: " << countries[i].peop.people << endl;

cout << "Площадь: " << countries[i].sq.square << endl;

cout << "Фамилия президента: " << countries[i].pr.prezident << endl;

}

}

void inFile() // функция записи в файл

{

ofstream file1("FILE1.txt");

if (file1)

{

for (int i = 0; i < current\_size; i++)

{

file1 << "Информация о " << i + 1 << " стране: " << endl;

file1 << endl << "Название: " << countries[i].nam.name << endl;

file1 << "Столица: " << countries[i].cap.capital << endl;

file1 << "Численность: " << countries[i].peop.people << endl;

file1 << "Площадь: " << countries[i].sq.square << endl;

file1 << "Фамилия президента: " << countries[i].pr.prezident << endl;

}

cout << "Информация записана в файл!" << endl;

file1.close();

}

else

cout << "Не удалось открыть файл!" << endl;

}

void outFile() // функция чтения из файла

{

ifstream file1("FILE1.txt");

string line;

while (getline(file1, line))

{

cout << line << endl;

}

file1.close();

}

void find() // функция поиска

{

string line;

bool f = true;

cout << "Введите фамилию президента: ";

cin.ignore();

getline(cin, line);

for (int i = 0;i < current\_size;i++)

{

if (line == countries[i].pr.prezident)

{

cout << "Информация о " << i + 1 << " стране: " << endl;

cout << endl << "Название: " << countries[i].nam.name << endl;

cout << "Столица: " << countries[i].cap.capital << endl;

cout << "Численность: " << countries[i].peop.people << endl;

cout << "Площадь: " << countries[i].sq.square << endl;

cout << "Фамилия президента: " << countries[i].pr.prezident << endl;

f = false;

}

}

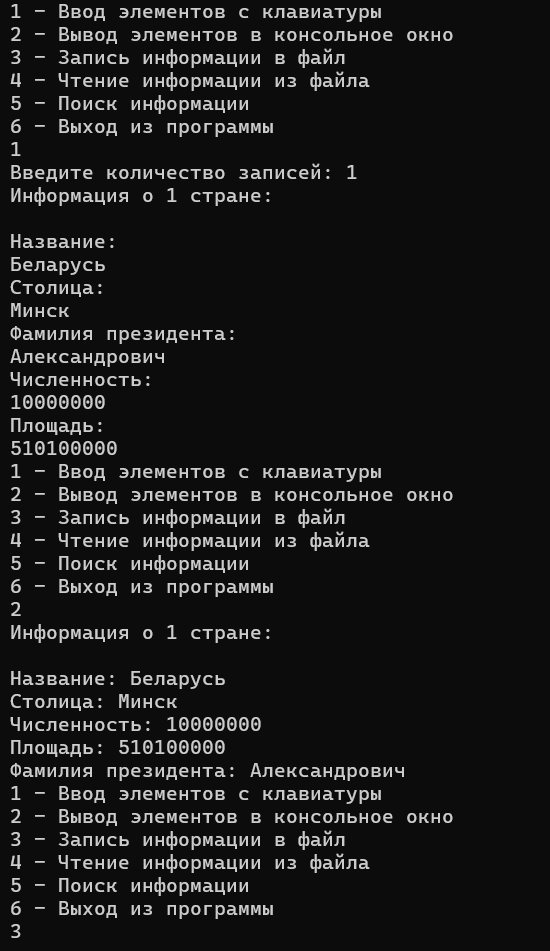
if (f)

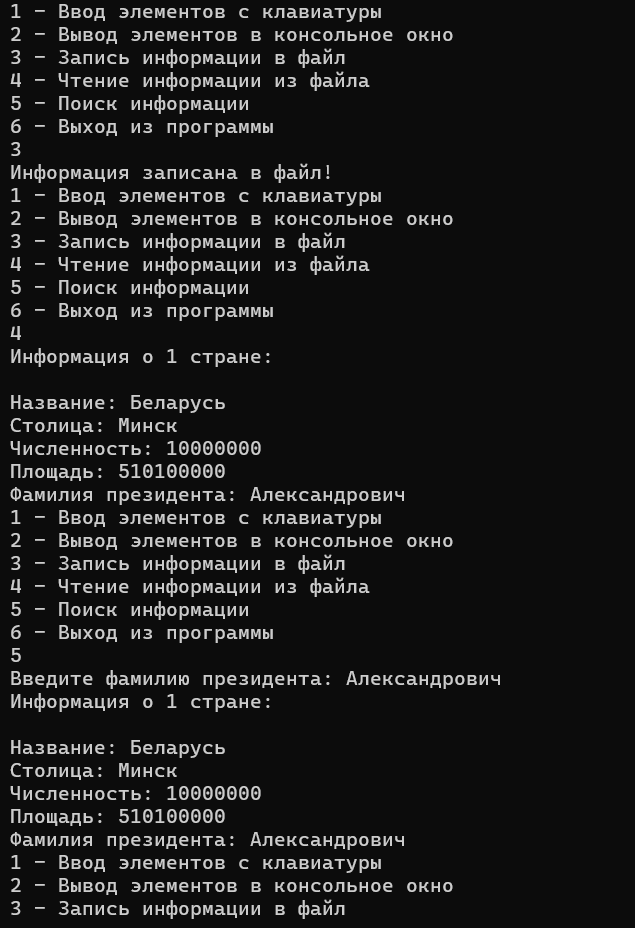
{

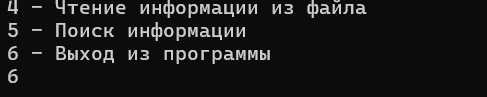
cout << "Информация не найдена!" << endl;

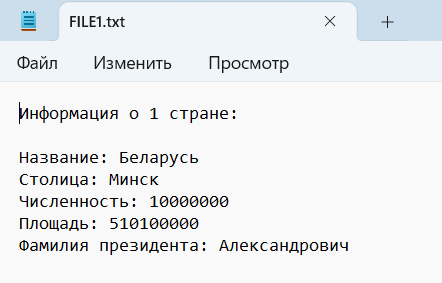
}

}

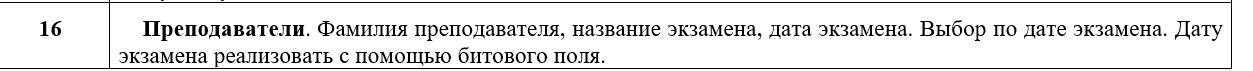








Вариант 16



#include <iostream>

#include <string> // для работы со строками

#include <windows.h> // для корректного отображения символов

#define size 10 //подключение макроса для длины массива

using namespace std;

void input(); //прототип функции ввода

void output(); // прототип функции вывода

void del(); // прототип функции для удаления

void find(); // прототип функции поиска

struct teacher // структура

{

string name;

string exam;

unsigned day : 5;

unsigned month : 4;

unsigned year : 14;

};

teacher teachers[size];

int choice;

int current\_size=0;

int main()

{

setlocale(LC\_ALL, "Rus");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

//меню выбора

do

{

cout << "1 - Ввод элементов с клавиатуры" << endl;

cout << "2 - Вывод элементов в консольное окно" << endl;

cout << "3 - Удаление заданной структурированной переменной" << endl;

cout << "4 - Поиск информации " << endl;

cout << "5 - Выход из программы" << endl;

cin >> choice;

// анализируем choice

switch (choice)

{

case 1:

input();

break;

case 2:

output();

break;

case 3:

del();

break;

case 4:

find();

break;

}

} while (choice != 5);

return 0;

}

void input() // функция ввода

{

int day, month, year;

int teach; // для хранения количества преподавателей

cout << "Введите количество преподавателей: "; cin >> teach;

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

{

if (current\_size < size) // условие для проверки наличия места

{

// заполнение информации

cout << "Информация о " << i + 1 << " преподавателе: " << endl;

cin.ignore();

cout << endl << "ФИО преподавателя: " << endl;

getline(cin, teachers[current\_size].name);

cout << "Название экзамена: " << endl;

getline(cin, teachers[current\_size].exam);

cout << "Введите дату в формате ДД ММ ГГГГ: " << endl;

cin >> day>>month>>year;

teachers[current\_size].day = day;

teachers[current\_size].month = month;

teachers[current\_size].year = year;

current\_size++;

}

}

}

void output() // функция вывода

{

// вывод информации

for (int i = 0; i < current\_size;i++)

{

cout << "Информация о " << i + 1 << " преподавателе " << endl;

cout << endl << "ФИО преподавателя: " << teachers[i].name <<endl;

cout << "Название экзамена: " << teachers[i].exam <<endl;

cout << "Дата экзамена: " << teachers[i].day <<" "<< teachers[i].month <<" " << teachers[i].year << endl;

}

}

void del() // функция для удаления

{

int d;

bool f = false;

int day, month, year;

cout << "Введите дату экзамен которого хотите удалить в формате ДД ММ ГГГГ" << endl;

cin >> day >> month >> year;

cin.ignore();

for (int i = 0;i < current\_size;i++) // ищем подходящее

{

if (day==teachers[i].day && month == teachers[i].month && year == teachers[i].year) // если нашли

{

d = i; // запоминаем

f = true;

}

}

if (f) // если нашли

{

for (int de1 = d; de1 < current\_size; de1++) // цикл для удаления

{

teachers[de1] = teachers[de1 + 1];

}

current\_size = current\_size - 1;

cout << "Информация удалена" << endl;

}

else

cout << "Преподаватель не найден!" << endl; // иначе выводим что не нашли

}

void find() // функция поиска

{

bool f=true;

int day, month, year;

cout << "Введите дату экзамен чтобы найти преподаватаеля в формате ДД ММ ГГГГ" << endl;

cin >> day >> month >> year;

cin.ignore();

for (int i = 0;i < current\_size;i++)

{

if (day == teachers[i].day && month == teachers[i].month && year == teachers[i].year) // если нашли выводим

{

cout << "Информация о " << i + 1 << " преподавателе: " << endl;

cout << endl << "ФИО преподавателя: " << teachers[i].name<<endl;

cout << "Название экзамена: " << teachers[i].exam<<endl;

cout << "Дата экзамена: " << teachers[i].day <<" " << teachers[i].month <<" " << teachers[i].year << endl;

f = false;

} }

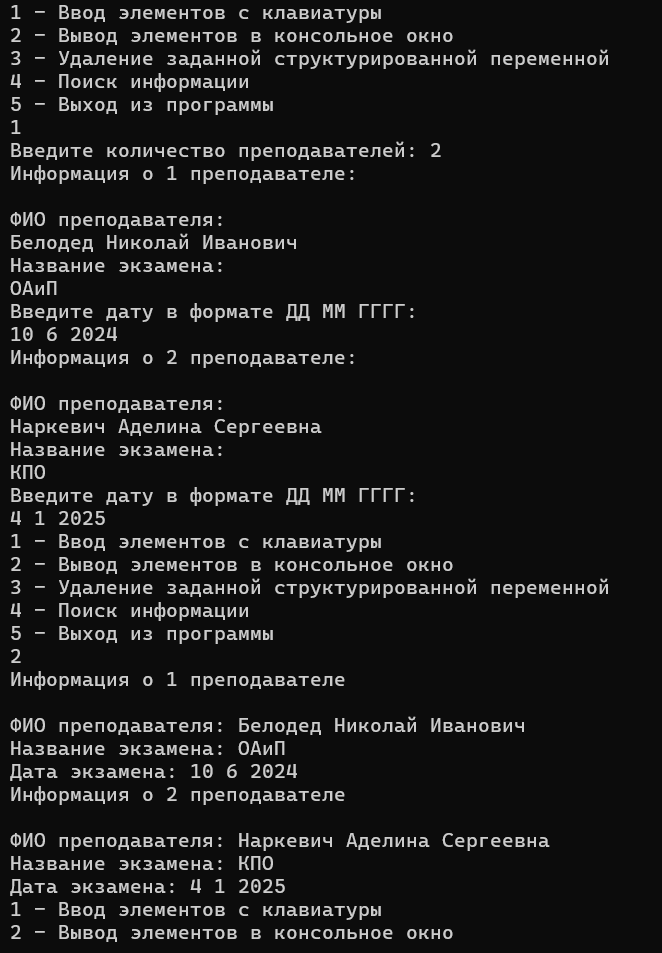
if (f)

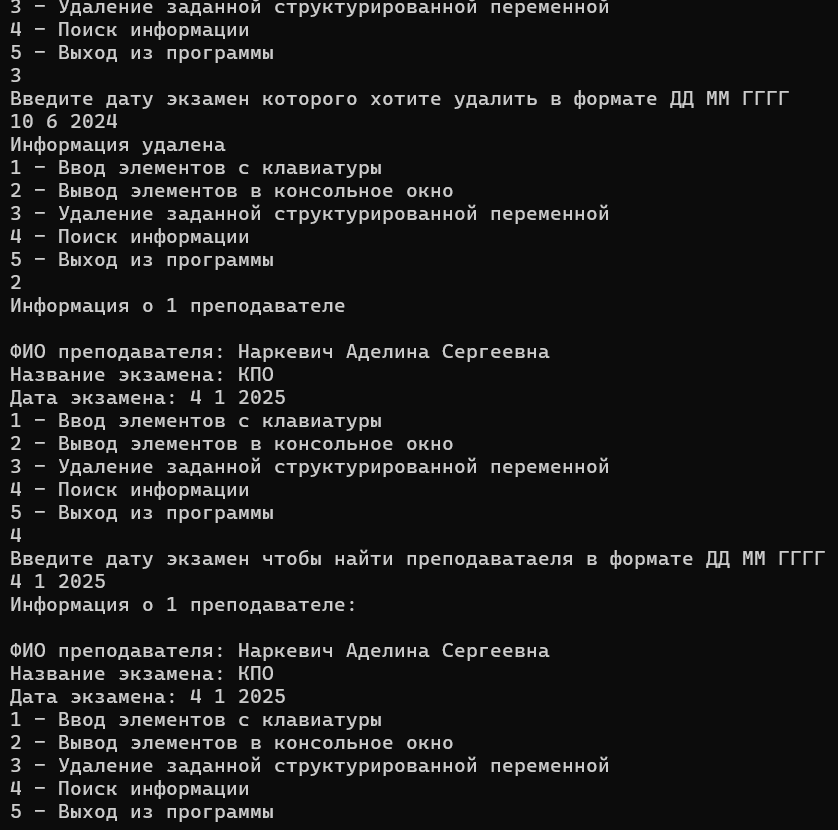
{

cout << "Преподаватель не найден!" << endl;

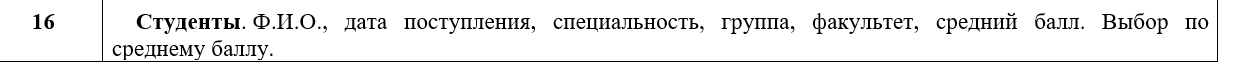
}

}









#include <iostream>

#include <fstream> // для работы с файлами

#include <string> // для работы со строками

#include <windows.h> // для корректного отображения символов

#define size 10 //подключение макроса для длины массива

using namespace std;

void input(); //прототип функции ввода

void output(); // прототип функции вывода

void inFile(); // прототип функции для записи в файл

void outFile(); // прототип функции для чтения из файла

void find(); // прототип функции поиска

union buf // объединение

{

char fullname[50];

char day[50];

char spec[50];

int gr;

char fac[50];

float av;

};

struct student // структура

{

buf full\_name;

buf entrday;

buf speciality;

buf group;

buf facultyd;

buf average;

};

student students[size];

int choice;

int current\_size;

int main()

{

setlocale(LC\_ALL, "Rus");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

//меню выбора

do

{

cout << "1 - Ввод элементов с клавиатуры" << endl;

cout << "2 - Вывод элементов в консольное окно" << endl;

cout << "3 - Запись информации в файл" << endl;

cout << "4 - Чтение информации из файла" << endl;

cout << "5 - Поиск информации" << endl;

cout << "6 - Выход из программы" << endl;

cin >> choice;

// анализируем choice

switch (choice)

{

case 1:

input();

break;

case 2:

output();

break;

case 3:

inFile();

break;

case 4:

outFile();

break;

case 5:

find();

break;

}

} while (choice != 6);

return 0;

}

void input() // функция ввода

{

int numb; // для хранения количества

int choice; // выбор из перечесления

cout << "Введите количество студентов: "; cin >> numb;

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

{

if (current\_size < size) // условие для проверки наличия места

{

// заполнение информации

cout << "Информация о " << i + 1 << " студенте: " << endl;

cin.ignore();

cout << endl << "ФИО: " << endl;

gets\_s(students[current\_size].full\_name.fullname, 50);

cout << "Дата поступления: " << endl;

gets\_s(students[current\_size].entrday.day, 50);

cout << "Специальность: " << endl;

gets\_s(students[current\_size].speciality.spec, 50);

cout << "Группа: " << endl;

cin >> students[current\_size].group.gr;

cin.ignore();

cout << "Факультет: " << endl;

gets\_s(students[current\_size].facultyd.fac, 50);

cout << "Средний балл: " << endl;

cin >> students[current\_size].average.av;

current\_size++;

}

}

}

void output() // функция вывода

{

// вывод информации

for (int i = 0; i < current\_size;i++)

{

cout << "Информация о " << i + 1 << " студенте: " << endl;

cout << endl << "ФИО: " << students[i].full\_name.fullname << endl;

cout << "Дата поступления: " << students[i].entrday.day << endl;

cout << "Специальность: " << students[i].speciality.spec << endl;

cout << "Группа: " << students[i].group.gr << endl;

cout << "Факультет: " << students[i].facultyd.fac << endl;

cout << "Средний балл: " << students[i].average.av << endl;

}

}

void inFile() // функция записи в файл

{

ofstream file1("FILE1.txt");

if (file1)

{

for (int i = 0; i < current\_size; i++)

{

file1 << "Информация о " << i + 1 << " студенте: " << endl;

file1 << endl << "ФИО: " << students[i].full\_name.fullname << endl;

file1 << "Дата поступления: " << students[i].entrday.day << endl;

file1 << "Специальность: " << students[i].speciality.spec << endl;

file1 << "Группа: " << students[i].group.gr << endl;

file1 << "Факультет: " << students[i].facultyd.fac << endl;

file1 << "Средний балл: " << students[i].average.av << endl;

}

cout << "Информация записана в файл!" << endl;

file1.close();

}

else

cout << "Не удалось открыть файл!" << endl;

}

void outFile() // функция чтения из файла

{

ifstream file1("FILE1.txt");

string line;

while (getline(file1, line))

{

cout << line << endl;

}

file1.close();

}

void find() // функция поиска

{

float aver;

bool f = true;

cout << "Введите средний балл: ";

cin >> aver;

for (int i = 0;i < current\_size;i++)

{

if (aver == students[i].average.av)

{

cout << "Информация о " << i + 1 << " студенте: " << endl;

cout << endl << "ФИО: " << students[i].full\_name.fullname << endl;

cout << "Дата поступления: " << students[i].entrday.day << endl;

cout << "Специальность: " << students[i].speciality.spec << endl;

cout << "Группа: " << students[i].group.gr << endl;

cout << "Факультет: " << students[i].facultyd.fac << endl;

cout << "Средний балл: " << students[i].average.av << endl;

f = false;

}

}

if (f)

{

cout << "Информация не найдена!" << endl;

}

}

