**Задание 3**

* 1. ***Условие задачи (13.40)***

Известны данные о 16-ти сотрудниках фирмы: фамилия, возраст и отношение к воинской службе (военнообязанный или нет). Определить:

1. Фамилию самого младшего по возрасту человека среди военнообязанных (считать, что такой есть и он единственный);
2. Фамилии самых старших по возрасту людей среди военнообязанных и среди невоеннообязанных (считать, что такие есть и они единственные в своей группе).
   1. ***Алгоритм выполнения программы:***

Создаем класс сотрудник, в который входят: его имя, возраст, отношение к воинской службе (военнообязанный или нет). Вводим в вектор класса информацию о сотрудниках. Находим среди военнообязанных младшего по возрасту, сравнивая возраст элементов вектора, если они военнообязанные. Находим среди всех старшего по возрасту, сравнивая возраст элементов вектора.

* 1. ***Листинг 3 – Программа «Сотрудники фирмы»***

#include <iostream>

#include <fstream>

#include <vector>

#include <ctime>

using namespace std;

class military {

string name;

int age;

bool opinion;

public:

military(string NAME, int AGE, int OPINION) {

name = NAME;

age = AGE;

opinion = OPINION;

}

string getName() {

return name;

}

int getAge() {

return age;

}

bool getOpinion() {

return opinion;

}

friend ostream& operator<<(ostream& out, military& digit) {

out << digit.name << " " << digit.age << " ";

if (digit.opinion) out << "Военнообязанный" << endl;

else out << "Не военнообязанный" << endl;

return out;

}

};

void generate() {

ofstream f("People.txt");

int\* c = new int[16];

int\* e = new int[16];

const char\* m[] = { "Grisha","Vadim","Kolya","Ivan","Dima","Anton","Kirill","Igor" };

for (int i = 0; i < 16; i++) {

c[i] = rand() % 80 + 10;

e[i] = rand() % 2;

f << m[rand() % 7] << " " << c[i] << " " << e[i] << endl;

}

}

int main() {

srand(static\_cast<unsigned int>(time(0)));

setlocale(0, "Ru");

generate();

ifstream file("People.txt");

vector<military> array;

string name;

int age;

bool opinion;

bool check = 0;

while (!file.eof()) {//вводим в вектор из файла

file >> name >> age >> opinion;

if (opinion) check = true;// проверяем наличие военнообязанных

military digit(name, age, opinion);

array.push\_back(digit);

}

array.erase(array.end() - 1);// удаляем последний элемент в векторе(там два последних повторяются)

vector<military> ::iterator iter = array.begin();

int i = 0;

int min=999;

int j;

if (check) {

while (iter != array.end())// ищем маленького военнообязанного

{

cout << array[i];

if (array[i].getOpinion()) {

if (array[i].getAge() < min) {

min = array[i].getAge();

j = i;

}

}

iter++;

i++;

}

cout << "\nСамый младший военнообязанный\n" << array[j].getName() << " " << array[j].getAge() << endl;

}

else cout << "\nНет военнообязанных\n";

iter = array.begin();

i = 0;

int max = 0;

while (iter != array.end()){ // ищем максимальный возраст

if (array[i].getAge() > max) {

max = array[i].getAge();

}

iter++;

i++;

}

cout << "\nСтаршие\n";

iter = array.begin();

i = 0;

while (iter != array.end()) {

if (array[i].getAge() == max) {

cout << array[i];

}

iter++;

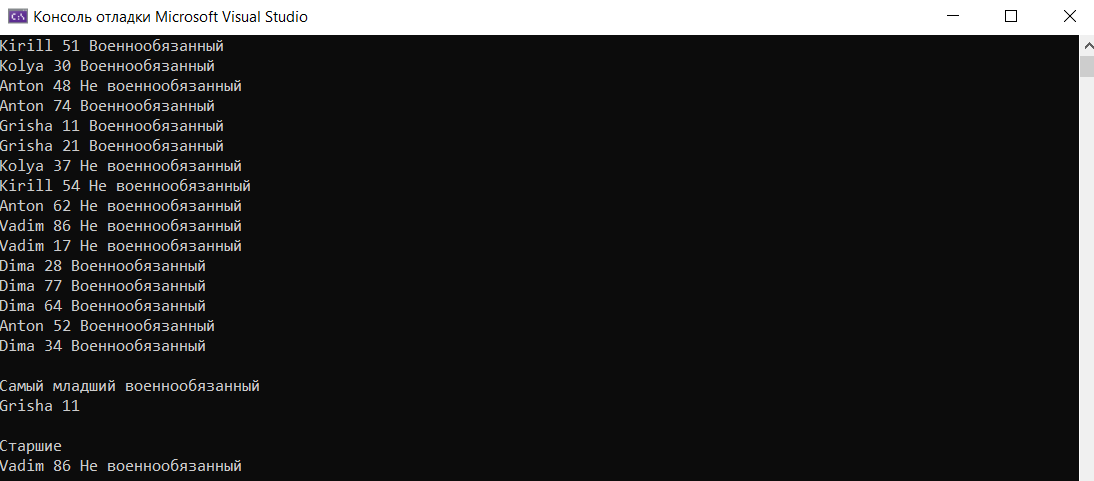
i++;

}

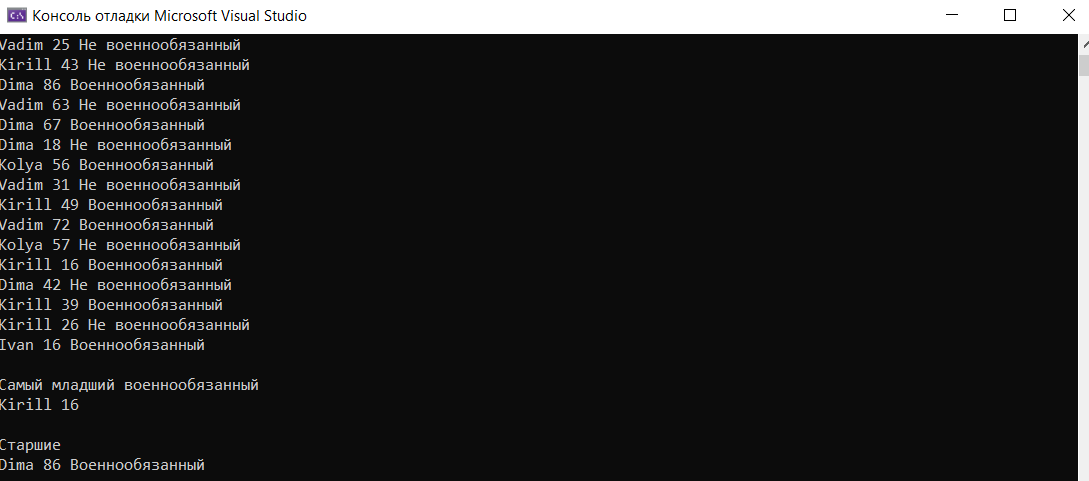
}

* 1. ***Контрольный тест:***

1)



2)



3)

