Министерство науки и высшего образования Российской Федерации

Федеральное агентство по образованию

КАЗАНСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. А.Н.Туполева - КАИ

Кафедра АСОИУ

Лабораторная работа № 3

по дисциплине

«Основы программирования»

……………………………………………………………………………………

РАБОТА С ТАБЛИЦАМИ

Обучающийся 4109 Сагиров А.И.

Принял старший препод. каф. АСОИУ Захарова З.Х.

Казань 2020

**Выполнение работы**

1. Введите и выполните программу из образца. Создайте файл st.txt для проверки программы и снова запустите программу.

Код программы:

#include <stdio.h>

#include <conio.h>

#include <string.h>

#include <locale.h>

#define MAX 30

struct STUDENT

{

char fio[15];

char oc[7];

};

struct EL\_TAB

{

char fio[15];

int oc[5];

float srball;

};

void PechTab (EL\_TAB \* tab, int n);

void Stud (EL\_TAB \* tab, int n);

void main()

{

setlocale(LC\_ALL, "Rus");

FILE \*f;

STUDENT tz;

EL\_TAB tab[MAX];

int n;

int i;

float s;

fopen\_s (&f, "st.txt", "r");

if (f == NULL)

{

puts ("Файл st.txt не найден");

return;

}

n = 0;

while (fgets((char \*)&tz, sizeof(struct STUDENT), f) != NULL)

{

for (i=0,s=0; i<5; i++)

{

int ocenka = tz.oc[i]-'0';

tab[n].oc[i] = ocenka;

s += ocenka;

}

tz.fio[14]='\0';

strcpy\_s (tab[n].fio, tz.fio);

tab[n++].srball = s/5;

}

fclose(f);

char nom;

do

{

puts ("------------------------------------------------------");

puts (" Выберите номер пункта меню:");

puts ("1 - средние баллы студентов");

puts ("2 - информация о заданном студенте");

puts ("3 - выход");

puts ("-------------------------------------------------------");

nom = \_getche();

switch (nom)

{

case '1': PechTab(tab,n); break;

case '2': Stud(tab,n); break;

case '3': putchar('\n');

break;

default: puts ("\nНужно ввести номер от 1 до 3");

}

if (nom != '3')

puts("\nДля продолжения нажмите любую клавишу");

}

while (nom != '3');

}

void PechTab (EL\_TAB \* tab, int n)

{

int j;

puts ("\nФамилия И.О. Ср.балл");

puts ("------------------------------");

for (j=0; j<n; j++)

{

printf("%s %.1f\n", tab[j].fio, tab[j].srball);

}

}

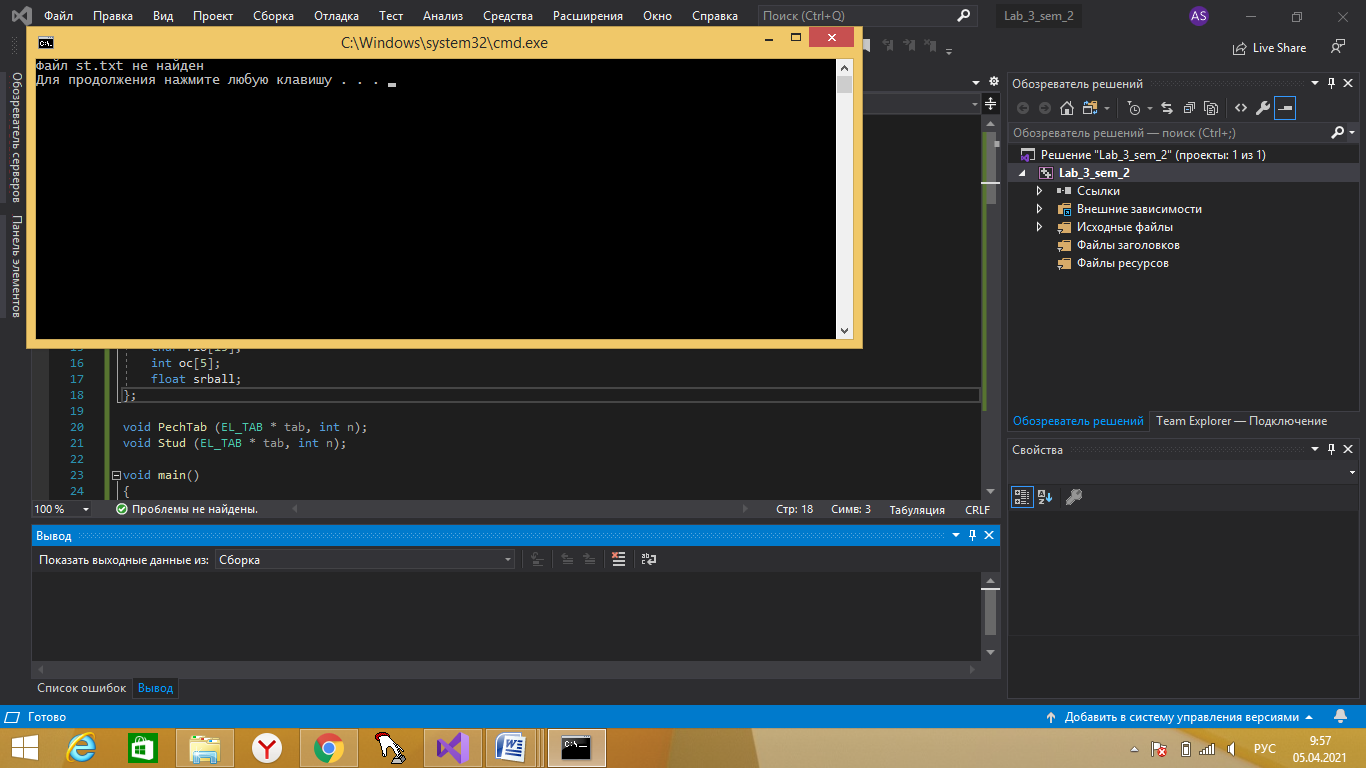
void Stud (EL\_TAB \* tab, int n)

{

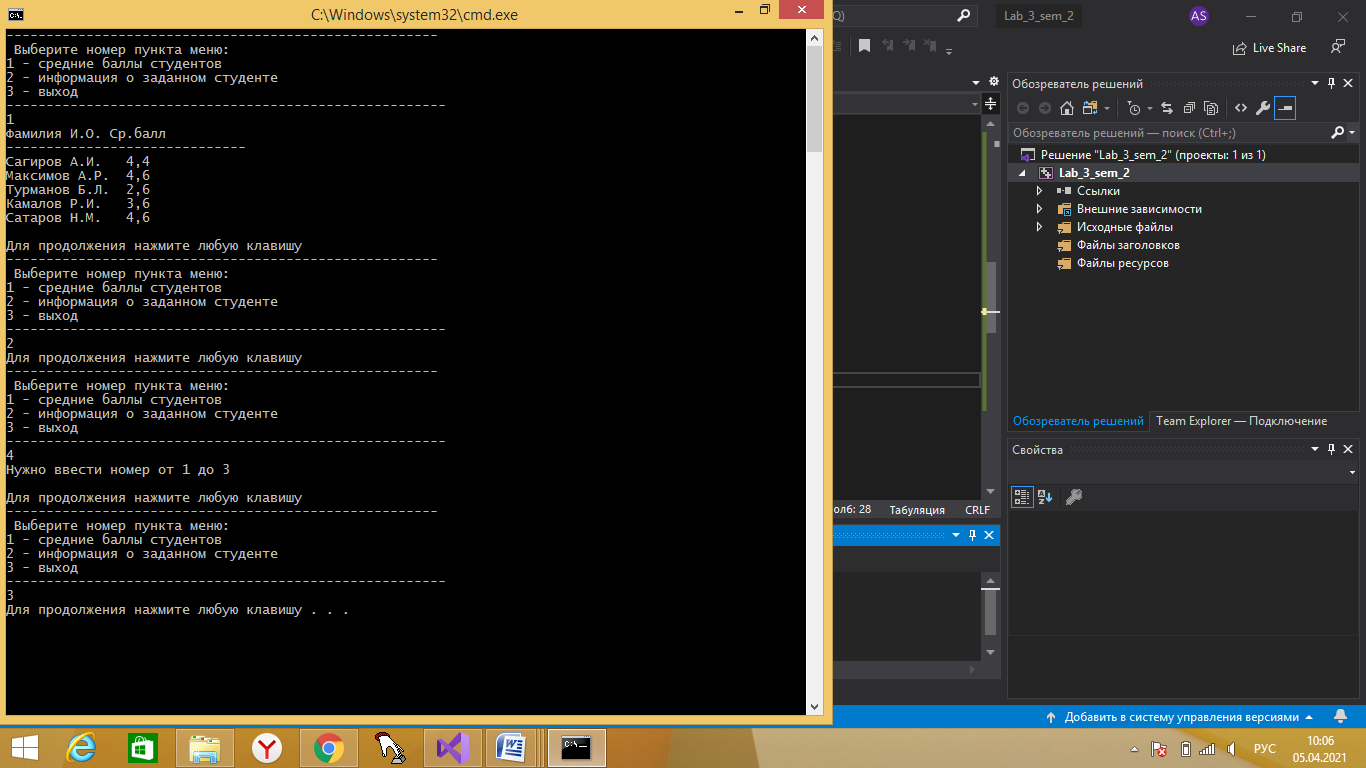
}

Результаты программы:

Без файла st.txt:



После создания файла st.txt:



1. Вместо заглушки функции Stud напишите функцию, которая запрашивает фамилию студента, выполняет линейный поиск элемента с заданной фамилией в таблице и выводит значение найденного элемента или сообщение об ошибке, если такой фамилии нет в таблице.

Код программы:

#include <stdio.h>

#include <conio.h>

#include <string.h>

#include <locale.h>

#define MAX 30

struct STUDENT

{

char fio[15];

char oc[7];

};

struct EL\_TAB

{

char fio[15];

int oc[5];

float srball;

};

void PechTab (EL\_TAB \* tab, int n);

void Stud (EL\_TAB \* tab, int n);

void main()

{

setlocale(LC\_ALL, "Rus");

FILE \*f;

STUDENT tz;

EL\_TAB tab[MAX];

int n;

int i;

float s;

fopen\_s (&f, "st.txt", "r");

if (f == NULL)

{

puts ("Файл st.txt не найден");

return;

}

n = 0;

while (fgets((char \*)&tz, sizeof(struct STUDENT), f) != NULL)

{

for (i=0,s=0; i<5; i++)

{

int ocenka = tz.oc[i]-'0';

tab[n].oc[i] = ocenka;

s += ocenka;

}

tz.fio[14]='\0';

strcpy\_s (tab[n].fio, tz.fio);

tab[n++].srball = s/5;

}

fclose(f);

char nom;

do

{

puts ("------------------------------------------------------");

puts (" Выберите номер пункта меню:");

puts ("1 - средние баллы студентов");

puts ("2 - информация о заданном студенте");

puts ("3 - выход");

puts ("-------------------------------------------------------");

nom = \_getche();

switch (nom)

{

case '1': PechTab(tab,n); break;

case '2': Stud(tab,n); break;

case '3': putchar('\n');

break;

default: puts ("\nНужно ввести номер от 1 до 3");

}

if (nom != '3')

puts("\nДля продолжения нажмите любую клавишу");

}

while (nom != '3');

}

void PechTab (EL\_TAB \* tab, int n)

{

int j;

puts ("\nФамилия И.О. Ср.балл");

puts ("------------------------------");

for (j=0; j<n; j++)

{

printf("%s %.1f\n", tab[j].fio, tab[j].srball);

}

}

void Stud (EL\_TAB \* tab, int n)

{

puts("\nВведите фамилию студента");

char zfio[15];

int i = 0;

bool found = false;

gets\_s(zfio);

for (; i < n; ++i)

{

if (strncmp(tab[i].fio, zfio, strlen(zfio)) == 0)

{

printf("%s %.1f\n", tab[i].fio, tab[i].srball);

found = true;

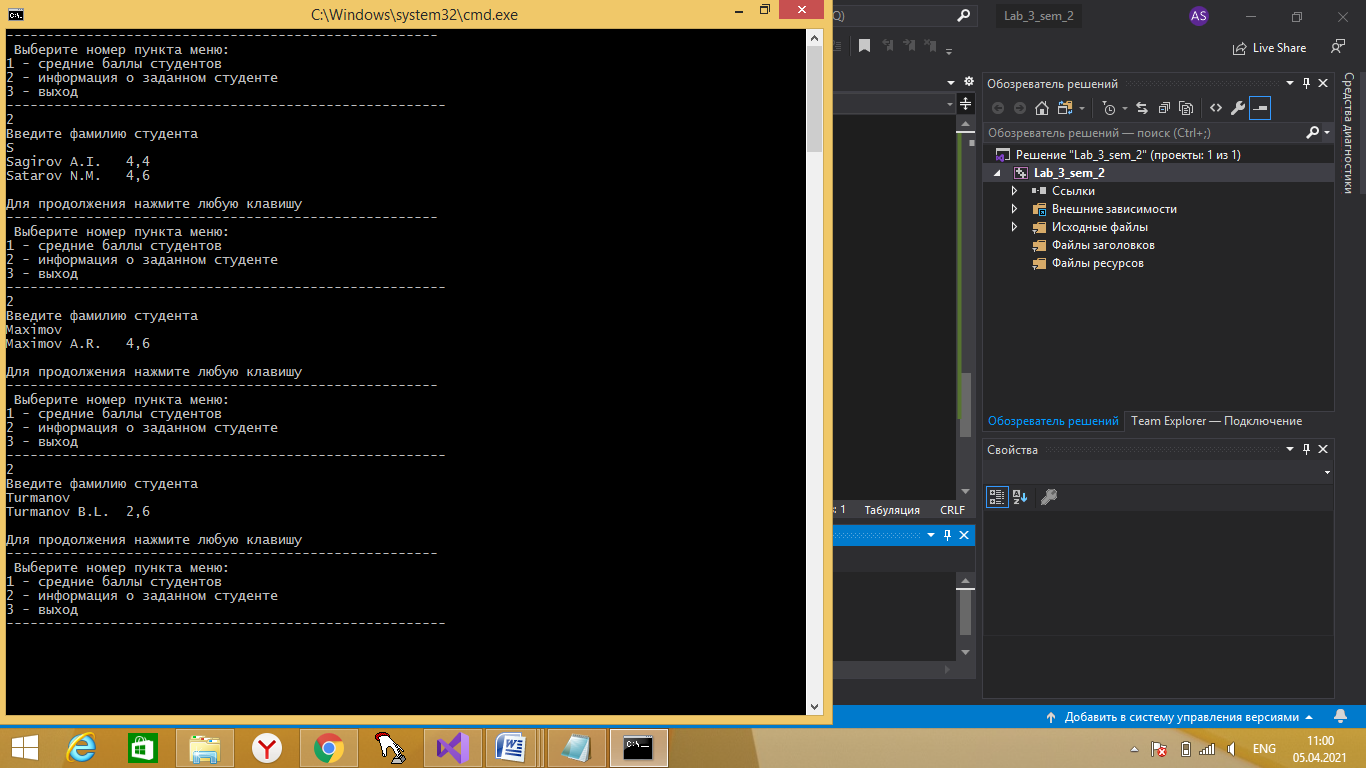
}

}

if(!found) puts("\nДанной фамилии нет в таблице");

}

Результаты программы:



1. Добавьте в программу подпрограмму сортировки таблицы в порядке убывания среднего балла.

Код программы:

#include <stdio.h>

#include <conio.h>

#include <string.h>

#include <locale.h>

#define MAX 30

struct STUDENT

{

char fio[15];

char oc[7];

};

struct EL\_TAB

{

char fio[15];

int oc[5];

float srball;

};

void PechTab (EL\_TAB \*tab, int n);

void Stud (EL\_TAB \*tab, int n);

void SortTab(EL\_TAB\* tab, int n);

void main()

{

setlocale(LC\_ALL, "Rus");

FILE \*f;

STUDENT tz;

EL\_TAB tab[MAX];

int n;

int i;

float s;

fopen\_s (&f, "st.txt", "r");

if (f == NULL)

{

puts ("Файл st.txt не найден");

return;

}

n = 0;

while (fgets((char \*)&tz, sizeof(struct STUDENT), f) != NULL)

{

for (i=0,s=0; i<5; i++)

{

int ocenka = tz.oc[i]-'0';

tab[n].oc[i] = ocenka;

s += ocenka;

}

tz.fio[14]='\0';

strcpy\_s (tab[n].fio, tz.fio);

tab[n++].srball = s/5;

}

fclose(f);

char nom;

do

{

puts ("------------------------------------------------------");

puts (" Выберите номер пункта меню:");

puts ("0 - сотировка баллов в порядке убывания");

puts ("1 - средние баллы студентов");

puts ("2 - информация о заданном студенте");

puts ("3 - выход");

puts ("-------------------------------------------------------");

nom = \_getche();

switch (nom)

{

case '0': SortTab(tab, n); break;

case '1': PechTab(tab,n); break;

case '2': Stud(tab,n); break;

case '3': putchar('\n');

break;

default: puts ("\nНужно ввести номер от 1 до 3");

}

if (nom != '3')

puts("\nДля продолжения нажмите любую клавишу");

}

while (nom != '3');

}

void PechTab (EL\_TAB \* tab, int n)

{

int j;

puts ("\nФамилия И.О. Ср.балл");

puts ("------------------------------");

for (j=0; j<n; j++)

{

printf("%s %.1f\n", tab[j].fio, tab[j].srball);

}

}

void Stud (EL\_TAB \* tab, int n)

{

puts("\nВведите фамилию студента");

char zfio[15];

int i = 0;

bool found = false;

gets\_s(zfio);

for (; i < n; ++i)

{

if (strncmp(tab[i].fio, zfio, strlen(zfio)) == 0)

{

printf("%s %.1f\n", tab[i].fio, tab[i].srball);

found = true;

}

}

if(!found) puts("\nДанной фамилии нет в таблице");

}

void SortTab(EL\_TAB\* tab, int n)

{

int k, i;

EL\_TAB s;

for (k = --n; k > 0; --k)

for (i = 0; i < k; ++i)

if (tab[i].srball < tab[i+1].srball)

{

s = tab[i];

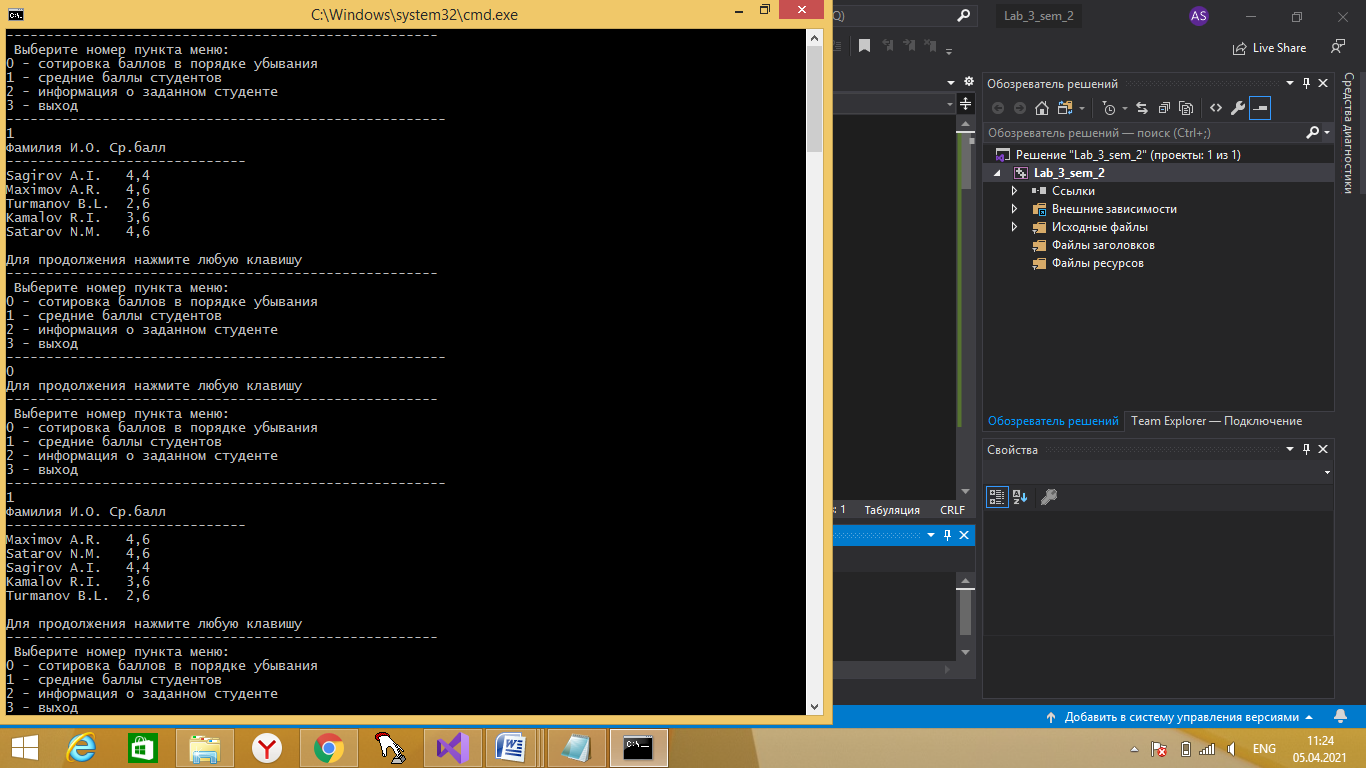
tab[i] = tab[i+1];

tab[i+1] = s;

}

}

Результаты программы:



1. Добавьте подпрограмму вывода:

г) студентов, у которых нет 3 и 2;

Код программы:

#include <stdio.h>

#include <conio.h>

#include <string.h>

#include <locale.h>

#define MAX 30

struct STUDENT

{

char fio[15];

char oc[7];

};

struct EL\_TAB

{

char fio[15];

int oc[5];

float srball;

};

void PechTab (EL\_TAB \*tab, int n);

void Stud (EL\_TAB \*tab, int n);

void SortTab(EL\_TAB\* tab, int n);

void Print\_without\_2\_3(EL\_TAB\* tab, int n);

void main()

{

setlocale(LC\_ALL, "Rus");

FILE \*f;

STUDENT tz;

EL\_TAB tab[MAX];

int n;

int i;

float s;

fopen\_s (&f, "st.txt", "r");

if (f == NULL)

{

puts ("Файл st.txt не найден");

return;

}

n = 0;

while (fgets((char \*)&tz, sizeof(struct STUDENT), f) != NULL)

{

for (i=0,s=0; i<5; i++)

{

int ocenka = tz.oc[i]-'0';

tab[n].oc[i] = ocenka;

s += ocenka;

}

tz.fio[14]='\0';

strcpy\_s (tab[n].fio, tz.fio);

tab[n++].srball = s/5;

}

fclose(f);

char nom;

do

{

puts ("------------------------------------------------------");

puts (" Выберите номер пункта меню:");

puts ("0 - сотировка баллов в порядке убывания");

puts ("1 - средние баллы студентов");

puts ("2 - информация о заданном студенте");

puts ("3 - вывести студентов без 2 и 3");

puts ("4 - выход");

puts ("-------------------------------------------------------");

nom = \_getche();

switch (nom)

{

case '0': SortTab(tab, n); break;

case '1': PechTab(tab,n); break;

case '2': Stud(tab,n); break;

case '3': putchar('\n'); Print\_without\_2\_3(tab, n); break;

case '4': putchar('\n');

break;

default: puts ("\nНужно ввести номер от 1 до 3");

}

if (nom != '4')

puts("\nДля продолжения нажмите любую клавишу");

}

while (nom != '4');

}

void PechTab (EL\_TAB \* tab, int n)

{

int j;

puts ("\nФамилия И.О. Ср.балл");

puts ("------------------------------");

for (j=0; j<n; j++)

{

printf("%s %.1f\n", tab[j].fio, tab[j].srball);

}

}

void Stud (EL\_TAB \* tab, int n)

{

puts("\nВведите фамилию студента");

char zfio[15];

int i = 0;

bool found = false;

gets\_s(zfio);

for (; i < n; ++i)

{

if (strncmp(tab[i].fio, zfio, strlen(zfio)) == 0)

{

printf("%s %.1f\n", tab[i].fio, tab[i].srball);

found = true;

}

}

if(!found) puts("\nДанной фамилии нет в таблице");

}

void SortTab(EL\_TAB\* tab, int n)

{

int k, i;

EL\_TAB s;

for (k = --n; k > 0; --k)

for (i = 0; i < k; ++i)

if (tab[i].srball < tab[i+1].srball)

{

s = tab[i];

tab[i] = tab[i+1];

tab[i+1] = s;

}

}

void Print\_without\_2\_3(EL\_TAB\* tab, int n)

{

bool print;

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

{

print = true;

for (int j = 0; j < 5; ++j)

{

if ((tab[i].oc[j] == 2) || (tab[i].oc[j] == 3))

{

print = false;

break;

}

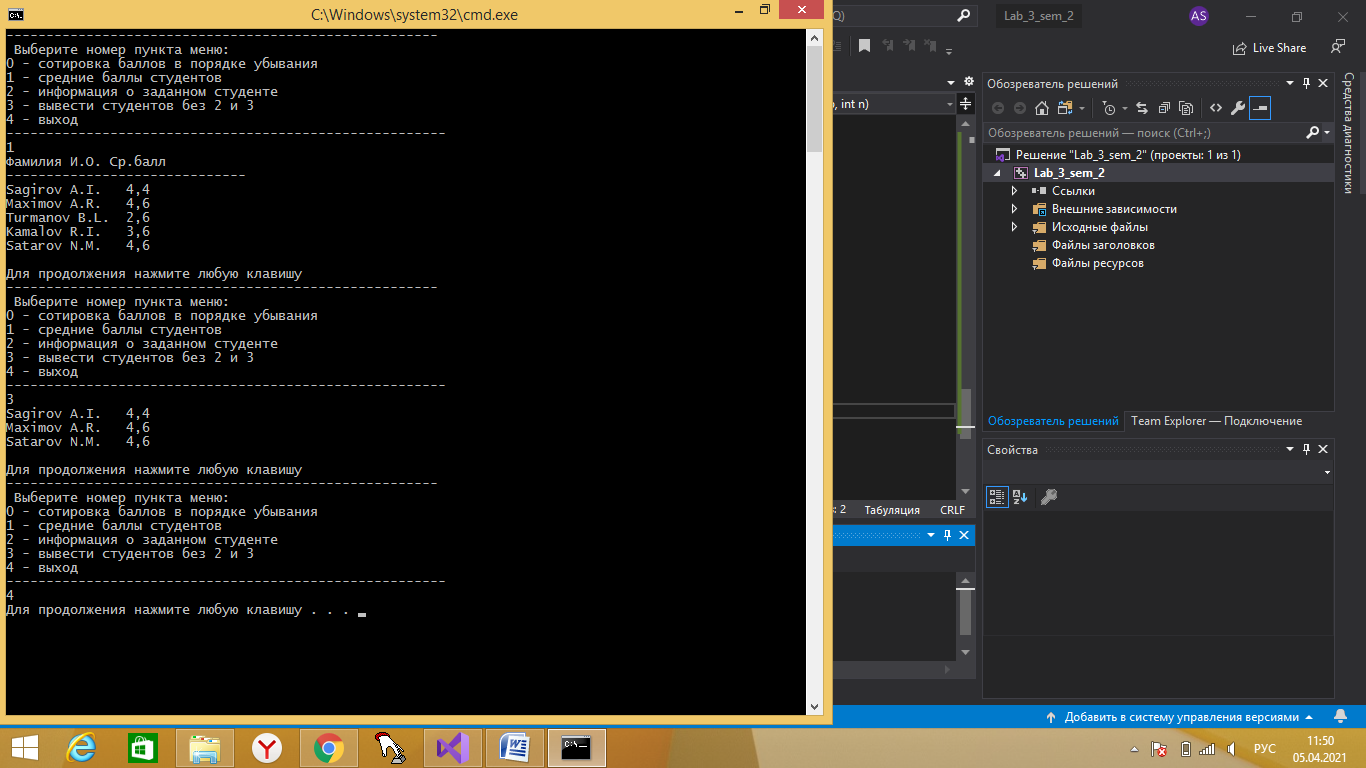
}

if(print) printf("%s %.1f\n", tab[i].fio, tab[i].srball);

}

}

Результаты программы:



Ответы на контрольные вопросы:

1. Таблица – это набор элементов, содержащих ключ – отличительный признак для поиска элементов, и тело – сопутствующую информацию.
2. Последовательной (линейной) называют таблицу, в которой производится линейный поиск, т.е. последовательный перебор элементов с начала таблицы. Последовательные таблицы бывают сортированными и несортированными. Внутренние таблицы (создаваемые в оперативной памяти) могут храниться в виде вектора или в виде списка.
3. Инициализация (подготовка к работе), поиск элемента по ключу, включение в таблицу данного элемента, распечатка элементов таблицы в порядке, определяемом ключами.

struct EL\_TAB

{

сhar month[10];

int quality;

};

EL\_TAB tab[12];



struct EL\_TAB

{

сhar fio[15];

int year;

double height;

double weight;

};

EL\_TAB tab[50];

struct EL\_TAB

{

int number;

char destination[30];

char time[6];

};

EL\_TAB tab[100];