

МИНОБРНАУКИ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ
БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет прикладной математики, информатики и механики
Кафедра математического обеспечения ЭВМ

КУРСОВАЯ РАБОТА
по дисциплине «Информатика и программирование»

Направление 02.03.02 – Фундаментальная информатика
и информационные технологии
Профиль – Инженерия программного обеспечения (ФГОС3++)

Зав. кафедрой, доктор техн. наук, проф. _____ Г.В.Абрамов

Обучающийся, студент 1 курса _____ И.Ю.Смехнев

Руководитель, канд. физ.-мат. наук, доц. _____ О.Д.Горбенко

Воронеж 2022

Содержание

Постановка задачи	3
Описание данных и алгоритма решения задачи.....	4
Условие с динамическими массивами.....	8
Условие с функцией	11
Описание структуры программы.....	14
Интерфейс:	14
Структура программы:	14
Результаты тестирования	15
Тест 1.	15
Тест 2.	15
Список использованной литературы	17
Приложение	18

Постановка задачи

С клавиатуры вводится информация об итогах последней экзаменационной сессии. Эта информация включает в себя:

- 1) целое число n – количество студентов;
- 2) n объединенных в структуру данных: <имя> <фамилия> <оценка> <оценка>
<оценка> <оценка> <оценка>

где <имя>, <фамилия> — символьные строки, содержащие не более 20 символов, оценка за экзамен — десятичная цифра из диапазона '2'..'5'.

Требуется сформировать массив структур, в котором каждый элемент массива содержит фамилию студента и его средний балл, причем вначале размещаются данные о студентах, не имеющих отличных оценок, затем об остальных студентах, при этом в каждой части сохраняется взаимный порядок следования данных, определенный вводом, и вывести этот массив на экран. Вывести также фамилии и оценки студентов, имеющих наибольшее количество отличных оценок.

Технические требования.

- 1) В основной памяти исходные данные также следует хранить в виде массива структур.
- 2) Не допускается использование дополнительных массивов.
- 3) Результатом курсовой работы должны стать три программы, где:
 - а) В первой – решение поставленной задачи осуществляется с использованием традиционного подхода к объявлению и созданию статического массива данных;
 - б) Во второй – используются языковые инструменты создания и объявления динамических массивов данных;
 - с) В третьей – для реализации одного из многошаговых действий следует определить и применить функцию.

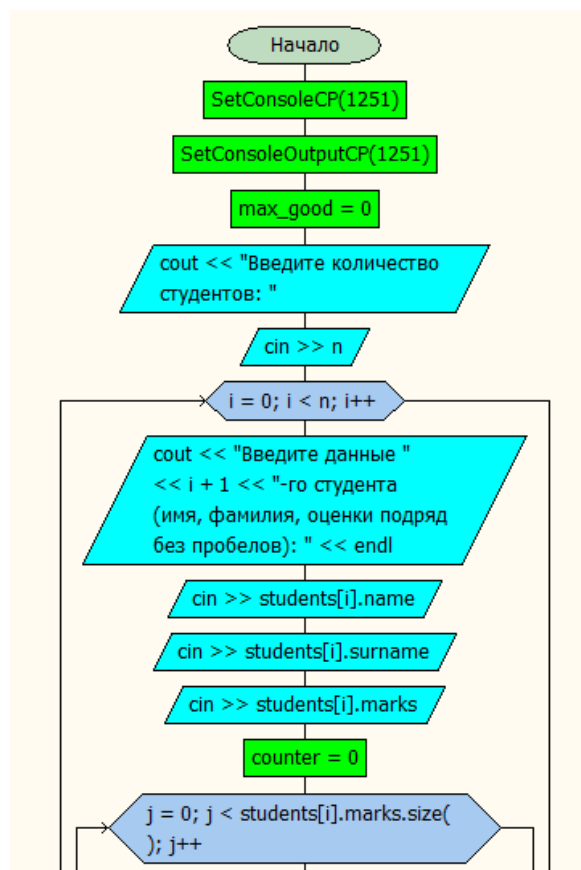
Описание данных и алгоритма решения задачи

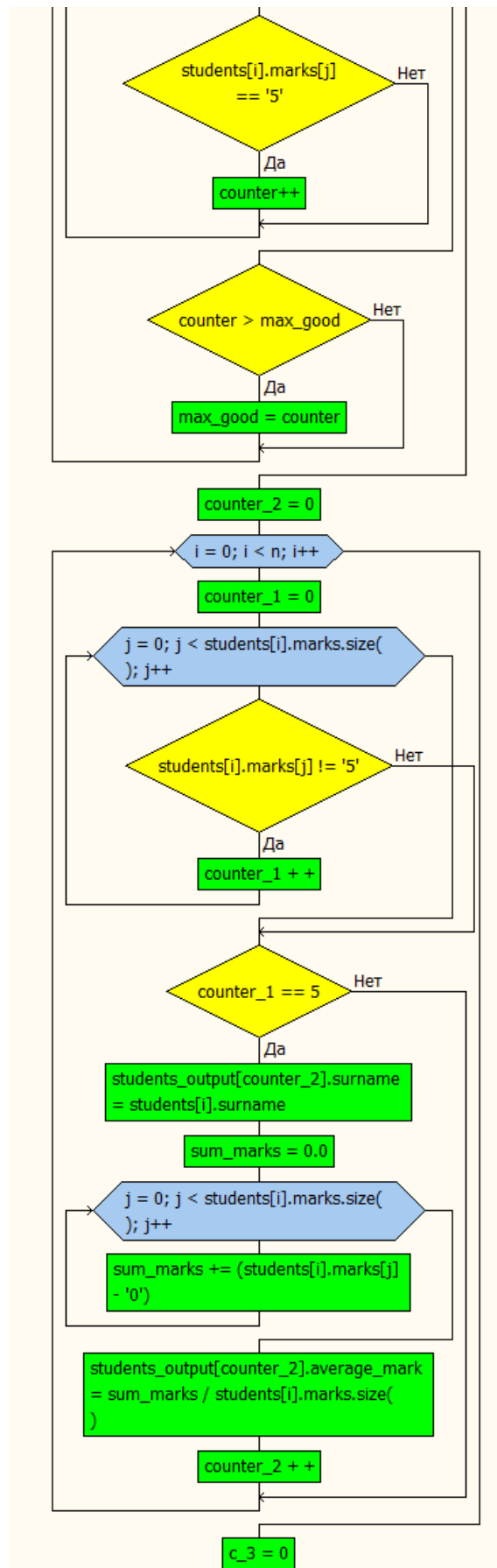
Для реализации решения задачи использовались следующие простые и структурированные данные:

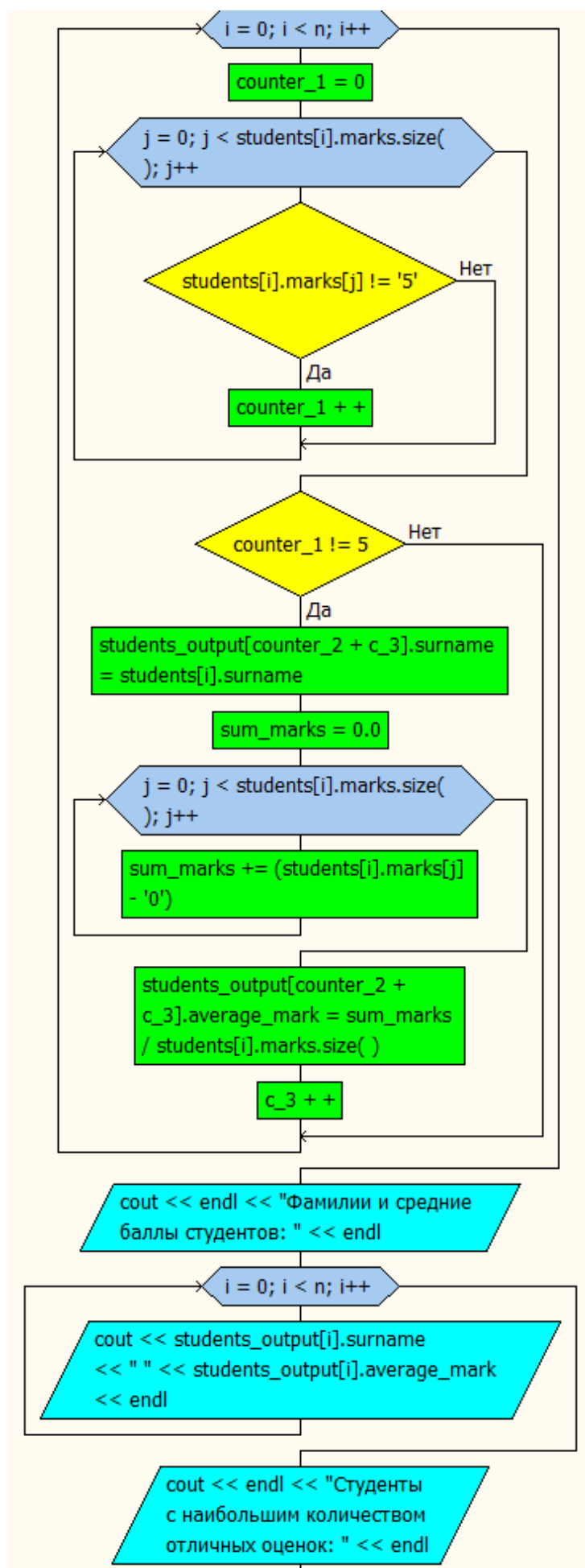
- **student** – структурный тип, в котором хранятся вводимые данные о студенте. Поле *name* хранит имя студента, *surname* – фамилию, *marks* – оценки, полученные за сессию.
- **student_output** – структурный тип, в котором хранятся выводимые данные о студенте. Поле *surname* хранит фамилию студента, *average_mark* – среднюю оценку.
- Далее были использованы массивы структурных типов **student** и **student_output**, объявленные по-разному в зависимости от поставленной задачи:
 - Для условия со статическими массивами: **student** *studs*[100] и **student_output** *studs_output*[100], в каждом из которых заполнялись первые *n* элементов, *n* – вводимое число студентов.
 - Для условия с динамическими массивами:
 - **student*** *studs* = new **student**[*n*]
 - **student_output*** *studs_output* = new **student_output**[*n*]
 - *n* – вводимое число студентов
 - Для условия с функцией: те же динамические массивы, только для одного из действий была использована функция.

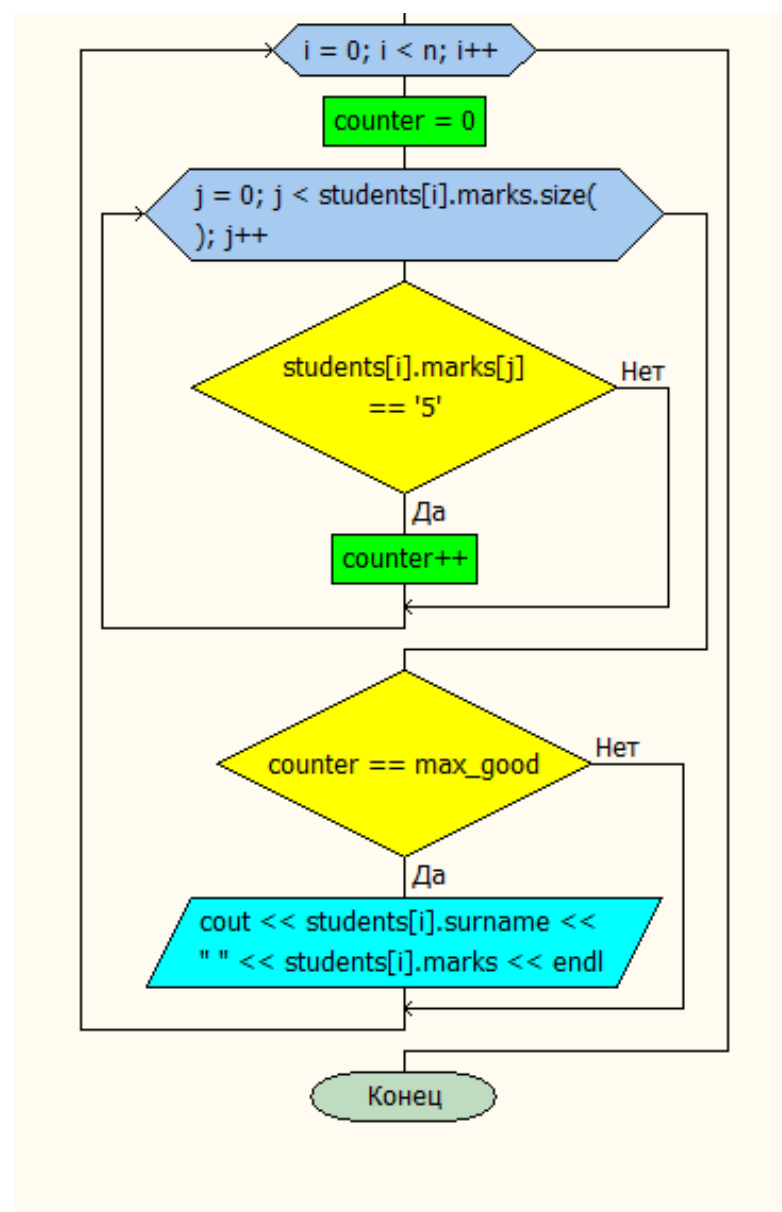
Для наглядности представим алгоритмы всех трех частей задачи в виде блок-схем.

Условие со статическим массивом

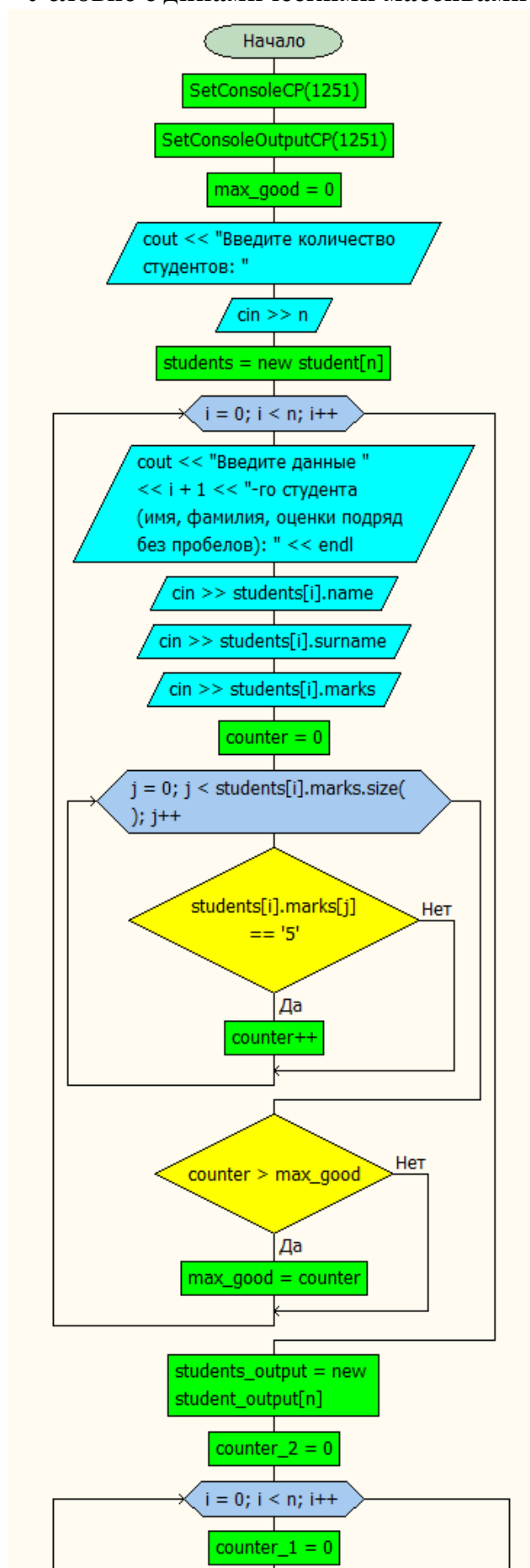


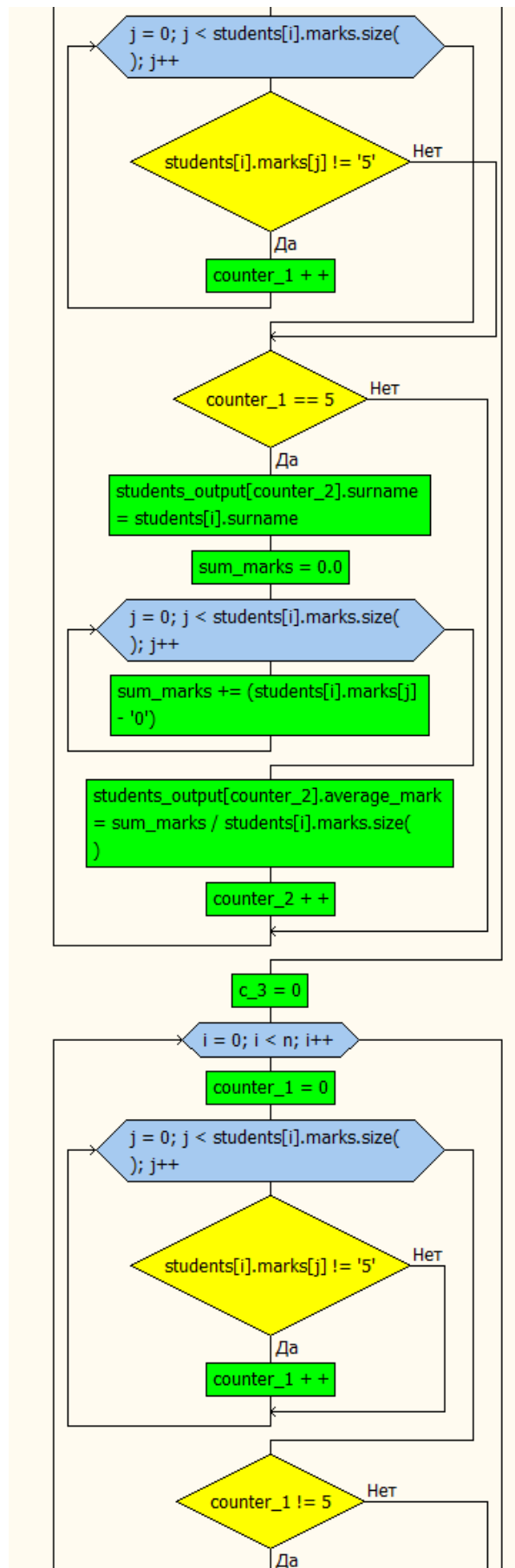


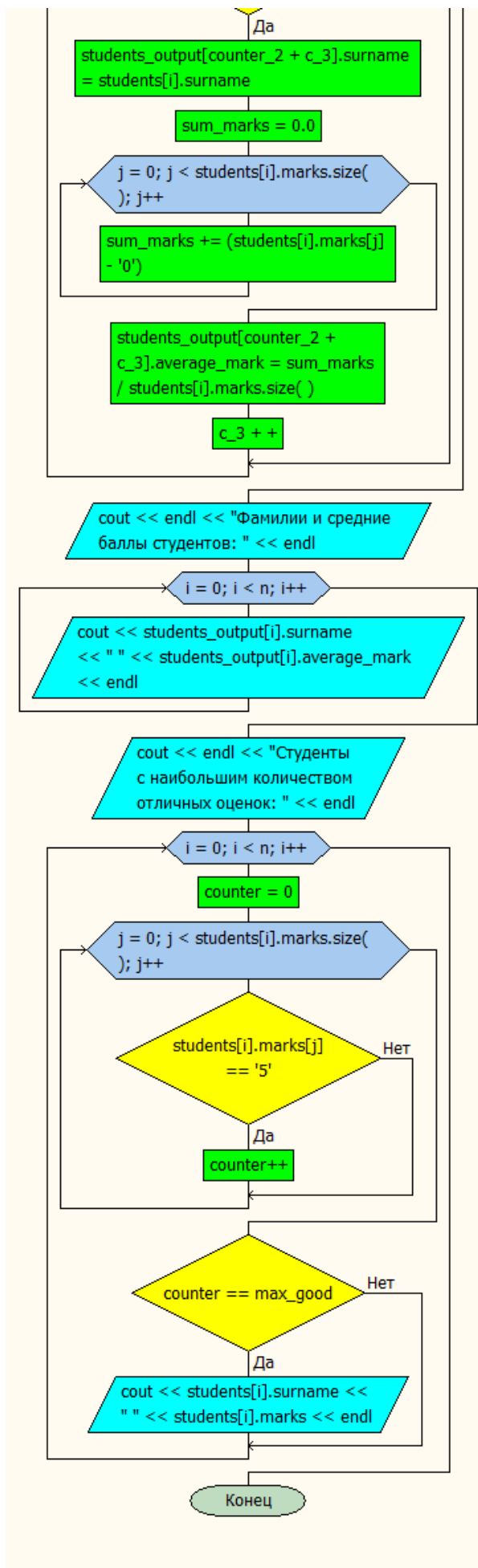




Условие с динамическими массивами

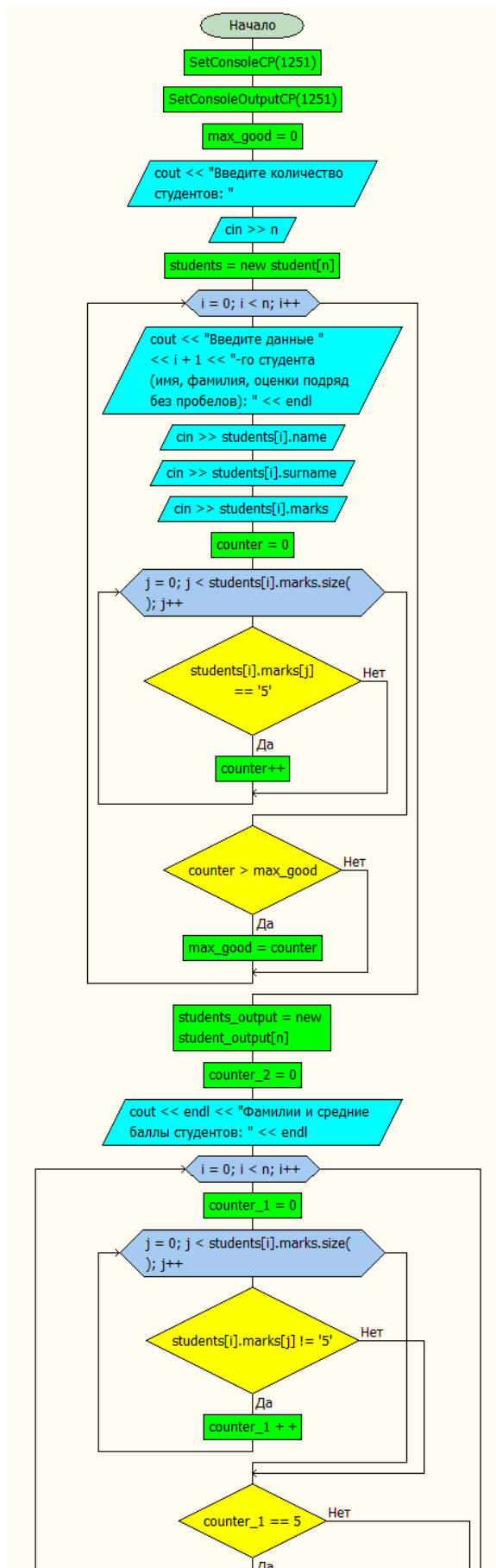


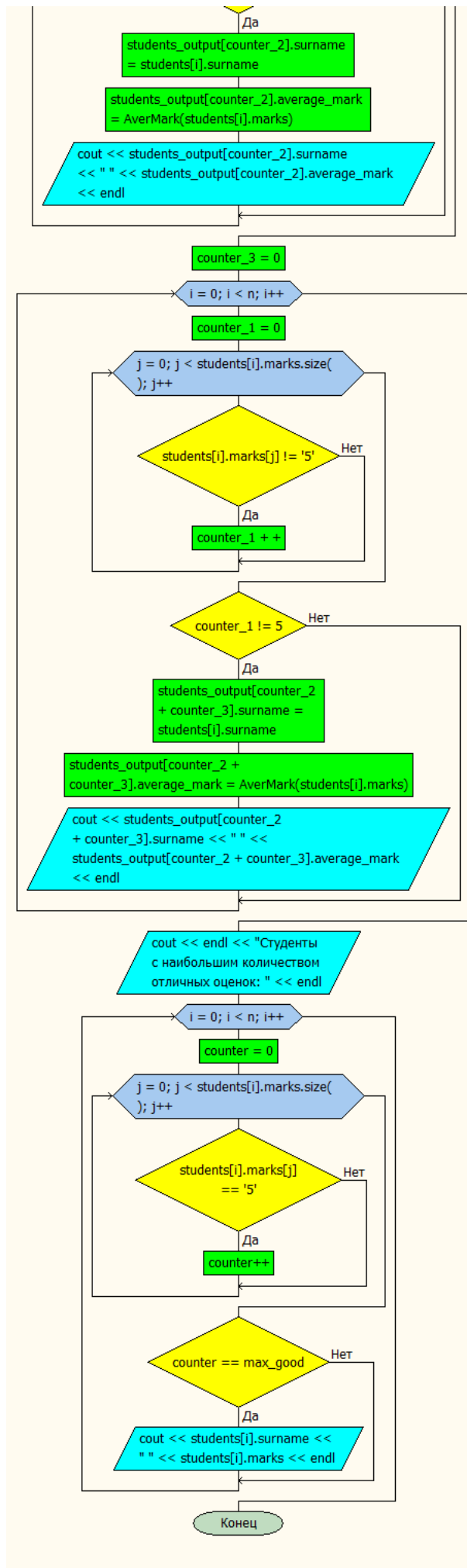




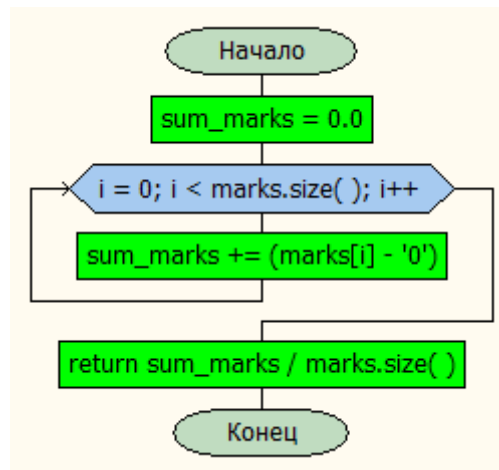
Условие с функцией

int main():





double AverMark(string& marks)



Описание структуры программы

Интерфейс:

1. Запрос на ввод количества студентов: «Введите количество студентов: ». Вводимое значение – n ;
2. Запрос на ввод информации о студентах: «Введите данные (номер студента)-го студента (имя, фамилия, оценки подряд без пробелов): ». Вводимое значение – строка вида <имя> <фамилия> <оценка><оценка><оценка><оценка><оценка>

Структура программы:

Каждая задача разбита на блоки:

1. Создание входного массива, ввод данных
2. Создание выходного массива, в котором каждый элемент массива содержит фамилию студента и его средний балл, причем вначале размещаются данные о студентах, не имеющих отличных оценок, затем об остальных студентах, при этом в каждой части сохраняется взаимный порядок следования данных, определенный вводом, его заполнение (в условии с функцией именно здесь используется функция *AverMark*, вычисляющая и возвращающая значение средней оценки)
3. Вывод выходного массива
4. Вывод данных, полученных при выполнении дополнительного условия задачи (вывести на экран фамилии и оценки студентов, получивших наибольшее количество отличных оценок)

Результаты тестирования

Тест 1.

Входные данные при $n = 2$:

Введите количество студентов: 2

Введите данные 1-го студента (имя, фамилия, оценки подряд без пробелов):

Никита Уразов 24345

Введите данные 2-го студента (имя, фамилия, оценки подряд без пробелов):

Иван Смехнев 22344

Результат выполнения программы:

Фамилии и средние баллы студентов:

Смехнев 3

Уразов 3.6

Студенты с наибольшим количеством отличных оценок:

Уразов 24345

Тест 2.

Входные данные при $n = 5$:

Введите количество студентов: 5

Введите данные 1-го студента (имя, фамилия, оценки подряд без пробелов):

Никита Уразов 24345

Введите данные 2-го студента (имя, фамилия, оценки подряд без пробелов):

Иван Смехнев 22344

Введите данные 3-го студента (имя, фамилия, оценки подряд без пробелов):

Всеволод Брянцев 34455

Введите данные 4-го студента (имя, фамилия, оценки подряд без пробелов):

Дмитрий Оболонский 23344

Введите данные 5-го студента (имя, фамилия, оценки подряд без пробелов):

Артём Петросян 23455

Результат выполнения программы:

Фамилии и средние баллы студентов:

Смехнев 3

Оболонский 3.2

Уразов 3.6

Брянцев 4.2

Петросян 3.8

Студенты с наибольшим количеством отличных оценок:

Брянцев 34455

Петросян 23455

Список использованной литературы

1. О.Ф.Ускова, Н.А.Каплиева, О.Д.Горбенко. Информатика и программирование. Задачник-практикум по структурному программированию на языке C++. – Воронеж, Издательский дом ВГУ,- 2020.
2. C/C++. Программирование на языке высокого уровня / Т.А. Павловская — СПб.: Питер, 2011 .
3. Ашарина, И.В. Основы программирования на языках C и C++: Курс лекций для высших учебных заведений / И.В. Ашарина. — М.: Гор. линия-Телеком, 2018. — 208 с.
4. Дэвис, С. C++ для «чайников» / С. Дэвис. – Киев : Диалектика, 1996
5. Мейерс Скотт. Эффективный и современный C++: 42 рекомендации по использованию C++11 и C++14. Издательство Вильямс. 2019

Приложение

Файл ConsoleApplicaton1.cpp

```
/*
Смехнев Иван Юрьевич, 61 группа
Условие 1: со статическими массивами
С клавиатуры вводится информация об итогах последней экзаменационной сессии. Эта
информация включает в себя: 1) целое число n – количество студентов; 2) n объединенных
в структуру данных: где , — символьные строки, содержащие не более 20 символов,
оценка за экзамен — десятичная цифра из диапазона '2'..'5'. Требуется сформировать
массив структур, в котором каждый элемент массива содержит фамилию студента и его
средний балл, причем вначале размещаются данные о студентах, не имеющих отличных
оценок, затем об остальных студентах, при этом в каждой части сохраняется взаимный
порядок следования данных, определенный вводом, и вывести этот массив на экран.
Вывести также фамилии и оценки студентов, имеющих наибольшее количество отличных
оценок.
*/

#include <iostream>
#include <Windows.h>
#include <string>

using namespace std;

struct student //структура, в которой будет храниться входная информация о студенте (имя,
фамилия, оценки)
{
    string name;
    string surname;
    string marks;
};

struct student_output //структура, в которой будет храниться выходная информация о
студенте (фамилия) и его средняя оценка
{
    string surname;
    double average_mark = 0.0;
};

int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);

    student studs[100]; //входной массив структур
    int n; //количество студентов
    int max_good = 0; // максимально количество отличных оценок

    //ввод данных
    cout << "Введите количество студентов: "; cin >> n;
```

```

for (int i = 0; i < n; i++)
{
    cout << "Введите данные " << i + 1 << "-го студента (имя, фамилия, оценки подряд без пробелов): " << endl;
    cin >> studs[i].name;
    cin >> studs[i].surname;
    cin >> studs[i].marks;
    int counter = 0; // счетчик отличных оценок
    for (int j = 0; j < studs[i].marks.size(); j++)
    {
        if (studs[i].marks[j] == '5')
        {
            counter++;
        }
    }
    if (counter > max_good)
    {
        max_good = counter;
    }
}

```

```

student_output studs_output[100]; //выходной массив структур
int counter_1; //счетчик оценок без отлично
int counter_2 = 0; //счетчик студентов, не имеющих отличных студентов

```

//заполнение выходного массива студентами, не имеющих отличных оценок

```

for (int i = 0; i < n; i++)
{
    counter_1 = 0;
    for (int j = 0; j < studs[i].marks.size(); j++)
    {
        if (studs[i].marks[j] != '5')
        {
            counter_1++;
        }
        else break;
    }
    if (counter_1 == 5)
    {
        studs_output[counter_2].surname = studs[i].surname;
        double sum_marks = 0.0;
        for (int j = 0; j < studs[i].marks.size(); j++)
        {
            sum_marks += (studs[i].marks[j] - '0');
        }
        studs_output[counter_2].average_mark = sum_marks / studs[i].marks.size();
        counter_2++;
    }
}

```

// заполнение выходного массива остальными студентами

```

int c_3 = 0;

```

```

for (int i = 0; i < n; i++)
{
    counter_1 = 0;
    for (int j = 0; j < studs[i].marks.size(); j++)
    {
        if (studs[i].marks[j] != '5')
        {
            counter_1++;
        }
    }
    if (counter_1 != 5)
    {
        studs_output[counter_2 + c_3].surname = studs[i].surname;
        double sum_marks = 0.0;
        for (int j = 0; j < studs[i].marks.size(); j++)
        {
            sum_marks += (studs[i].marks[j] - '0');
        }
        studs_output[counter_2 + c_3].average_mark = sum_marks / studs[i].marks.size();
        c_3++;
    }
}

//ВЫВОД ВЫХОДНОГО МАССИВА
cout << endl << "Фамилии и средние баллы студентов: " << endl;
for (int i = 0; i < n; i++)
{
    cout << studs_output[i].surname << " " << studs_output[i].average_mark << endl;
}

//ВЫВОД ДОПОЛНИТЕЛЬНОГО УСЛОВИЯ
cout << endl << "Студенты с наибольшим количеством отличных оценок: " << endl;
for (int i = 0; i < n; i++)
{
    int counter = 0;
    for (int j = 0; j < studs[i].marks.size(); j++)
    {
        if (studs[i].marks[j] == '5')
        {
            counter++;
        }
    }
    if (counter == max_good)
    {
        cout << studs[i].surname << " " << studs[i].marks << endl;
    }
}
}

```

Файл ConsoleApplicaton2.cpp

```
/*
Смехнев Иван Юрьевич, 61 группа
Условие 2: с динамическими массивами
С клавиатуры вводится информация об итогах последней экзаменационной сессии. Эта
информация включает в себя: 1) целое число n – количество студентов; 2) n объединенных
в структуру данных: где , — символьные строки, содержащие не более 20 символов,
оценка за экзамен — десятичная цифра из диапазона '2'..'5'. Требуется сформировать
массив структур, в котором каждый элемент массива содержит фамилию студента и его
средний балл, причем вначале размещаются данные о студентах, не имеющих отличных
оценок, затем об остальных студентах, при этом в каждой части сохраняется взаимный
порядок следования данных, определенный вводом, и вывести этот массив на экран.
Вывести также фамилии и оценки студентов, имеющих наибольшее количество отличных
оценок.
*/

#include <iostream>
#include <Windows.h>
#include <string>

using namespace std;

struct student //структура, в которой будет храниться входная информация о студенте (имя,
фамилия, оценки)
{
    string name;
    string surname;
    string marks;
};

struct student_output //структура, в которой будет храниться выходная информация о
студенте (фамилия) и его средняя оценка
{
    string surname;
    double average_mark = 0.0;
};

int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);

    int n; //количество студентов
    int max_good = 0; // максимально количество отличных оценок

    //ввод данных
    cout << "Введите количество студентов: "; cin >> n;
    student* studs = new student[n]; //входной массив структур
    for (int i = 0; i < n; i++)
    {
```

```

        cout << "Введите данные " << i + 1 << "-го студента (имя, фамилия, оценки подряд без пробелов): " << endl;
        cin >> studs[i].name;
        cin >> studs[i].surname;
        cin >> studs[i].marks;
        int counter = 0; // счетчик отличных оценок
        for (int j = 0; j < studs[i].marks.size(); j++)
        {
            if (studs[i].marks[j] == '5')
            {
                counter++;
            }
        }
        if (counter > max_good)
        {
            max_good = counter;
        }
    }

```

```

student_output* studs_output = new student_output[n]; //выходной массив структур
int counter_1; //счетчик оценок без отлично
int counter_2 = 0; //счетчик студентов, не имеющих отличных студентов

```

```

//заполнение выходного массива студентами, не имеющих отличных оценок

```

```

for (int i = 0; i < n; i++)
{
    counter_1 = 0;
    for (int j = 0; j < studs[i].marks.size(); j++)
    {
        if (studs[i].marks[j] != '5')
        {
            counter_1++;
        }
        else break;
    }
    if (counter_1 == 5)
    {
        studs_output[counter_2].surname = studs[i].surname;
        double sum_marks = 0.0;
        for (int j = 0; j < studs[i].marks.size(); j++)
        {
            sum_marks += (studs[i].marks[j] - '0');
        }
        studs_output[counter_2].average_mark = sum_marks / studs[i].marks.size();
        counter_2++;
    }
}

```

```

// заполнение выходного массива остальными студентами

```

```

int c_3 = 0;
for (int i = 0; i < n; i++)
{

```

```

counter_1 = 0;
for (int j = 0; j < studs[i].marks.size(); j++)
{
    if (studs[i].marks[j] != '5')
    {
        counter_1++;
    }
}
if (counter_1 != 5)
{
    studs_output[counter_2 + c_3].surname = studs[i].surname;
    double sum_marks = 0.0;
    for (int j = 0; j < studs[i].marks.size(); j++)
    {
        sum_marks += (studs[i].marks[j] - '0');
    }
    studs_output[counter_2 + c_3].average_mark = sum_marks / studs[i].marks.size();
    c_3++;
}
}

//ВЫВОД ВЫХОДНОГО МАССИВА
cout << endl << "Фамилии и средние баллы студентов: " << endl;
for (int i = 0; i < n; i++)
{
    cout << studs_output[i].surname << " " << studs_output[i].average_mark << endl;
}

//ВЫВОД ДОПОЛНИТЕЛЬНОГО УСЛОВИЯ
cout << endl << "Студенты с наибольшим количеством отличных оценок: " << endl;
for (int i = 0; i < n; i++)
{
    int counter = 0;
    for (int j = 0; j < studs[i].marks.size(); j++)
    {
        if (studs[i].marks[j] == '5')
        {
            counter++;
        }
    }
    if (counter == max_good)
    {
        cout << studs[i].surname << " " << studs[i].marks << endl;
    }
}
}

```

Файл ConsoleApplicaton3.cpp

```
/*  
Смехнев Иван Юрьевич, 61 группа  
Условие 3: с динамическими массивами и функциями  
С клавиатуры вводится информация об итогах последней экзаменационной сессии. Эта информация включает в себя: 1) целое число n – количество студентов; 2) n объединенных в структуру данных: где , — символьные строки, содержащие не более 20 символов, оценка за экзамен — десятичная цифра из диапазона '2'..'5'. Требуется сформировать массив структур, в котором каждый элемент массива содержит фамилию студента и его средний балл, причем вначале размещаются данные о студентах, не имеющих отличных оценок, затем об остальных студентах, при этом в каждой части сохраняется взаимный порядок следования данных, определенный вводом, и вывести этот массив на экран. Вывести также фамилии и оценки студентов, имеющих наибольшее количество отличных оценок.
```

```
*/
```

```
#include <iostream>  
#include <Windows.h>  
#include <string>
```

```
using namespace std;
```

```
struct student //структура, в которой будет храниться входная информация о студенте (имя, фамилия, оценки)  
{  
    string name;  
    string surname;  
    string marks;  
};
```

```
struct student_output //структура, в которой будет храниться выходная информация о студенте (фамилия) и его средняя оценка  
{  
    string surname;  
    double average_mark = 0.0;  
};
```

```
//Функция подсчета среднего балла
```

```
double AverMark(string& marks)  
{  
    double sum_marks = 0.0;  
    for (int i = 0; i < marks.size(); i++)  
    {  
        sum_marks += (marks[i] - '0');  
    }  
    return sum_marks / marks.size();  
}
```

```
int main()
```



```

{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);

    int n;
    int max_good = 0;

    //ВВОД ДАННЫХ
    cout << "Введите количество студентов: "; cin >> n;
    student* studs = new student[n]; //входной массив структур
    for (int i = 0; i < n; i++)
    {
        cout << "Введите данные " << i + 1 << "-го студента (имя, фамилия, оценки подряд без пробелов): " << endl;
        cin >> studs[i].name;
        cin >> studs[i].surname;
        cin >> studs[i].marks;
        int counter = 0; // счетчик отличных оценок
        for (int j = 0; j < studs[i].marks.size(); j++)
        {
            if (studs[i].marks[j] == '5')
            {
                counter++;
            }
        }
        if (counter > max_good)
        {
            max_good = counter;
        }
    }

    student_output* studs_output = new student_output[n]; //выходной массив структур
    int counter_1; //счетчик оценок без отлично
    int counter_2 = 0; //счетчик студентов, не имеющих отличных студентов

    //заполнение и вывод выходного массива студентами, не имеющих отличных оценок
    cout << endl << "Фамилии и средние баллы студентов: " << endl;
    for (int i = 0; i < n; i++)
    {
        counter_1 = 0;
        for (int j = 0; j < studs[i].marks.size(); j++)
        {
            if (studs[i].marks[j] != '5')
            {
                counter_1++;
            }
            else break;
        }
        if (counter_1 == 5)
        {
            studs_output[counter_2].surname = studs[i].surname;
            studs_output[counter_2].average_mark = AverMark(studs[i].marks);
        }
    }
}

```

```

        cout << studs_output[counter_2].surname << " " <<
studs_output[counter_2].average_mark << endl;
    }
}
//заполнение и вывод выходного массива остальными студентами
int counter_3 = 0;
for (int i = 0; i < n; i++)
{
    counter_1 = 0;
    for (int j = 0; j < studs[i].marks.size(); j++)
    {
        if (studs[i].marks[j] != '5')
        {
            counter_1++;
        }
    }
    if (counter_1 != 5)
    {
        studs_output[counter_2 + counter_3].surname = studs[i].surname;
        studs_output[counter_2 + counter_3].average_mark = AverMark(studs[i].marks);
        cout << studs_output[counter_2 + counter_3].surname << " " << studs_output[counter_2
+ counter_3].average_mark << endl;
    }
}

//вывод дополнительного условия
cout << endl << "Студенты с наибольшим количеством отличных оценок: " << endl;
for (int i = 0; i < n; i++)
{
    int counter = 0;
    for (int j = 0; j < studs[i].marks.size(); j++)
    {
        if (studs[i].marks[j] == '5')
        {
            counter++;
        }
    }
    if (counter == max_good)
    {
        cout << studs[i].surname << " " << studs[i].marks << endl;
    }
}
}

```