**Основы разработки приложений с использованием STL**

**Задача 1**

Выведите фамилии и имена учащихся в порядке убывания их среднего балла.

**Входные данные**

Заданы сначала количество учащихся n, затем n строк, каждая из которых содержит фамилию, имя и три числа (оценки по трем предметам: математике, физике, информатике). Данные в строке разделены одним пробелом. Оценки принимают значение от 1 до 5.

**Выходные данные**

Необходимо вывести пары фамилия-имя по одной на строке, разделяя фамилию и имя одним пробелом. Выводить оценки не нужно. Если несколько учащихся имеют одинаковые средние баллы, то их нужно выводить в порядке, заданном во входных данных.

**Sample Input:**

3

Markov Valeriy 5 5 5

Sergey Petrov 1 1 1

Petrov Petr 3 3 3

**Sample Output:**

Markov Valeriy

Petrov Petr

Sergey Petrov

Завдання

// Massive DZ SHAG.cpp : This file contains the 'main' function. Program execution begins and ends there.

//

#define CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include<Windows.h>

#include<vector>

#include<string>

#include<algorithm>

#include<functional>

#include<numeric>

using namespace std;

class Student

{

private:

string lastname;

string name;

vector<int>marks;

float GPA;

public:

Student()

{

}

Student(string lastname, string name, vector<int>marks)

{

this->lastname = lastname;

this->name = name;

this->marks = marks;

GPA = accumulate(marks.begin(), marks.end(), 0) / (float)marks.size();

}

void Add\_Marks(int mark)

{

marks.push\_back(mark);

}

void Show()

{

cout << lastname << " " << name << endl;

}

bool operator<(const Student& obj) const

{

return this->GPA < obj.GPA;

}

bool operator>(const Student& obj) const

{

return this->GPA > obj.GPA;

}

bool operator<=(const Student& obj) const

{

return this->GPA <= obj.GPA;

}

bool operator>=(const Student& obj) const

{

return this->GPA >= obj.GPA;

}

bool operator==(const Student& obj) const

{

return this->GPA == obj.GPA;

}

};

class StudyJournal

{

private:

vector<Student>students;

public:

void Filling\_Journal(string name, string lastname, vector<int>marks)

{

students.push\_back(Student(name, lastname, marks));

}

void Show\_Rating()

{

sort(students.begin(), students.end(), greater<Student>());

for (int i = 0; i < students.size(); i++)

{

students[i].Show();

}

}

};

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

vector<int>marks;

StudyJournal journal;

string name, lastname;

int number\_of\_student,count\_marks,mark;

cout << "Введіть скільки студентів додати ->";

cin >> number\_of\_student;

cout << "------------------------------------\n";

cout << "Введіть скільки оцінок у студентів ->";

cin >> count\_marks;

cout << "------------------------------------\n";

for (int i = 0; i < number\_of\_student; i++)

{

cout << "Введіть ім'я " << i + 1 << " студента ->";

cin >> name;

cout << "Введіть прізвище " << i + 1 << " студента ->";

cin >> lastname;

for (int j = 0; j < count\_marks; j++)

{

cout << "Введіть " << j + 1 << " оцінку ->";

cin >> mark;

marks.push\_back(mark);

}

journal.Filling\_Journal(name, lastname, marks);

marks.clear();

cout << "------------------------------------\n";

}

journal.Show\_Rating();

system("pause");

return 0;

}