#include <iostream>

#include <fstream>

#include <vector>

#include <algorithm>

using namespace std;

// Constants

const int NUM\_BIO\_STREAM\_CLASSES = 3;

const int NUM\_MATH\_STREAM\_CLASSES = 5;

const int STUDENTS\_PER\_CLASS = 40;

// Student structure

struct Student {

string name;

double biology;

double physics;

double chemistry;

double totalMarks;

int stream; // 0 for Bio, 1 for Math

int classNumber;

int schoolRank;

int classRank;

};

// Function prototypes

void readStudentData(vector<Student>& students);

void calculateTotalMarks(vector<Student>& students);

void rankStudents(vector<Student>& students);

void printResults(const vector<Student>& students);

void writeToFile(const vector<Student>& students);

int main() {

vector<Student> students;

// Read student data from the user

readStudentData(students);

// Calculate total marks and rank students

calculateTotalMarks(students);

rankStudents(students);

// Print results to the console and write to a file

printResults(students);

writeToFile(students);

return 0;

}

void readStudentData(vector<Student>& students) {

// Read data for Bio stream students

for (int stream = 0; stream < 2; ++stream) {

int numClasses = (stream == 0) ? NUM\_BIO\_STREAM\_CLASSES : NUM\_MATH\_STREAM\_CLASSES;

for (int classNum = 1; classNum <= numClasses; ++classNum) {

for (int studentNum = 1; studentNum <= STUDENTS\_PER\_CLASS; ++studentNum) {

Student student;

student.stream = stream;

student.classNumber = classNum;

cout << "Enter details for Student " << studentNum << " in ";

cout << ((stream == 0) ? "Bio" : "Math") << " stream, Class " << classNum << ":\n";

cout << "Name: ";

cin >> student.name;

cout << "Biology marks: ";

cin >> student.biology;

cout << "Physics marks: ";

cin >> student.physics;

cout << "Chemistry marks: ";

cin >> student.chemistry;

students.push\_back(student);

}

}

}

}

void calculateTotalMarks(vector<Student>& students) {

for (auto& student : students) {

if (student.stream == 0) {

// Bio stream

student.totalMarks = student.biology + student.physics + student.chemistry;

} else {

// Math stream

student.totalMarks = student.biology + student.physics + student.chemistry;

}

}

}

void rankStudents(vector<Student>& students) {

// Sort students based on total marks

sort(students.begin(), students.end(), [](const Student& a, const Student& b) {

return a.totalMarks > b.totalMarks;

});

// Assign school ranks

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

students[i].schoolRank = i + 1;

}

// Assign class ranks

for (int classNum = 1; classNum <= NUM\_BIO\_STREAM\_CLASSES + NUM\_MATH\_STREAM\_CLASSES; ++classNum) {

vector<Student> classStudents;

copy\_if(students.begin(), students.end(), back\_inserter(classStudents),

[classNum](const Student& s) { return s.classNumber == classNum; });

sort(classStudents.begin(), classStudents.end(), [](const Student& a, const Student& b) {

return a.totalMarks > b.totalMarks;

});

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

auto it = find(students.begin(), students.end(), classStudents[i]);

it->classRank = i + 1;

}

}

}

void printResults(const vector<Student>& students) {

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

cout << "Name\t\tStream\tClass\tTotal Marks\tSchool Rank\tClass Rank\n";

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

for (const auto& student : students) {

cout << student.name << "\t" << ((student.stream == 0) ? "Bio" : "Math") << "\t";

cout << student.classNumber << "\t" << student.totalMarks << "\t\t";

cout << student.schoolRank << "\t\t" << student.classRank << "\n";

}

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

}

void writeToFile(const vector<Student>& students) {

ofstream outputFile("rankings.txt");

if (outputFile.is\_open()) {

outputFile << "---------------------------------------------------------------------\n";

outputFile << "Name\t\tStream\tClass\tTotal Marks\tSchool Rank\tClass Rank\n";

outputFile << "---------------------------------------------------------------------\n";

for (const auto& student : students) {

outputFile << student.name << "\t" << ((student.stream == 0) ? "Bio" : "Math") << "\t";

outputFile << student.classNumber << "\t" << student.totalMarks << "\t\t";

outputFile << student.schoolRank << "\t\t" << student.classRank << "\n";

}

outputFile << "---------------------------------------------------------------------\n";

outputFile.close();

} else {

cout << "Error opening the file for writing.\n";

}

}