#include <iostream>

#include <string>

#include <fstream>

using namespace std;

int getspaces(string str)

{

    int space = 0;

    for (int i = 0; i < str.length(); i++)

    {

        if (str.at(i) == ' ')

        {

            space++;

        }

    }

    return space;

}

string getGrade(double num)

{

    if (num >= 90 && num <= 100)

    {

        return "A";

    }

    if (num >= 80 && num <= 89)

    {

        return "B";

    }

    if (num >= 70 && num <= 79)

    {

        return "C";

    }

    if (num >= 60 && num <= 69)

    {

        return "D";

    }

    if (num >= 0 && num <= 59)

    {

        return "F";

    }

    return "F";

}

int main()

{

    const string ASSIGNMENT\_NAMES[] = {"POLICY QUIZ", "QUIZZES", "HOMEWORK", "LABS", "PROJECT", "TEAMWORK", "DISCUSSION TOPIC", "TESTS"};

    int assignmentSize[8] = {3, 3, 3, 3, 3, 3, 3, 3};

    float maxScore[8] = {100, 100, 100, 100, 100, 100, 100, 100}; // to store the max scores of assignments, for example, maxScore[3] = 30 if max score of each lab is 30 points

    float studentScores[8];                                       // to store the scores of students, for example studentScores[7] stores total scores of 3 student tests. studentScores[3] stores total scores of student labs

    string listScores[8];

    double totalStudentScore;

    double totalMaxScore;

    double numericgrade;

    while (1)

    {

        int choice;

        cout << "FA2021\_CSGradingStudentsApplication\_Smith.cpp" << endl;

        cout << "TASK OF GRADING - JAMES SMITH" << endl;

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

        cout << "1.Grading One Student " << endl;

        cout << "2.Printing The Grade of One Student " << endl;

        cout << "3.Printing The Grades of Class " << endl;

        cout << "0.Exit " << endl;

        cout << "Enter choice: ";

        cin >> choice;

        // Read student id, last name, first name of one student

        switch (choice)

        {

        case 1:

        {

            string coursename;

            cout << "Enter course name: ";

            getline(cin, coursename);

            cout << "Enter student id: ";

            string stid;

            getline(cin, stid);

            cout << "Enter last name : ";

            string lastname;

            getline(cin, lastname);

            cout << "Enter first name : ";

            string firstname;

            getline(cin, firstname);

            for (int i = 0; i < 8; i++)

            {

                string scores;

                cout << "Enter score for " << ASSIGNMENT\_NAMES[i] << ": ";

                getline(cin, scores);

                int num = 0;

                listScores[i] = scores;

                maxScore[i] = 0;

                int NUMBERS = getspaces(scores);

                for (int j = 0; j < NUMBERS + 1; j++)

                {

                    int index = scores.find\_first\_of(" ");

                    string s = scores.substr(0, index);

                    num += stod(s);

                    if (stod(s) > maxScore[i])

                    {

                        maxScore[i] = stod(s);

                    }

                    scores.erase(0, index + 1);

                }

                studentScores[i] = num;

            }

            totalStudentScore = 0;

            for (int i = 0; i < 8; i++)

            {

                totalStudentScore += studentScores[i];

            }

            totalMaxScore = 0;

            for (int i = 0; i < 8; i++)

            {

                totalMaxScore += maxScore[i] \* assignmentSize[i];

            }

            double numericgrade = 100 \* totalStudentScore / totalMaxScore;

            cout << "FA2021\_CSGradingStudentsApplication\_Smith.cpp" << endl;

            cout << "TASK OF GRADING - JAMES SMITH" << endl;

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

            for (int i = 0; i < 8; i++)

            {

                cout << ASSIGNMENT\_NAMES[i] << "    \t\t" << listScores[i] << endl;

            }

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

            cout << "Total Scores : \t" << totalStudentScore << "/" << totalMaxScore << endl;

            cout << "Numeric grade : \t" << numericgrade << endl;

            cout << "Letter grade: \t" << getGrade(numericgrade) << endl;

            cout << "Do you want to store in student data file ? Y/N";

            char stores;

            cin >> stores;

            if (stores == 'Y' || stores == 'y')

            {

                ofstream MyFile("studentGrade.txt");

                // Write to the file

                MyFile << stid << "," << firstname << "," << numericgrade << "," << getGrade(numericgrade);

                for (int i = 0; i < 8; i++)

                {

                    MyFile << listScores[i] << ",";

                }

                // Close the file

                MyFile.close();

            }

        }

        break;

        case 2:

            cout << "FA2021\_CSGradingStudentsApplication\_Smith.cpp" << endl;

            cout << "TASK OF GRADING - JAMES SMITH" << endl;

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

            for (int i = 0; i < 8; i++)

            {

                cout << ASSIGNMENT\_NAMES[i] << "\t\t" << listScores[i] << endl;

            }

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

            cout << "Total Scores : \t" << totalStudentScore << "/" << totalMaxScore << endl;

            cout << "Numeric grade : \t" << numericgrade << endl;

            cout << "Letter grade: \t" << getGrade(numericgrade) << endl;

            break;

        case 3:

            /\* code \*/

            break;

        case 0:

            /\* code \*/

            break;

        default:

            break;

        }

    }

    return 0;

}



