LAB REPORT

Lab 1
Introduction

CSE 4308
DATABASE MANAGEMENT SYSTEMS LAB

NAME: CHOWDHURY ASHFAQ
STUDENT ID: 200042123
PROGRAM: SWE
GROUP: 1A
DATE: 14/08/22

Tasks:

Winter 2021-22 Introduction Lab 1

Before the advent of databases, information was recorded using the traditional file system. In this lab, we want to experience how life was back then.
Consider that your database consists of two files, namely 'studentInfo.txt' and 'grades.txt'. Each row of the studentInfo.txt file contains *Student ID*, *Name*, *Age*, *Blood Group*, and *Department* of a student. And each row of the grades.txt file contains the *Student ID*, *GPA*, and the *Semester* in which that GPA was achieved.
Assume that the values of *Student ID* and *Age* fit within the range of a 32-bit integer. The values under *Name*, *Blood Group*, and *Department* have at most 10 characters. The *GPA*s are given as floating point values within [2.50, 4.00].
It is guaranteed that the *Student ID*s are unique for each student. The values for each *field* is separated by semicolon (;).
Now, write separate programs (using any programming language that you prefer) to accomplish the following tasks:

1. Print the *Student ID* having the highest GPA among all the students.
2. Take *Student ID*, *GPA*, and *Semester* as input. Then after validating the input, insert the information as a new row in the grades.txt file. If the information is invalid, discard the input and show an error message.
3. Take *Student ID* as input and show his/her name and CGPA (average GPA for all the semesters he/she attended). Print an error message if the *Student ID* does not exist in your database.

The first DBMS Lab was basically an introduction to our course which was given so that we could realize the need and importance of database management system. So, we had to complete this lab with C++ or Java or C# but not SQL or other Database Languages.

The Lab was divided into 3 parts which can be said as Task 1, Task 2, and Task 3. Each of these tasks have been analyzed below with the explanation of the solutions as well.

I have solved all of the problems in C++ language.

Task 01:

Before the advent of databases, information was recorded using the traditional file system. In this lab, we want to experience how life was back then.
Consider that your database consists of two files, namely 'studentInfo.txt' and 'grades.txt'. Each row of the studentInfo.txt file contains *Student ID*, *Name*, *Age*, *Blood Group*, and *Department* of a student. And each row of the grades.txt file contains the *Student ID*, *GPA*, and the *Semester* in which that GPA was achieved.
Assume that the values of *Student ID* and *Age* fit within the range of a 32-bit integer. The values under *Name*, *Blood Group*, and *Department* have at most 10 characters. The *GPA*s are given as floating point values within [2.50, 4.00].
It is guaranteed that the *Student ID*s are unique for each student. The values for each *field* is separated by semicolon (;).
Now, write separate programs (using any programming language that you prefer) to accomplish the following tasks:

1. Print the *Student ID* having the highest GPA among all the students.

Analysis of the problem:

The problem in Task 01 says us to print the student ID who have the highest GPA among all the students. The GPA of the students are stored in a .txt file named as grades.txt. It has 3 columns containing Student ID,grades and semester respectively separated by a ';'.

We'd need to read the values from the text file and then find the highest GPA among all the records. The corresponding Student ID to the GPA needs to be printed.

Solution to Task 01:

```cpp
#include <iostream>
#include<fstream>
#include<sstream>
#include<string>
using namespace std;

int main()
{
    ifstream file1("grades.txt");

    double highestGrade = 0;
    int studentID;

    if (file1.is_open()) {

        string line;
        while (getline(file1, line)) {

            stringstream ss(line);

            int ID;
            string ID1,grade1,semester;
            double grade;

            getline(ss, ID1,';');
            ID = stoi(ID1);

            getline(ss, grade1,';');
            grade = stof(grade1);

            getline(ss, semester, ',');

            if (grade > highestGrade) {
                highestGrade = grade;
                studentID = ID;
            }
        }
        cout << "Highest grade is: " << highestGrade<<" of ID: "<< studentID<< endl;
    }
}
```

Output:

```
Microsoft Visual Studio Debug Console
Highest grade is: 3.99 of ID: 190658

C:\Users\ASUS\Desktop\DBMS LAB\Lab1\C++ custom1\x64\Debug\C++ custom1.exe (process 29476) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

Explanation:

- As we need to read data from a file so we need to use the '**ifstream**'(Input File stream) keyword. It is a stream of data used for reading input from any file. With ifstream we open the grades.txt file which is placed in the same directory as our program. We take two variables '**highestGrade**' of type double,initializing it with 0, and '**studentID**' of type int which would store the highestGrade in the grades.txt file and the corresponding student ID respectively.

```
ifstream file1("grades.txt");

double highestGrade = 0;
int studentID;
```

- Then we check if the input stream file is open, if it's open then we read the values from the file line by line until it reaches the last line with '**getline()**' function which can take multi-line input as well. The string is stored in the '**line**' variable.

```
if (file1.is_open()) {

    string line;
    while (getline(file1, line)) {
```

- Next, we use '**stringstream**' from the 'sstream' header which allows us to manipulate data treating the string as a stream. With the help of 'getline()', we separate each line of the stream in 3 different parts with the separator '**;**'. The 3 parts are initially stored in ID1,grade1, and semester. The stringstream can only return a string. So, we store the string in a string variable then convert the string to double or int with the '**stof()**' and '**stoi()**' functions respectively.

```
stringstream ss(line);

int ID;
string ID1,grade1,semester;
double grade;

getline(ss, ID1,';');
ID = stoi(ID1);

getline(ss, grade1,';');
grade = stof(grade1);

getline(ss, semester, ';');
```

- After that, for grades in each of the line we check if it's greater than the highestGrade. If the grade is greater than highestGrade then we store the grade in highestGrade and also store the student ID of that person in studentID variable.

```
if (grade > highestGrade) {
    highestGrade = grade;
    studentID = ID;
}
```

- In this way we get the highest Grade and the student ID of that student and later print it.

```
cout << "Highest grade is: " << highestGrade<<" of ID: "<< studentID<< endl;
```

Task 02

> 2. Take *Student ID*, *GPA*, and *Semester* as input. Then after validating the input, insert the information as a new row in the `grades.txt` file. If the information is invalid, discard the input and show an error message.

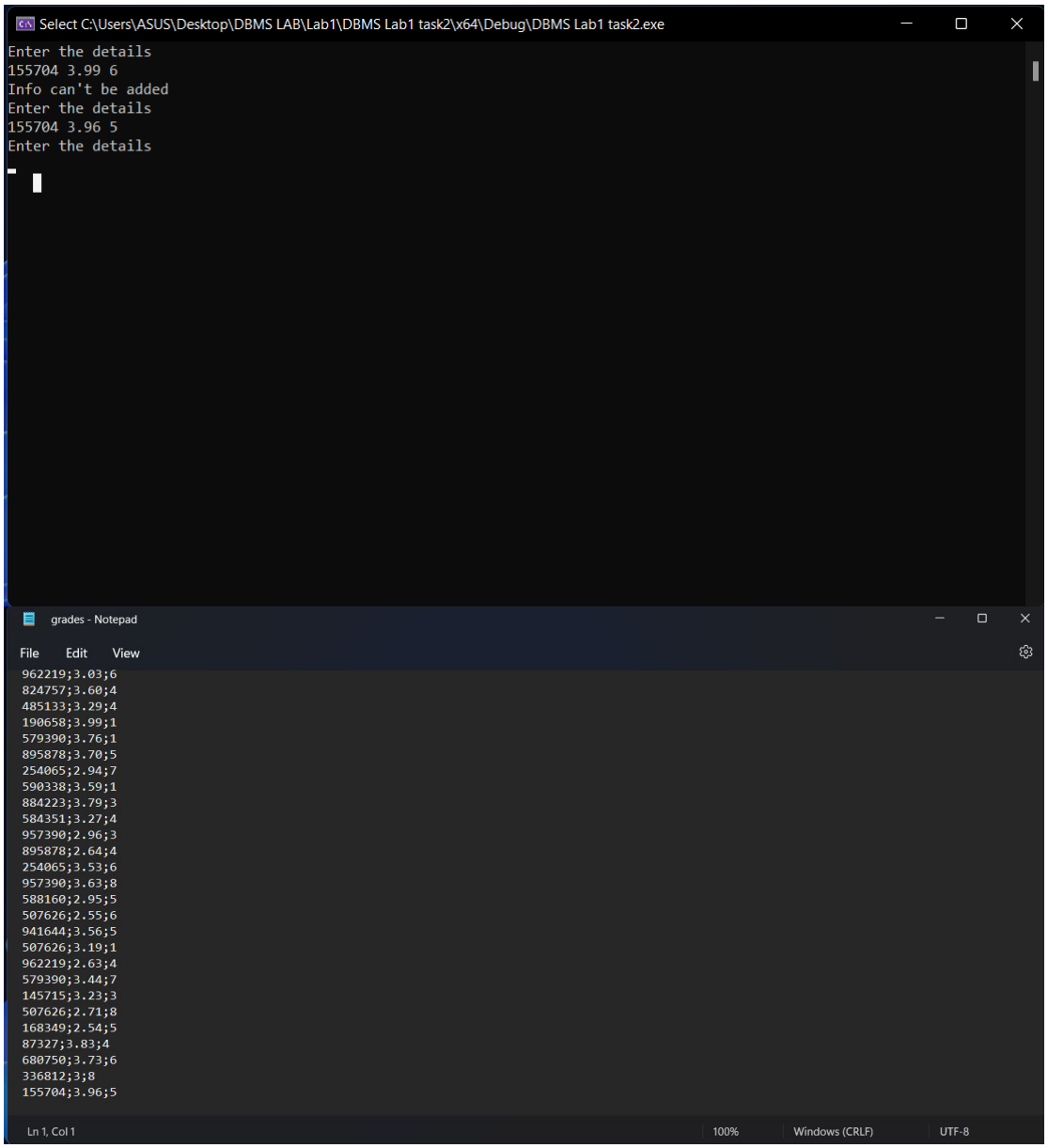
Analysis:

The problem in Task 02 tells us to take Student ID, GPA, and Semester as input from the user. Then we need to check if the information is valid because a student can't have 2 GPA's for one semester. So, if the information is valid then we'd have to insert the information as a new row in the grades.txt file.

Solution:

```cpp
#include <iostream>
#include<vector>
#include<map>
#include<fstream>
#include<sstream>
using namespace std;

int main()
{
    ifstream file2("grades.txt");

    while (1) {

        ofstream file1;
        file1.open("grades.txt", ios_base::app);

        cout << "Enter the details" << endl;

        int ID, f=0;
        string semester;
        double GPA;

        cin >> ID >> GPA >> semester;

        if (file2.is_open()) {

            string line;
            while (getline(file2, line)) {
                stringstream ss(line);

                int ID2;
                string ID1, grade1, semester1;
                double grade;

                getline(ss, ID1, ';');
                ID2 = stoi(ID1);

                getline(ss, grade1, ';');
                getline(ss, semester1, ';');

                if (ID==ID2 && semester==semester1) {
                    f = 1;
                    break;
                }
            }
            if (f == 1) {
                cout << "Info can't be added" << endl;
            }
            else {
                file1 << ID << ";" << GPA << ";" << semester<<"\n";
            }
        }
    }
    return 0;
}
```

Output:



Explanation:

- First we take the "**grades.txt**" file as input file stream**(ifstream)** so that we can check the validity of the data entered by the user. Next in a while loop we open the "**grades.txt**" file as output file stream(**ofstream**) as we need to append rows on the file entered by the user. Afterwards, we take the inputs of the user which are stored in ID(int), GPA(double), and semester(string) variables respectively.

```
ifstream file2("grades.txt");

while (1) {

    ofstream file1;
    file1.open("grades.txt", ios_base::app);

    cout << "Enter the details" << endl;

    int ID, f=0;
    string semester;
    double GPA;

    cin >> ID >> GPA >> semester;
```

- In the next step, we compare if the student ID and semester entered matches with any of the student ID stored in the file and its corresponding semester. If, it matches then we change the value of the flag f to 1 else it remains 0. The student ID, grade, and semester is separated from the file using the "**stringstream**".

```
while (getline(file2, line)) {
    stringstream ss(line);

    int ID2;
    string ID1, grade1, semester1;
    double grade;

    getline(ss, ID1, ';');
    ID2 = stoi(ID1);

    getline(ss, grade1, ';');
    getline(ss, semester1, ';');

    if (ID==ID2 && semester==semester1) {
        f = 1;
        break;
    }
}
```

- At last we check if value of flag **f=1** then the input data is invalid and we discard it. If the data is valid that is **f=0** then we insert the information to grades.txt file with the help of "**ofstream**".

```
if (f == 1) {
    cout << "Info can't be added" << endl;
}
else {
    file1 << ID << ";" << GPA << ";" << semester<<"\n";
}
```

Task 03

> 3. Take *Student ID* as input and show his/her name and CGPA (average GPA for all the semesters he/she attended). Print an error message if the *Student ID* does not exist in your database.

Analysis:

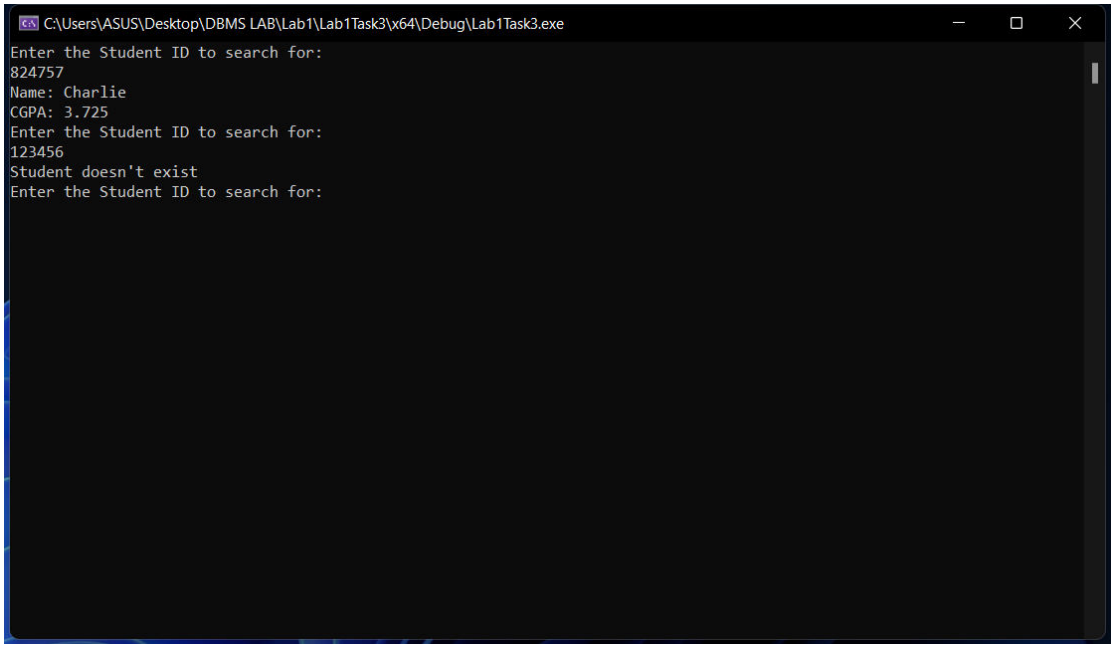
The problem says us to take Student ID as input from the user then we need to show the name of the student with that ID. The names and other information are stored in the "studentInfo.txt" file. With the name we'll also have to show the CGPA or the average GPA for the semesters that student attended of the student. We'd get the grades of the student from the "grades.txt" file.
If there isn't any student of that student ID then we need to print an error message.

Solution:

```
int main()
{
    ifstream file1("grades.txt");
    ifstream file2("studentInfo.txt");
    while (1) {
        cout << "Enter the Student ID to search for: " << endl;

        int ID, f = 0;
        double sum = 0, occ = 0, avg;
        string name;
        cin >> ID;

        if (file2.is_open()) {

            string line;
            while (getline(file2, line)) {

                stringstream ss(line);

                int ID2;
                string ID1, name1, age, blood_group, dept;
                double grade;
                double* mem = &grade;

                getline(ss, ID1, ';');
                ID2 = stoi(ID1);

                getline(ss, name1, ';');
                getline(ss, age, ';');
                getline(ss, blood_group, ';');
                getline(ss, dept, ';');

                if (ID == ID2) {
                    name = name1;
                    f = 1;
                    break;
                }
            }
```

```
            while (getline(file1, line)) {
                istringstream ss(line);

                double grade;
                string ID3, grade1, semester;
                int ID4;

                getline(ss, ID3, ';');
                ID4 = stoi(ID3);

                getline(ss, grade1, ';');
                grade = stof(grade1);

                getline(ss, semester, ';');
                if (ID4 == ID) {
                    sum += grade;
                    occ++;
                }
            }
            avg = sum / occ;
        }
        if (f == 1) {
            cout << "Name: " << name << endl << "CGPA: " << avg << endl;
        }
        else {
            cout << "Student doesn't exist" << endl;
        }
    }
    return 0;
}
```

Output:



Explanation:

- First we need to take the input from the user which is a Student ID. We need to check whether the student exists so we take "**studentInfo.txt**" file as input file stream or "**ifstream**".

```
ifstream file1("grades.txt");
ifstream file2("studentInfo.txt");
```

- We take variables ID(int),f(int),sum(double),avg(double),name(string) and occ(int) to store the needed values.

```
int ID, f = 0;
double sum = 0, occ = 0, avg;
string name;
```

- The user given student ID is stored in ID variable.

```
cin >> ID;
```

- Next we separate the fields which are ';' separated of the "studentInfo.txt" file using "**stringstream**". Each field is stored in ID2,name1,age,blood_group, and dept variable respectively.

```
string line;
while (getline(file2, line)) {

    stringstream ss(line);

    int ID2;
    string ID1, name1, age, blood_group, dept;
    double grade;
    double* mem = &grade;

    getline(ss, ID1, ';');
    ID2 = stoi(ID1);

    getline(ss, name1, ';');
    getline(ss, age, ';');
    getline(ss, blood_group, ';');
    getline(ss, dept, ';');
```

- Then we check through the file if the student ID inputted exists or not. If it exists then we store the name of that student in the name variable and change the value of flag f to 1.

```
if (ID == ID2) {
    name = name1;
    f = 1;
    break;
}
```

- After that we open the "**grades.txt**" file with "**ifstream**" and separate all the fields with the help of "**istringstream**". We store the ID converted to int(with **stoi()**) in ID4 and the grade converted to double(with **stof()**) in grade variable.

```
while (getline(file1, line)) {
    istringstream ss(line);

    double grade;
    string ID3, grade1, semester;
    int ID4;

    getline(ss, ID3, ';');
    ID4 = stoi(ID3);

    getline(ss, grade1, ';');
    grade = stof(grade1);

    getline(ss, semester, ';');
```

- Then we check for each line if the ID matches with the inputted ID. If match is found then we add the grade to sum variable and increase occ by +1 times.

```
if (ID4 == ID) {
    sum += grade;
    occ++;
}
```

- After the checking of the whole file is finished we find the average which is stored in **avg** variable by **dividing sum by occ**.

```
avg = sum / occ;
```

- Lastly if f=1 then we'd print the name of the student stored in '**name**' variable and CGPA stored in '**avg**'. Else if f=0, print "Student doesn't exist".

Interesting Findings:

- Using the stod() function in 'Visual Studio Community 2022' gives an exception whereas the stof() function works for the same case.

- We can't delete the txt file if the program is running because the file is open in the program.

Problems faced and solution:

- Couldn't find any keyword which can return an integer or double from the stream. We can get only string from a stream. So, we need to use some other function to convert from string to integer or double.

- The stod() function shows exception but stof() function works for the same case. I couldn't find an alternative to stod() so I couldn't complete my Lab in due time, everything else was working. I fixed the problem by simply using the stof() function instead of stod().

N.B: The file "grades.txt", "studentInfo.txt", and "program.cpp" were in the same directory.