


CSE 4308

DATABASE MANAGEMENT SYSTEMS LAB

NAME: Nazia Karim Khan Oishee
Student ID: 200042137



Introduction	2
Explanation of the solution of task 1	2
Final working code of task 1	4
Explanation of the solution of task 2	7
Final working Code of task 2	8
Explanation of the solution of task 3	13
Final working code of task 3	14
Problems which I faced	18

Introduction

Before the inception of databases, information was recorded using the traditional file system. In our very first lab, got a glimpse of how life was back then.

For the lab we were given two text files and 3 tasks. The files were '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.

Explanation of the solution of task 1

Our first task was to print the Student ID having the highest GPA among all the students.

So for task 1, I first created a struct named Grade to store the information of a student given in the text file.

```
typedef struct Grade
{
    long long int id;
    float gpa;
    int semester;
} gradefile;
```

After that I created a list of that struct as there were multiple students.

```
gradefile List1[10000];
```

In the main function I used a file pointer to read the file. The file pointer would return a NULL value if the file was null and would terminate the program.

```
FILE* fp;
fp=fopen("E:\\Download\\CSE 4308\\grades.txt","r");
if(fp==NULL)
{
    printf("Error in opening input file");
    return 0;
}
```

```
}
```

Otherwise it would enter into the while loop and through the fscanf function read the whole file until the end of file.

```
else
{
    char temp[100000];
    while(fscanf(fp,"%s",temp)!=EOF)
    {
        long long int student_id;
        float student_gpa;
        int student_semester;
        char ID[10000]= {'\0'};
        char GPA[10000]= {'\0'};
        char SEMESTER[10000]= {'\0'};
        int i=0,j=0;
```

In the loop I took necessary variables and arrays to implement the task.I used the string array “temp” to read the file and store its data in three different arrays.

```
while(temp[j]!=';')
{
    ID[i]=temp[j];
    i++;
    j++;
}
i=0;
j++;
while(temp[j]!=';')
{
    GPA[i]=temp[j];
    i++;
    j++;
}
i=0;
j++;
while(temp[j]!='\0')
{
    SEMESTER[i]=temp[j];
```

```

        i++;
        j++;
    }

```

Then I used the `sscanf` function to convert these string data into integer and float data types and pushed them in the list.

```

    sscanf(ID, "%lld", &student_id);
    sscanf(GPA, "%f", &student_gpa);
    sscanf(SEMESTER, "%d", &student_semester);
    List1[x].id=student_id;
    List1[x].gpa=student_gpa;
    List1[x].semester=student_semester;
    x++;

```

The loop will go on till the file pointer reaches the end of the file. After that the loop will terminate. And the file pointer will be closed. After that I used a for loop to iterate through the list and find the maximum GPA holder's index in the list and finally print the student ID.

```

for(int i=0; i<x ; i++)
{
    if(List1[i].gpa>MaxGpa)
    {
        MaxGpa=List1[i].gpa;
        k=i;
    }
}

printf(" Student ID  of who has maximum GPA is : %lld",List1[k].id);

```

Final working code of task 1

```

#include<stdio.h>
#include<stdlib.h>

typedef struct Grade
{
    long long int id;

```

```

float gpa;
int semester;

} gradefile;

int x=0;
gradefile List1[10000];

int main()
{

FILE* fp;
fp=fopen("E:\\Download\\CSE 4308\\grades.txt","r");
if(fp==NULL)
{
    printf("Error in opening input file");
    return 0;
}
else
{
    char temp[100000];
    while(fscanf(fp,"%s",temp)!=EOF)
    {
        long long int student_id;
        float student_gpa;
        int student_semester;
        char ID[10000]= {'\0'};
        char GPA[10000]= {'\0'};
        char SEMESTER[10000]= {'\0'};
        int i=0,j=0;

        while(temp[j]!=';')
        {
            ID[i]=temp[j];
            i++;
            j++;
        }
        i=0;
        j++;
        while(temp[j]!=';')
        {

```

```

        GPA[i]=temp[j];
        i++;
        j++;
    }
    i=0;
    j++;
    while(temp[j]!='\0')
    {
        SEMESTER[i]=temp[j];
        i++;
        j++;
    }
    sscanf(ID,"%lld",&student_id);
    sscanf(GPA,"%f",&student_gpa);
    sscanf(SEMESTER,"%d",&student_semester);
    List1[x].id=student_id;
    List1[x].gpa=student_gpa;
    List1[x].semester=student_semester;
    x++;
}

}
fclose(fp);
float MaxGpa = 0;
int k;
for(int i=0; i<x ; i++)
{
    if(List1[i].gpa>MaxGpa)
    {
        MaxGpa=List1[i].gpa;
        k=i;
    }
}
printf(" Student ID  of who has maximum GPA is : %lld",List1[k].id);
return 0;
}

```

Explanation of the solution of task 2

Our second task was to take Student ID, GPA, and Semester as input. Then after validating the input we had to insert the information as a new row in the grades.txt file. If the information was invalid, we had to discard the input and show an error message.

To complete this task I took another struct along with the previous one to store the studentinfo files's information .

```
typedef struct StudentInfo
{
    char name[10000];
    long long studentID;
    long age;
    char department[10000];
    char bloodGroup[10000];
} Info;
```

And then I read the whole second file also and store its data into another list. Then I took one more file pointer to add the input information in the grades file.

```
///append
FILE* f;
f=fopen("E:\\Download\\CSE 4308\\grades.txt","a");
```

After that I used a for loop to iterate through the loop and see if the student id exists in the students' list. If that exists, it means that the id is valid. Then I checked the constraints which were given upon the semester and GPA range to completely verify and validate the input. If the input is valid, then it will append at the end of the file.

```
for(int i=0; i<g ; i++)
{
    if(STUDENT_ID==List2[i].studentID)
    {
        check=true;
    }
}
if(check)
{
    if(STUDENT_GPA<2.50 || STUDENT_GPA>4.00)
    {
        printf("GPA is out of range");
    }
}
```



```

        return 0;
    }
    else if(STUDENT_SEMESTER<1 || STUDENT_SEMESTER>8)
    {
        printf("Semester out of range");
        return 0;
    }
    for(int i=0; i<x ; i++)
    {
        if(STUDENT_ID==List1[i].id &&
List1[i].semester==STUDENT_SEMESTER)
        {
            printf("GPA for semester already exists");
            return 0;
        }
    }
    fprintf(f,"%lld;%.2f;%d\n",STUDENT_ID,STUDENT_GPA,STUDENT_SEMESTER);
}

```

Final working Code of task 2

```

#include<stdio.h>
#include<stdlib.h>
#include<stdbool.h>

typedef struct StudentInfo
{
    char name[10000];
    long long studentID;
    long age;
    char department[10000];
    char bloodGroup[10000];
} Info;

typedef struct Grade
{
    long long int id;
    float gpa;
    int semester;
}

```

```

} gradefile;

int x=0,g=0;
gradefile List1[10000];
Info List2[10000];

int main()
{
    ///1st file read
    FILE* fp;
    fp=fopen("E:\\Download\\CSE 4308\\grades.txt","r");
    if(fp==NULL)
    {
        printf("Error in opening file");
        return 0;
    }
    else
    {
        char temp[100000];
        while(fscanf(fp,"%s",temp)!=EOF)
        {
            long long int student_id;
            float student_gpa;
            int student_semester;
            char ID[10000]= {'\0'};
            char GPA[10000]= {'\0'};
            char SEMESTER[10000]= {'\0'};
            int i=0,j=0;

            while(temp[j]!=';')
            {
                ID[i]=temp[j];
                i++;
                j++;
            }
            i=0;
            j++;
            while(temp[j]!=';')
            {
                GPA[i]=temp[j];
                i++;
            }
        }
    }
}

```

```

        j++;
    }
    i=0;
    j++;
    while(temp[j]!='\0')
    {
        SEMESTER[i]=temp[j];
        i++;
        j++;
    }
    sscanf(ID,"%11d",&student_id);
    sscanf(GPA,"%f",&student_gpa);
    sscanf(SEMESTER,"%d",&student_semester);
    List1[x].id=student_id;
    List1[x].gpa=student_gpa;
    List1[x].semester=student_semester;
    x++;
}

}
fclose(fp);
///2nd file read
FILE* p;
p=fopen("E:\\Download\\CSE 4308\\studentInfo.txt","r");
if(p==NULL)
{
    printf("Error in opening 2nd file");
    return 0;
}
else
{
    char temp2[10000];
    while(fscanf(fp,"%s",temp2)!=EOF)
    {
        char S_ID[100]= {'\0'};
        char S_name[10]= {'\0'};
        char S_age[15]= {'\0'};
        char S_bloodGroup[10]= {'\0'};
        char S_department[10]= {'\0'};
        long StudentID;
        long Age;
    }
}

```

```

int j=0;
int i=0;
while(temp2[j]!=';')
{
    S_ID[i]=temp2[j];
    i++;
    j++;
}
i=0;
j++;
while(temp2[j]!=';')
{
    S_name[i]=temp2[j];
    i++;
    j++;
}
i=0;
j++;
while(temp2[j]!=';')
{
    S_age[i]=temp2[j];
    i++;
    j++;
}
i=0;
j++;
while(temp2[j]!=';')
{
    S_bloodGroup[i]=temp2[j];
    i++;
    j++;
}
i=0;
j++;
while(temp2[j]!='\0')
{
    S_department[i]=temp2[j];
    i++;
    j++;
}
sscanf(S_ID,"%11d",&List2[g].studentID);

```

```

        sscanf(S_name,"%s", List2[g].name);
        sscanf(S_age,"%ld",&List2[g].age);
        sscanf(S_bloodGroup,"%s",List2[g].bloodGroup);
        sscanf(S_department,"%s",List2[g].department);
        g++;
    }
}
fclose(p);
///append
FILE* f;
f=fopen("E:\\Download\\CSE 4308\\grades.txt","a");

if(f==NULL)
{
    printf("Error in opening file");
    return 0;
}
else
{
    bool check=false;
    long long STUDENT_ID;
    float STUDENT_GPA;
    int STUDENT_SEMESTER;
    scanf("%lld %f %d",&STUDENT_ID,&STUDENT_GPA,&STUDENT_SEMESTER);
    for(int i=0; i<g ; i++)
    {
        if(STUDENT_ID==List2[i].studentID)
        {
            check=true;
        }
    }

    if(check)
    {
        if(STUDENT_GPA<2.50 || STUDENT_GPA>4.00)
        {
            printf("GPA is out of range");
            return 0;
        }
        else if(STUDENT_SEMESTER<1 || STUDENT_SEMESTER>8)
        {

```

```

        printf("Semester out of range");
        return 0;
    }
    for(int i=0; i<x ; i++)
    {
        if(STUDENT_ID==List1[i].id &&
List1[i].semester==STUDENT_SEMESTER)
        {
            printf("GPA for semester already exists");
            return 0;
        }
    }

    fprintf(f, "%lld;%.2f;%d\n", STUDENT_ID, STUDENT_GPA, STUDENT_SEMESTER);
    }
    else
    {
        printf("Student ID not found");
        return 0;
    }
}
fclose(f);
return 0;
}

```

Explanation of the solution of task 3

Our third task was to take Student ID as input and show his/her name and CGPA (average GPA for all the semesters he/she attended) and to print an error message if the Student ID did not exist in the database.

The working process of the third task is quite similar to the previous task. I read those two files and then validate the input information using a loop through the studentinfo file. If the input is valid then the programme will return the average value of the students GPA.

Final working code of task 3

```
#include<stdio.h>
#include<stdlib.h>
#include<stdbool.h>

typedef struct Grade
{
    long long int id;
    float gpa;
    int semester;
} gradeinfo;

typedef struct StudentInfo
{
    char name[10000];
    long long studentID;
    long age;
    char department[10000];
    char bloodGroup[10000];
} Info;

int x=0,g=0;
gradeinfo List1[10000];
Info List2[10000];

int main()
{
    ///1st file read
    FILE* fp;
    fp=fopen("E:\\Download\\CSE 4308\\grades.txt","r");
    if(fp==NULL)
    {
        printf("Error in opening file");
        return 0;
    }
    else
    {
        char temp[100000];
        while(fscanf(fp,"%s",temp)!=EOF)
        {
            long long int student_id;
```

```

float student_gpa;
int student_semester;
char ID[10000]= {'\0'};
char GPA[10000]= {'\0'};
char SEMESTER[10000]= {'\0'};
int i=0,j=0;
while(temp[j]!=';')
{
    ID[i]=temp[j];
    i++;
    j++;
}
i=0;
j++;
while(temp[j]!=';')
{
    GPA[i]=temp[j];
    i++;
    j++;
}
i=0;
j++;
while(temp[j]!='\0')
{
    SEMESTER[i]=temp[j];
    i++;
    j++;
}
sscanf(ID,"%11d",&student_id);
sscanf(GPA,"%f",&student_gpa);
sscanf(SEMESTER,"%d",&student_semester);
List1[x].id=student_id;
List1[x].gpa=student_gpa;
List1[x].semester=student_semester;
x++;
}

}
fclose(fp);
///2nd file read
FILE* p;

```



```

p=fopen("E:\\Download\\CSE 4308\\studentInfo.txt","r");
if(p==NULL)
{
    printf("Error in opening 2nd file");
    return 0;
}
else
{
    char temp2[10000];
    while(fscanf(fp,"%s",temp2)!=EOF)
    {
        char S_ID[100]= {'\0'};
        char S_name[10]= {'\0'};
        char S_age[15]= {'\0'};
        char S_bloodGroup[10]= {'\0'};
        char S_department[10]= {'\0'};
        long StudentID;
        long Age;

        int j=0;
        int i=0;
        while(temp2[j]!=';')
        {
            S_ID[i]=temp2[j];
            i++;
            j++;
        }
        i=0;
        j++;
        while(temp2[j]!=';')
        {
            S_name[i]=temp2[j];
            i++;
            j++;
        }
        i=0;
        j++;
        while(temp2[j]!=';')
        {
            S_age[i]=temp2[j];

```

```

        i++;
        j++;
    }
    i=0;
    j++;
    while(temp2[j]!=';')
    {
        S_bloodGroup[i]=temp2[j];
        i++;
        j++;
    }
    i=0;
    j++;
    while(temp2[j]!='\0')
    {
        S_department[i]=temp2[j];
        i++;
        j++;
    }
    sscanf(S_ID,"%lld",&List2[g].studentID);
    sscanf(S_name,"%s", List2[g].name);
    sscanf(S_age,"%ld",&List2[g].age);
    sscanf(S_bloodGroup,"%s",List2[g].bloodGroup);
    sscanf(S_department,"%s",List2[g].department);
    g++;
}
}
fclose(p);
///3rd task
int count=0;
bool check=false;
long long STUDENT_ID;
float sum;
int STUDENT_SEMESTER;
scanf("%lld",&STUDENT_ID);
for(int i=0; i<g ; i++)
{
    if(STUDENT_ID==List2[i].studentID)
    {
        check=true;
    }
}

```

```

    }
    if(check)
    {
        for(int i=0; i<x ; i++)
        {
            if(STUDENT_ID==List1[i].id)
            {
                sum = sum + List1[i].gpa;
                count++;
            }
        }
        float average;
        average = sum/count;
        printf("Average of ID %lld is : %.2f", STUDENT_ID, average);
    }
    else
    {
        printf("Student ID not found");
        return 0;
    }
    return 0;
}

```

Problems which I faced

This task made me realize the difficulties of the absence of database. It took me a day to complete the whole task. My file pointer was returning null value at first. Then I couldn't understand what to use to extract the string data and convert them into numerical data. It was hard for me to visualize the whole solution. I faced many syntax errors also . So it took me more than the lab hours to google my problems , solve them and finally complete the tasks.