Report

Student management system

Using queue

Problem statement:

Write a program to build a simple software for student information management system which can perform the following operations:

1- Store the first name of the student.
2- Store the last name of the student.
3- Store the unique Roll number for every student.
4- Store the GPA of every student.
5- Store the courses registered by the student.

Approach:

1- Add student details manually.
2- Add students details from text file.
3- Find student by a given roll number.
4- Find students by a given first name.
5- Find students registered in specific course.
6- Number of students registered.
7- Delete student by Roll number.
8- Update student information.

Implementation:

1- create structure to represent student details

2- create array of structures to act as a buffer that contains

All student's database limited by 100 record.

3- create structure to control the buffer and navigate through it using pointers and counter.

4- create functions that achieves our target

5- create Enum to represent status of each function

Lets see code

1-Structures, Enum and buffer:

as we mentioned before

-structure to represent student

object.

- Buffer to contain the whole database.
- structure to control buffer.
- Enum to represent status.

```
typedef struct {     // student information structure
    char fname[50];
    char lname[50];
    int roll;
    float GPA;
     int cid[10];
} st;
st buffer[100];   // the buffer contains students data
typedef struct{
    st *head;
    st *tail;
    st *base;
    int counter;
    int length;
} x;
typedef enum{
    fifo_no_error,
    fifo_full,
    fifo_empty,
    fifo_null,
    fifo_error

} fifo_buffer_state;
```

2- initialization:

this function takes pointer to structure

(controller buffer) and pointer to

Buffer and buffer length, then return status

-first it checks if these addresses exist or

Not then adjust pointers of controller to

Points to the buffer and initialize counter.

```
fifo_buffer_state fifo_init(x*fifo,st* buf,int lenght)
{
if(!fifo || !buf)
{
return fifo_null;
}
fifo->base=buf;
fifo->head=buf;
fifo->tail=buf;
fifo->length=lenght;
fifo->counter=0;
return fifo_no_error;
}
```

3- Check for Roll number

this function takes pointer to buffer controller and take integer number represents Roll number wanted to be check if is exist before or not because the

roll number must be a unique number for

a student.

It returns 0 if roll number found and 1

If not found.

```
int check_roll(x*fifo,int x)  // to check if roll number is exist before or not
{ int y;
 st* ptr=fifo->base;

for(y=0;y<(fifo->counter);y++)
{
    if(ptr->roll==x)
    {
        return 0;
    }
    ptr++;
}
 return 1;
}
```

4- add student manually

- this function to add students details manually takes pointer to

Buffer controller.

-starts with checking if the buffer is existing or not then checks if the buffer is full or not.

- then starts to take student information and check if the Roll number has been taken before or not using check_roll function.

- after that it takes another data like first name and last name and GPA

- finally, it takes the student courses that should have a valid course id and the function will handle any non-valid course id

- if everything is going well, the function finally prints the max number of students that the buffer can hold and the number of empty places for new records.

```
22  int check_roll(x*fifo,int x)  // to check if roll number is exist before or not
36  fifo_buffer_state add_student_manually(x*fifo)
37  {    int temp_int,y, x;
38      char temp_str[30];
39      if(!fifo->base || !fifo->head || !fifo->tail )  // check queue is exist or not
40      {
41      DPRINTF("database not exist  \n");
42      return fifo_null;
43      }
44    if(fifo->counter==fifo->length)   // check if full
45    {
46        DPRINTF("[ERROR] data base is full\n");
47        return fifo_full;
48    }
49    DPRINTF("-------------------------------------------\n");
50    DPRINTF("Add Student Details \n");
51    DPRINTF("-------------------------------------------\n");
52    DPRINTF("Enter the Roll Number\n");
53    gets(temp_str);
54    temp_int =atoi(temp_str);
55    if(check_roll(fifo,temp_int)==0)
56    {
57        DPRINTF("[ERROR] Roll Number is already taken before \n");
58        return fifo_error;
59    }
60    fifo->head->roll=atoi(temp_str);
61    DPRINTF("Enter First name of the student:\n");
62    gets(fifo->head->fname);
63    DPRINTF("Enter Last name of the student:\n");
64    gets(fifo->head->lname);
65    DPRINTF("Enter the GPA you obtained\n");
66    gets(temp_str);
67    fifo->head->GPA=atof(temp_str);
68    DPRINTF("Enter the course id of each course\n");
69    for(x=0;x<5;x++)
70    {
71        DPRINTF("course %d id :\n",x+1);
72        gets(temp_str);
73        y=atoi(temp_str);
74        if(y>0 && y<30)              // check if course id is available id
75        {
76            fifo->head->cid[x]=y;
77            continue;
78        }
79        DPRINTF("[ERROR] course id is not correct\n");
80        x--;
81    }
82    fifo->head++;
83    fifo->counter++;
84    DPRINTF("[INFO] Student Details are added successfully \n");
85    DPRINTF("-----------------------------------------\n");
86    DPRINTF("[INFO] the total number of students is : %d\n",fifo->counter);
87    DPRINTF("[INFO] you can add up to %d students \n",fifo->length);
88    DPRINTF("[INFO] you can add up to %d students \n",fifo->length - fifo->counter);
89    DPRINTF("-----------------------------------------\n");
90    return fifo_no_error;
91  }
```

5-Show all database

- this function responsible for display all data for all students registered.
- It starts with checking if buffer is existing or not and check if the buffer is empty or not.
- It loops on the buffer and extract all data
- Finally, it informs us the total number of students

```
fifo_buffer_state show_s(x*fifo)  // show all students information
{   st *current_stuednt =fifo->base;
    int x,y;
    if(!fifo->base || !fifo->head || !fifo->tail )  // check queue is exist or not
        {
        DPRINTF("database not exist  \n");
        return fifo_null;
        }
    if(fifo->counter==0)         // check if is empty
    {
       DPRINTF("[ERROR] database is empty \n");
       DPRINTF("-----------------------------------------\n");
       return fifo_empty;
    }
for(x=0;x< fifo->counter;x++)   // show students data
{
    DPRINTF("-----------------------------------------\n");
    DPRINTF("Student Roll number : %d\n",current_stuednt->roll);
    DPRINTF("Student first name : %s\n",current_stuednt->fname);
    DPRINTF("Student last name : %s\n",current_stuednt->lname);
    DPRINTF("Student GPA : %.2f\n",current_stuednt->GPA);
    for(y=0;y<5;y++)
    {
        DPRINTF("course %d id : %d \n",y+1,current_stuednt->cid[y]);
    }
    current_stuednt++;
}
DPRINTF("-----------------------------------------\n");
DPRINTF("total number of students : %d\n",fifo->counter);
return fifo_no_error;
}
```

6-find by roll number

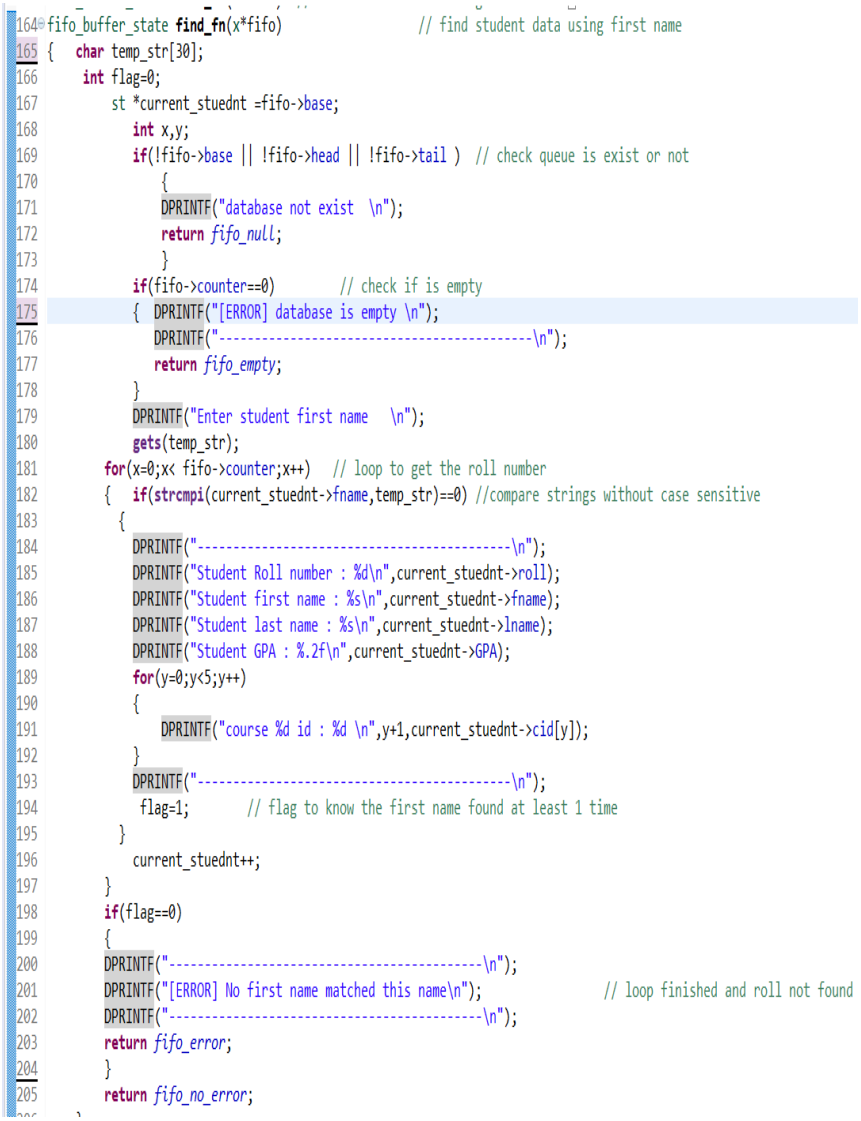
- this function responsible for find the student data by roll number
- It starts with checking if buffer is existing or not and check if the buffer is empty or not.
- Then loop on all database to get student information by roll number.
- If roll number is not found function will inform us.

```
fifo_buffer_state find_r1(x*fifo)  // find student data using Roll number
{   char temp_str[30];
    int temp_roll;
    st *current_stuednt =fifo->base;
       int x,y;
       if(!fifo->base || !fifo->head || !fifo->tail )  // check queue is exist or not
           {
           DPRINTF("database not exist  \n");
           return fifo_null;
           }
       if(fifo->counter==0)         // check if is empty
       {
          DPRINTF("[ERROR] database is empty \n");
          DPRINTF("-----------------------------------------\n");
          return fifo_empty;
       }
       DPRINTF("Enter student roll number  \n");
       gets(temp_str);
       temp_roll=atoi(temp_str);

   for(x=0;x< fifo->counter;x++)   // loop to get the roll number
   {   if(current_stuednt->roll==temp_roll)
     {
       DPRINTF("-----------------------------------------\n");
       DPRINTF("Student Roll number : %d\n",current_stuednt->roll);
       DPRINTF("Student first name : %s\n",current_stuednt->fname);
       DPRINTF("Student last name : %s\n",current_stuednt->lname);
       DPRINTF("Student GPA : %.2f\n",current_stuednt->GPA);
       for(y=0;y<5;y++)
       {
           DPRINTF("course %d id : %d \n",y+1,current_stuednt->cid[y]);
       }
       return fifo_no_error;
     }
       current_stuednt++;
   }
   DPRINTF("-----------------------------------------\n");
   DPRINTF("[ERROR] Roll number is not found\n");                // loop finished and roll not found
   DPRINTF("-----------------------------------------\n");
   return fifo_error;
}
```

7-Find by first name

- this function searches about all students that share the same student's first name
- It starts with checking if buffer is existing or not and check if the buffer is empty or not.
- Then loop on all database to get student information by first name.
- If roll the name is not found function will inform us.



```
fifo_buffer_state find_fn(x*fifo)                // find student data using first name
{   char temp_str[30];
    int flag=0;
        st *current_stuednt =fifo->base;
          int x,y;
          if(!fifo->base || !fifo->head || !fifo->tail )  // check queue is exist or not
              {
              DPRINTF("database not exist  \n");
              return fifo_null;
              }
          if(fifo->counter==0)         // check if is empty
          {  DPRINTF("[ERROR] database is empty \n");
             DPRINTF("----------------------------------------\n");
             return fifo_empty;
          }
          DPRINTF("Enter student first name   \n");
          gets(temp_str);
       for(x=0;x< fifo->counter;x++)   // loop to get the roll number
       {   if(strcmpi(current_stuednt->fname,temp_str)==0) //compare strings without case sensitive
         {
          DPRINTF("----------------------------------------\n");
          DPRINTF("Student Roll number : %d\n",current_stuednt->roll);
          DPRINTF("Student first name : %s\n",current_stuednt->fname);
          DPRINTF("Student last name : %s\n",current_stuednt->lname);
          DPRINTF("Student GPA : %.2f\n",current_stuednt->GPA);
          for(y=0;y<5;y++)
          {
              DPRINTF("course %d id : %d \n",y+1,current_stuednt->cid[y]);
          }
          DPRINTF("----------------------------------------\n");
           flag=1;        // flag to know the first name found at least 1 time
         }
           current_stuednt++;
       }
       if(flag==0)
       {
       DPRINTF("----------------------------------------\n");
       DPRINTF("[ERROR] No first name matched this name\n");                // loop finished and roll not found
       DPRINTF("----------------------------------------\n");
       return fifo_error;
       }
       return fifo_no_error;
```

8-find all students registered in a specific course using course id

-this function searches about all students that registered the same course

-we search using course id

-we loop on buffer and inside each student object we loop on courses if our course id matches any of student courses the function will print the student information.

- if the loop finished and there are no students registered this course the function will print message to inform us.

```
fifo_buffer_state find_c(x*fifo)  // display students info registered by course id
{        char temp_str[30];
int temp_course_id;
st *current_stuednt =fifo->base;
int x,y,z,flag=0;
if(!fifo->base || !fifo->head || !fifo->tail )  // check queue is exist or not
{
    DPRINTF("database not exist  \n");
    return fifo_null;
}
if(fifo->counter==0)         // check if is empty
{
    DPRINTF("[ERROR] database is empty \n");
    DPRINTF("-------------------------------------------\n");
    return fifo_empty;
}
DPRINTF("Enter course id number  \n");
gets(temp_str);
temp_course_id=atoi(temp_str);
for(x=0;x< fifo->counter;x++)   // loop to get the course id

{   for(z=0;z<5;z++)
{
    if(current_stuednt->cid[z]==temp_course_id)  // if course id matches
    {
        DPRINTF("-------------------------------------------\n");
        DPRINTF("Student Roll number : %d\n",current_stuednt->roll);
        DPRINTF("Student first name : %s\n",current_stuednt->fname);
        DPRINTF("Student last name : %s\n",current_stuednt->lname);
        DPRINTF("Student GPA : %.2f\n",current_stuednt->GPA);
        for(y=0;y<5;y++)
        {  DPRINTF("course %d id : %d \n",y+1,current_stuednt->cid[y]);}
        flag=1; // found at least one student
    }
}
current_stuednt++;
}
if(flag==0)
{
    DPRINTF("-------------------------------------------\n");
    DPRINTF("[ERROR] no student registered \n");                // loop finished a
    DPRINTF("-------------------------------------------\n");
    return fifo_error;
```

9- Total number of students

This function gets the total number of students in the database

```
fifo_buffer_state tot_s(x*fifo)                    // total number of student
{

if(!fifo->base || !fifo->head || !fifo->tail )  // check queue is exist or not
{
    DPRINTF("database not exist  \n");
    return fifo_null;
}
if(fifo->counter==0)         // check if is empty
{
    DPRINTF("[ERROR] database is empty \n");
    DPRINTF("-------------------------------------------\n");
    return fifo_empty;
}
DPRINTF("-------------------------------------------\n");
DPRINTF("[INFO] the total number of students is : %d\n",fifo->counter);
DPRINTF("[INFO] you can add up to %d students \n",fifo->length);
DPRINTF("[INFO] you can add more about %d students \n",fifo->length - fifo->counter);
DPRINTF("-------------------------------------------\n");
return fifo_no_error;
}
```

10- Delete student using roll number

- this function deletes student registration by roll number.
- First it loops on database to find the roll number
- It displays student information before delete
- Then asks us to confirm deleting process
- If roll number is not found it will inform us
- If we enter wrong option It will inform and back to main menu
- This function uses another function called shift_buffer we will discuss it later.

```
fifo_buffer_state del_s(x*fifo)                  // delete student by roll num
{   char temp_str[30];
    int x,y,temp_roll,index=0;
    st *current_stuednt=fifo->base;
    if(!fifo->base || !fifo->head || !fifo->tail )  // check queue is exist or not
    {
        DPRINTF("database not exist  \n");
        return fifo_null;
    }
    if(fifo->counter==0)         // check if is empty
    {
        DPRINTF("[ERROR] database is empty \n");
        DPRINTF("-----------------------------------------\n");
        return fifo_empty;
    }
    DPRINTF("Enter student roll number  \n");
    gets(temp_str);
    temp_roll=atoi(temp_str);

    for(x=0;x< fifo->counter;x++)   // loop to get the roll number
    {   if(current_stuednt->roll==temp_roll)
    {
        DPRINTF("-----------------------------------------\n");
        DPRINTF("Student Roll number : %d\n",current_stuednt->roll);
        DPRINTF("Student first name : %s\n",current_stuednt->fname);
        DPRINTF("Student last name : %s\n",current_stuednt->lname);
        DPRINTF("Student GPA : %.2f\n",current_stuednt->GPA);
        for(y=0;y<5;y++)
        {
            DPRINTF("course %d id : %d \n",y+1,current_stuednt->cid[y]);
        }
        DPRINTF("-----------------------------------------\n");
        DPRINTF("Delete student 1-yes 2-No \n");
        DPRINTF("-----------------------------------------\n");
        gets(temp_str);
        temp_roll=atoi(temp_str);
        if(temp_roll==1){
            shift_buffer(index,fifo);  // to shift buffer and override on location wanted to be deleted
            fifo->head--;
            fifo->counter--;
            return fifo_no_error;
        }
        else if(temp_roll==0)
        {
            DPRINTF("-----------------------------------------\n");
            DPRINTF("--------Process canceled-------\n");
            return fifo_no_error;
        }
        else
        {   DPRINTF("-----------------------------------------\n");
            DPRINTF("[ERROR]wrong choice ..\n Uncompleted process...\n back to main menu .....  \n");
            return fifo_no_error;
        }
    }
    current_stuednt++;
    index++;               //to find location in the buffer
    }
    DPRINTF("-----------------------------------------\n");
    DPRINTF("[ERROR] Roll number is not found\n");                // loop finished and roll not found
    DPRINTF("-----------------------------------------\n");
    return fifo_error;
```

11 – shift buffer

This function responsible for shifting the deleted object location and all objects after it left to fill the free location of deleted object inside the buffer.

```
fifo_buffer_state shift_buffer(int index,x*fifo) // to shift buffer and override
{
    int x;
    for(x=index;x<fifo->counter;x++)
    {
        buffer[x]=buffer[x+1];
    }
        DPRINTF("-----------------------------------------\n");
        DPRINTF("student deleted successfully\n");
        DPRINTF("-----------------------------------------\n");
        return fifo_no_error;
}
```

12-update student information

-this function updates a specific data in a previous registered student's data

- this data could be first name or second name or GPA or courses.

- it navigates on database based on student roll number.

- first it displays the data of student and allow us to choose the data wanted to be updated.

- if you entered wrong choice the function will return to the main menu with a message that is wrong choice.

-if you update data of student the function will display the student's information after updating.

-if you entered a wrong roll number it will tell you that is wrong roll number and back to main menu.

```
fifo_buffer_state up_s(x*fifo)                  // update student by roll num
{
     char temp_str[30];
    int x,y,i,j,temp_option, temp_roll;
    st *current_stuednt =fifo->base;
    if(!fifo->base || !fifo->head || !fifo->tail )  // check queue is exist or not
    {
        DPRINTF("database not exist  \n");
        return fifo_null;
    }
    if(fifo->counter==0)         // check if is empty
    {
        DPRINTF("[ERROR] database is empty \n");
        DPRINTF("-----------------------------------------\n");
        return fifo_empty;
    }
    DPRINTF("Enter student roll number  \n");
    gets(temp_str);
    temp_roll=atoi(temp_str);

    for(x=0;x< fifo->counter;x++)   // loop to get the roll number
    {   if(current_stuednt->roll==temp_roll)
    {
        DPRINTF("-----------------------------------------\n");
        DPRINTF("Student Roll number : %d\n",current_stuednt->roll);
        DPRINTF("Student first name : %s\n",current_stuednt->fname);
        DPRINTF("Student last name : %s\n",current_stuednt->lname);
        DPRINTF("Student GPA : %.2f\n",current_stuednt->GPA);
        for(y=0;y<5;y++)
        {
            DPRINTF("course %d id : %d \n",y+1,current_stuednt->cid[y]);
        }
        DPRINTF("-----------------------------------------\n");
        DPRINTF("Enter option to update data\n");
        DPRINTF("1- first name\n");
        DPRINTF("2- last name \n");
        DPRINTF("3- GPA\n");
        DPRINTF("4- courses \n");
        DPRINTF("-----------------------------------------\n");
        gets(temp_str);
        temp_option=atoi(temp_str);
        switch(temp_option)
        {
        case 1 :
        {
            DPRINTF("Enter New first name\n");
            gets(current_stuednt->fname);
            break;
        }
        case 2 :
        {
            DPRINTF("Enter New second name\n");
            gets(current_stuednt->lname);
            break;

        }
        case 3 :
        {
            DPRINTF("Enter New GPA \n");
            gets(temp_str);
            current_stuednt->GPA=atof(temp_str);
            break;

        }
        case 4 :
        {
            DPRINTF("Enter the course id of each course\n");
            for(i=0;i<5;i++)
            {
                DPRINTF("course %d id :\n",i+1);
                gets(temp_str);
                j=atoi(temp_str);
                if(j>0 && j<30)             // check if course id is available id
                {
                    current_stuednt->cid[i]=j;
                    continue;
                }
                DPRINTF("[ERROR] course id is not correct\n");
                i--;
            }
              break;

        }
        default :
       {
           DPRINTF("[ERROR] wrong choice \n");
           return fifo_error;
       }
     }
        // print student information after update
        DPRINTF("-----------------------------------------\n");
        DPRINTF("information updated successfully \n");
        DPRINTF("Student information after update \n");
        DPRINTF("-----------------------------------------\n");
                DPRINTF("Student Roll number : %d\n",current_stuednt->roll);
                DPRINTF("Student first name : %s\n",current_stuednt->fname);
                DPRINTF("Student last name : %s\n",current_stuednt->lname);
                DPRINTF("Student GPA : %.2f\n",current_stuednt->GPA);
                for(y=0;y<5;y++)
                {
                    DPRINTF("course %d id : %d \n",y+1,current_stuednt->cid[y]);
                }
                return fifo_no_error;
    }
    current_stuednt++;
    }
    DPRINTF("-----------------------------------------\n");
    DPRINTF("[ERROR] Roll number is not found\n");              // loop finished an
    DPRINTF("-----------------------------------------\n");
    return fifo_error;
}
```

13- Add student from text File

-this function reads from text file and save students information into data base.

-after checking buffer exist or not and is full or not the function stars connection with the text file

- it checks if roll number is taken or not and if this condition happened you get message with line error in text file and skip this record.

- if any student has non-valid course id, a message gets printed and this student is skipped.

-if buffer size reaches the full size

The function will stop adding and

A message gets printed with the number of students added and the remaining students.

-If all things are ok a message is printed with the number of students added and the number of error in students information

```
fifo_buffer_state add_student_file(x*fifo)                // add students into using text file
{
    char f_name[50];
    char l_name[50];
    int roll_num,cid[5],x,file_count=0,flag=0;
    float GPA;
    int line =0;
    if(!fifo->base || !fifo->head || !fifo->tail )  // check queue is exist or not
    {
        DPRINTF("database not exist  \n");
        return fifo_null;
    }
    if(fifo->counter==fifo->length)   // check if full
    {
        DPRINTF("[ERROR] data base is full\n");
        return fifo_full;
    }
    FILE *p_file = fopen("text.txt","r");
    if(p_file==NULL)
    {
        DPRINTF("----------------------------- \n");
        DPRINTF("[ERROR] File not found \n");
        return fifo_error;
    }


    // reading from file

    while( fscanf(p_file,"%d %s %s %f %d %d %d %d %d [^\n]", &roll_num, f_name,l_name,&GPA,&cid[0],&cid[1],&cid[2],&cid[3],&cid[4])!=EOF)
    {
        if(fifo->counter==fifo->length)
        {   DPRINTF("----------------------------\n");
        DPRINTF("[ERROR] data base is full\n");
        DPRINTF("[INFO] students added : %d\n",file_count);
        DPRINTF("[INFO] remaining students due to size or errors are :%d\n",line-file_count);

        return fifo_full;
        }
        if(check_roll(fifo,roll_num)==0)
        {
            DPRINTF("[ERROR] IN line %d : Roll Number is already taken before \n",line);
            line++;
            continue; // to skip this student
        }
        fifo->head->roll=roll_num;
        fifo->head->GPA=GPA;
        strcpy(fifo->head->fname,f_name);
        strcpy(fifo->head->lname,l_name);
        for(x=0;x<5;x++)
        {
            if(cid[x]<0 || cid[x]>30)
            {
                flag=1;    // that there is non-valid course id
                break;
            }
            fifo->head->cid[x]=cid[x];
        }
        if(flag==1)  // non valid course id
        {
            DPRINTF("[ERROR] IN line %d : non valid course id we will skip this student \n",line);
            line++;
            continue; // to skip this student
        }
        fifo->head++;
        fifo->counter++;
        line++;
        file_count++;    // to record successful records
    }
    DPRINTF("\nEnd of file.\n");

    // close connection
    fclose(p_file);
    DPRINTF("[INFO] students added : %d\n",file_count);
    DPRINTF("[INFO] remaining students due to errors are :%d\n",line-file_count);
    return fifo_error;
}
```

14- main function

```c
#include "queue.h"

int main(void) {
    int temp;
    x buffer_controller;  // that controls student buffer
    fifo_init(&buffer_controller,buffer,100);

    DPRINTF("Welcome to the Student Management System\n");
    while(1)
    {
        DPRINTF("-----------------------------------------\n");
        DPRINTF("Choose The Task that you want to perform\n");
        DPRINTF("1. Add the Student Details Manually\n");
        DPRINTF("2. Add the Student Details From Text File\n");
        DPRINTF("3. Find the Student Details by Roll Number\n");
        DPRINTF("4. Find the Student Details by First Name\n");
        DPRINTF("5. Find the Student Details by Course ID\n");
        DPRINTF("6. Find the Total number of Students\n");
        DPRINTF("7. Delete the Student Details by Roll Number \n");
        DPRINTF("8. Update the Student Details by Roll Number \n");
        DPRINTF("9. Show all information\n");
        DPRINTF(" Enter your choice to perform the task\n");
        scanf("%d",&temp);
        switch(temp)
        {
        case 1 :
        {
            add_student_manually(&buffer_controller);
            break;
        }
        case 2 :
        {
            add_student_file(&buffer_controller);
            break;
        }
        case 3 :
        {
            find_r1(&buffer_controller);
            break;
        }
        case 4 :
        {
            find_fn(&buffer_controller);
            break;
        }
        case 5 :
        {
            find_c(&buffer_controller);
            break;
        }
        case 6 :
        {
            tot_s(&buffer_controller);
            break;
        }
        case 7 :
        {
            del_s(&buffer_controller);
            break;
        }
        case 8 :
        {
            up_s(&buffer_controller);
            break;
        }
        case 9 :
        {
            show_s(&buffer_controller);
            break;
        }
        default :
        {
            DPRINTF("Wrong choice\n");

        }
        }
```

Testing code logic

1-Adding students manually

We will add 4 students with the same way.

```
Welcome to the Student Management System
-------------------------------------------
Choose The Task that you want to perform
1. Add the Student Details Manually
2. Add the Student Details From Text File
3. Find the Student Details by Roll Number
4. Find the Student Details by First Name
5. Find the Student Details by Course ID
6. Find the Total number of Students
7. Delete the Student Details by Roll Number
8. Update the Student Details by Roll Number
9. Show all information
 Enter your choice to perform the task
1
-------------------------------------------
Add Student Details
-------------------------------------------
Enter the Roll Number
1
Enter First name of the student:
shady
Enter Last name of the student:
mamdouh
Enter the GPA you obtained
2.5
Enter the course id of each course
course 1 id :
1
course 2 id :
2
course 3 id :
3
course 4 id :
4
course 5 id :
5
[INFO] Student Details are added successfully
-------------------------------------------
[INFO] the total number of students is : 1
[INFO] you can add up to 100 students
[INFO] you can add more about 99 students
```

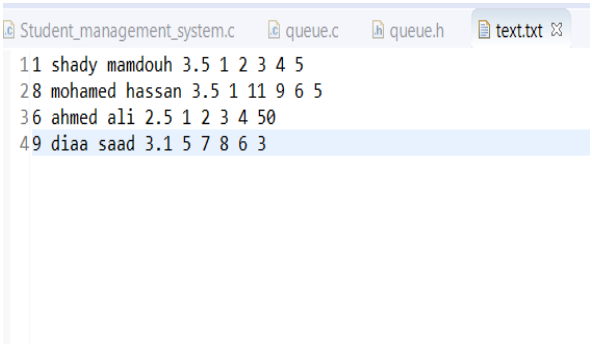
```
9. Show all information
 Enter your choice to perform the task
9
-------------------------------------------
Student Roll number : 1
Student first name : shady
Student last name : mamdouh
Student GPA : 2.50
course 1 id : 1
course 2 id : 2
course 3 id : 3
course 4 id : 4
course 5 id : 5
-------------------------------------------
Student Roll number : 2
Student first name : keroles
Student last name : shenouda
Student GPA : 3.80
course 1 id : 10
course 2 id : 5
course 3 id : 3
course 4 id : 2
course 5 id : 7
-------------------------------------------
Student Roll number : 3
Student first name : ahmed
Student last name : tawfek
Student GPA : 2.00
course 1 id : 20
course 2 id : 7
course 3 id : 2
course 4 id : 5
course 5 id : 1
-------------------------------------------
Student Roll number : 4
Student first name : shady
Student last name : essam
Student GPA : 2.50
course 1 id : 3
course 2 id : 5
```

2-Adding from text file

```
Choose The Task that you want to perform
1. Add the Student Details Manually
2. Add the Student Details From Text File
3. Find the Student Details by Roll Number
4. Find the Student Details by First Name
5. Find the Student Details by Course ID
6. Find the Total number of Students
7. Delete the Student Details by Roll Number
8. Update the Student Details by Roll Number
9. Show all information
 Enter your choice to perform the task
2
[ERROR] IN line 0 : Roll Number is already taken before
[ERROR] IN line 2 : non valid course id we will skip this student

End of file.
[INFO] students added : 2
[INFO] remaining students due to errors are :2
-------------------------------------------
```

text.txt



We have 2 students will added successfully, and 2 students will not be added

One of them has roll number (roll number 1) which is taken before and another entered a non-valid course id(50) as we know the valid course id range from 1 to 29

so look at text file

shady and **ahmed** will not be added.

Mohamed and **diaa** will be added successfully.

Before add

```
9. Show all information
 Enter your choice to perform the task
9
-----------------------------------------
Student Roll number : 1
Student first name : shady
Student last name : mamdouh
Student GPA : 2.50
course 1 id : 1
course 2 id : 2
course 3 id : 3
course 4 id : 4
course 5 id : 5
-----------------------------------------
Student Roll number : 2
Student first name : keroles
Student last name : shenouda
Student GPA : 3.80
course 1 id : 10
course 2 id : 5
course 3 id : 3
course 4 id : 2
course 5 id : 7
-----------------------------------------
Student Roll number : 3
Student first name : ahmed
Student last name : tawfek
Student GPA : 2.00
course 1 id : 20
course 2 id : 7
course 3 id : 2
course 4 id : 5
course 5 id : 1
-----------------------------------------
Student Roll number : 4
Student first name : shady
Student last name : essam
Student GPA : 2.50
course 1 id : 3
course 2 id : 5
```

after add

```
9. Show all information
 Enter your choice to perform the task
9
-----------------------------------------
Student Roll number : 1
Student first name : shady
Student last name : mamdouh
Student GPA : 2.50
course 1 id : 1
course 2 id : 2
course 3 id : 3
course 4 id : 4
course 5 id : 5
-----------------------------------------
Student Roll number : 2
Student first name : keroles
Student last name : shenouda
Student GPA : 3.80
course 1 id : 10
course 2 id : 5
course 3 id : 3
course 4 id : 2
course 5 id : 7
-----------------------------------------
Student Roll number : 3
Student first name : ahmed
Student last name : tawfek
Student GPA : 2.00
course 1 id : 20
course 2 id : 7
course 3 id : 2
course 4 id : 5
course 5 id : 1
-----------------------------------------
Student Roll number : 4
Student first name : shady
Student last name : essam
Student GPA : 2.50
course 1 id : 3
course 2 id : 5
course 3 id : 8
course 4 id : 12
course 5 id : 6
-----------------------------------------
Student Roll number : 8
Student first name : mohamed
Student last name : hassan
Student GPA : 3.50
course 1 id : 1
course 2 id : 11
course 3 id : 9
course 4 id : 6
course 5 id : 5
-----------------------------------------
Student Roll number : 9
Student first name : diaa
Student last name : saad
Student GPA : 3.10
course 1 id : 5
course 2 id : 7
course 3 id : 8
course 4 id : 6
course 5 id : 3
-----------------------------------------
total number of students : 6
-----------------------------------------
```

3-get by roll number

```
Choose The Task that you want to perform
1. Add the Student Details Manually
2. Add the Student Details From Text File
3. Find the Student Details by Roll Number
4. Find the Student Details by First Name
5. Find the Student Details by Course ID
6. Find the Total number of Students
7. Delete the Student Details by Roll Number
8. Update the Student Details by Roll Number
9. Show all information
 Enter your choice to perform the task
3
Enter student roll number
8
-----------------------------------------
Student Roll number : 8
Student first name : mohamed
Student last name : hassan
Student GPA : 3.50
course 1 id : 1
course 2 id : 11
course 3 id : 9
course 4 id : 6
course 5 id : 5
-----------------------------------------
```

4-get all students share the same first name

We have 2 students share first name

Shady

```
-----------------------------------------
Choose The Task that you want to perform
1. Add the Student Details Manually
2. Add the Student Details From Text File
3. Find the Student Details by Roll Number
4. Find the Student Details by First Name
5. Find the Student Details by Course ID
6. Find the Total number of Students
7. Delete the Student Details by Roll Number
8. Update the Student Details by Roll Number
9. Show all information
 Enter your choice to perform the task
4
Enter student first name
shady
-----------------------------------------
Student Roll number : 1
Student first name : shady
Student last name : mamdouh
Student GPA : 2.50
course 1 id : 1
course 2 id : 2
course 3 id : 3
course 4 id : 4
course 5 id : 5
-----------------------------------------
-----------------------------------------
Student Roll number : 4
Student first name : shady
Student last name : essam
Student GPA : 2.50
course 1 id : 3
course 2 id : 5
course 3 id : 8
course 4 id : 12
course 5 id : 6
-----------------------------------------
```

5-Get all students registered in a specific course id

For ex course id: 2

we have 3 students from 6 registered in course id 2

```
4. Find the Student Details by First Name
5. Find the Student Details by Course ID
6. Find the Total number of Students
7. Delete the Student Details by Roll Number
8. Update the Student Details by Roll Number
9. Show all information
 Enter your choice to perform the task
5
Enter course id number
2
-------------------------------------------
Student Roll number : 1
Student first name : shady
Student last name : mamdouh
Student GPA : 2.50
course 1 id : 1
course 2 id : 2
course 3 id : 3
course 4 id : 4
course 5 id : 5
-------------------------------------------
Student Roll number : 2
Student first name : keroles
Student last name : shenouda
Student GPA : 3.80
course 1 id : 10
course 2 id : 5
course 3 id : 3
course 4 id : 2
course 5 id : 7
-------------------------------------------
Student Roll number : 3
Student first name : ahmed
Student last name : tawfek
Student GPA : 2.00
course 1 id : 20
course 2 id : 7
course 3 id : 2
course 4 id : 5
course 5 id : 1
-------------------------------------------
```

6-Get total number of students

we have 6 students

```
-------------------------------------------
Choose The Task that you want to perform
1. Add the Student Details Manually
2. Add the Student Details From Text File
3. Find the Student Details by Roll Number
4. Find the Student Details by First Name
5. Find the Student Details by Course ID
6. Find the Total number of Students
7. Delete the Student Details by Roll Number
8. Update the Student Details by Roll Number
9. Show all information
 Enter your choice to perform the task
6
-------------------------------------------
[INFO] the total number of students is : 6
[INFO] you can add up to 100 students
[INFO] you can add more about 94 students
-------------------------------------------
```

7- Delete student by roll number

If you enter the roll number to be delete the function displays the student information and asks you to confirm deleting process.

```
-----------------------------------------
Choose The Task that you want to perform
1. Add the Student Details Manually
2. Add the Student Details From Text File
3. Find the Student Details by Roll Number
4. Find the Student Details by First Name
5. Find the Student Details by Course ID
6. Find the Total number of Students
7. Delete the Student Details by Roll Number
8. Update the Student Details by Roll Number
9. Show all information
 Enter your choice to perform the task
7
Enter student roll number
3
-----------------------------------------
Student Roll number : 3
Student first name : ahmed
Student last name : tawfek
Student GPA : 2.00
course 1 id : 20
course 2 id : 7
course 3 id : 2
course 4 id : 5
course 5 id : 1
-----------------------------------------
Delete student 1-yes 2-No
-----------------------------------------
1
-----------------------------------------
student deleted successfully
-----------------------------------------
```

Student “ahmed tawfek” has been

Deleted and total number of students decreased by 1

-this function uses another function to shift the deleted location from the buffer

Show all students after delete

```
Choose The Task that you want to perform
1. Add the Student Details Manually
2. Add the Student Details From Text File
3. Find the Student Details by Roll Number
4. Find the Student Details by First Name
5. Find the Student Details by Course ID
6. Find the Total number of Students
7. Delete the Student Details by Roll Number
8. Update the Student Details by Roll Number
9. Show all information
 Enter your choice to perform the task
9
-----------------------------------------
Student Roll number : 1
Student first name : shady
Student last name : mamdouh
Student GPA : 2.50
course 1 id : 1
course 2 id : 2
course 3 id : 3
course 4 id : 4
course 5 id : 5
-----------------------------------------
Student Roll number : 2
Student first name : keroles
Student last name : shenouda
Student GPA : 3.80
course 1 id : 10
course 2 id : 5
course 3 id : 3
course 4 id : 2
course 5 id : 7
-----------------------------------------
Student Roll number : 4
Student first name : shady
Student last name : essam
Student GPA : 2.50
course 1 id : 3
course 2 id : 5
course 3 id : 8
course 4 id : 12
course 5 id : 6
-----------------------------------------
Student Roll number : 8
Student first name : mohamed
Student last name : hassan
Student GPA : 3.50
course 1 id : 1
course 2 id : 11
course 3 id : 9
course 4 id : 6
course 5 id : 5
-----------------------------------------
Student Roll number : 9
Student first name : diaa
Student last name : saad
Student GPA : 3.10
course 1 id : 5
course 2 id : 7
course 3 id : 8
course 4 id : 6
course 5 id : 3
-----------------------------------------
total number of students : 5
-----------------------------------------
```

8- Update specific data of a student

For example:

update Diaa's GPA

```
7. Delete the Student Details by Roll Number
8. Update the Student Details by Roll Number
9. Show all information
 Enter your choice to perform the task
8
Enter student roll number
9
-------------------------------------------
Student Roll number : 9
Student first name : diaa
Student last name : saad
Student GPA : 3.10
course 1 id : 5
course 2 id : 7
course 3 id : 8
course 4 id : 6
course 5 id : 3
-------------------------------------------
Enter option to update data
1- first name
2- last name
3- GPA
4- courses
-------------------------------------------
3
Enter New GPA
2.8
-------------------------------------------
information updated successfully
Student information after update
-------------------------------------------
Student Roll number : 9
Student first name : diaa
Student last name : saad
Student GPA : 2.80
course 1 id : 5
course 2 id : 7
course 3 id : 8
course 4 id : 6
course 5 id : 3
-------------------------------------------
```

update Diaa courses id

note: if you try to enter a non-valid course id an error message will be printed as shown below:

```
Student GPA : 2.80
course 1 id : 5
course 2 id : 7
course 3 id : 8
course 4 id : 6
course 5 id : 3
-------------------------------------------
Enter option to update data
1- first name
2- last name
3- GPA
4- courses
-------------------------------------------
4
Enter the course id of each course
course 1 id :
1
course 2 id :
90
[ERROR] course id is not correct
course 2 id :
2
course 3 id :
3
course 4 id :
4
course 5 id :
5
-------------------------------------------
information updated successfully
Student information after update
-------------------------------------------
Student Roll number : 9
Student first name : diaa
Student last name : saad
Student GPA : 2.80
course 1 id : 1
course 2 id : 2
course 3 id : 3
course 4 id : 4
course 5 id : 5
-------------------------------------------
```

9-show all database

```
-----------------------------------------
Student Roll number : 1
Student first name : shady
Student last name : mamdouh
Student GPA : 2.50
course 1 id : 1
course 2 id : 2
course 3 id : 3
course 4 id : 4
course 5 id : 5
-----------------------------------------
Student Roll number : 2
Student first name : keroles
Student last name : shenouda
Student GPA : 3.80
course 1 id : 10
course 2 id : 5
course 3 id : 3
course 4 id : 2
course 5 id : 7
-----------------------------------------
Student Roll number : 4
Student first name : shady
Student last name : essam
Student GPA : 2.50
course 1 id : 3
course 2 id : 5
course 3 id : 8
course 4 id : 12
course 5 id : 6
-----------------------------------------
Student Roll number : 8
Student first name : mohamed
Student last name : hassan
Student GPA : 3.50
course 1 id : 1
course 2 id : 11
course 3 id : 9
course 4 id : 6
course 5 id : 5
-----------------------------------------
Student Roll number : 9
Student first name : diaa
Student last name : saad
Student GPA : 2.80
course 1 id : 1
course 2 id : 2
course 3 id : 3
course 4 id : 4
course 5 id : 5
-----------------------------------------
total number of students : 5
-----------------------------------------
```

Good luck