Learn IN Depth

Be Professional In Embedded System

Mastering Embedded System Online Diploma

➢ **www.learn-in-depth.com**

Project Number	**second Term (Final Project 2)**
Project Name	**Student Management System**
Name	**Osama Youssef Tawadrous**
My Profile	**https://www.learn-in-depth.com/online-diploma/osamayoussef996%40gmail.com**

Table OF Contents :

➢ Introduction :

A simple software for student information management system which can perform the following operations :

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

➢ Approach :

The idea is to form an individual functions for every operation. All the functions are unified to form software.

1. Add student details from file.
2. Add student details manually.
3. Find the student by the given roll number.
4. Find the student by the given first name.
5. Find the student registered in a course.
6. Count number of students.
7. Delete a student by the given roll number.
8. Update a student by the given roll number.
9. Exit the program

➢ Idea behind operation :

The software will consist of 4 files main.c, Student.c, Student.h and Students.txt.

The main.c will contain the interface and calling the function.

Student.c will contain the global variables and the body of the functions.

Student.h will contain the prototypes of The APIs , macros and student information

students.txt will contain the data of students , space is the tokan between the data .

1. We will have array of struct of configurable variable TotalNumber which is the max number of students can system manipulate .
2. Each struct contains the student details.
3. We will have a global index which refers to the number of students.
4. The index is initialized with zero and can go up to 50.

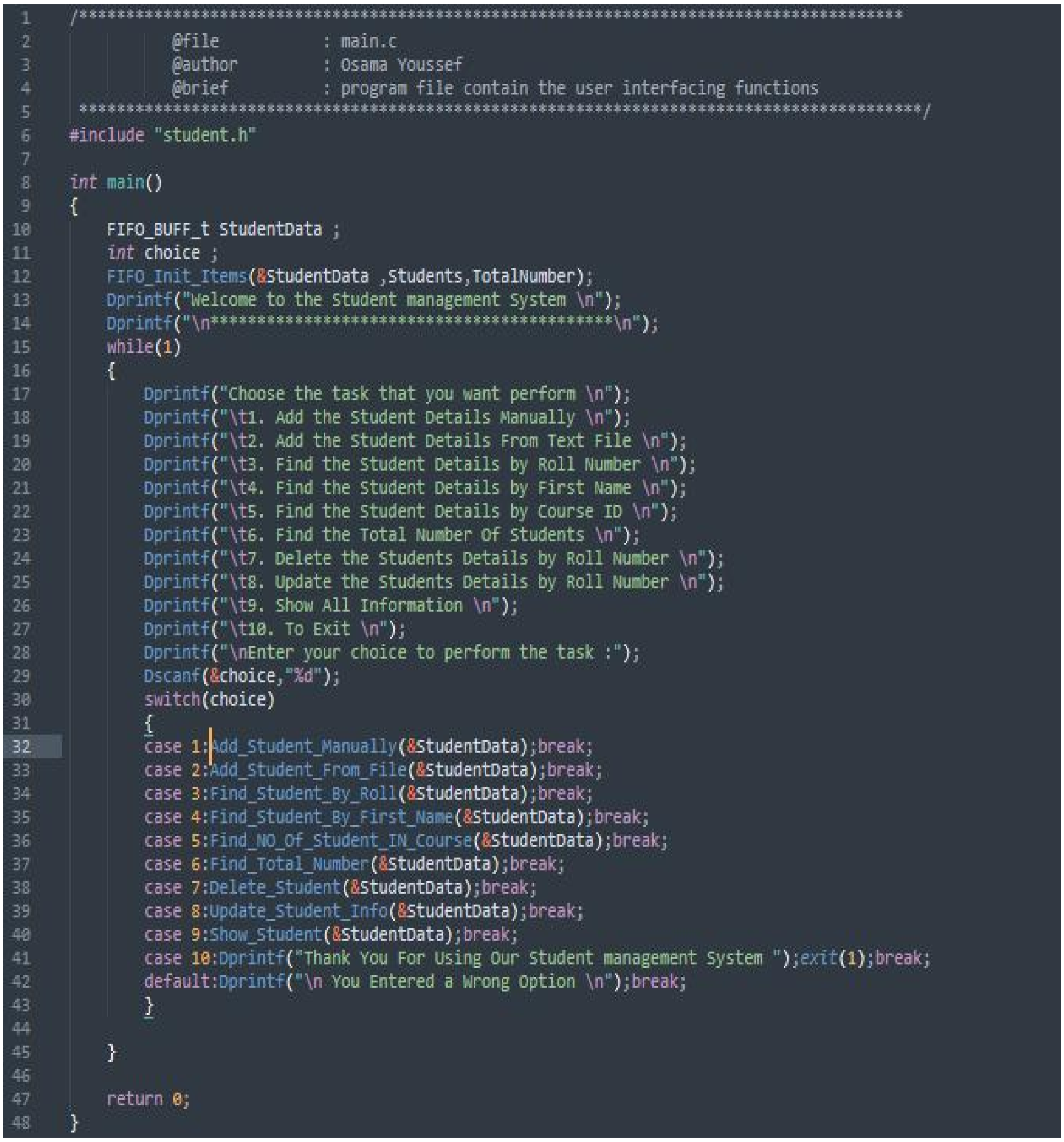
Here in our project , we will use Queue(FIFO) Data Structure , which can manipulate the students to be the first student enter the system , is the first student outer the system .

➢Codes Implementation and Testing :

❖ Main.c Code :

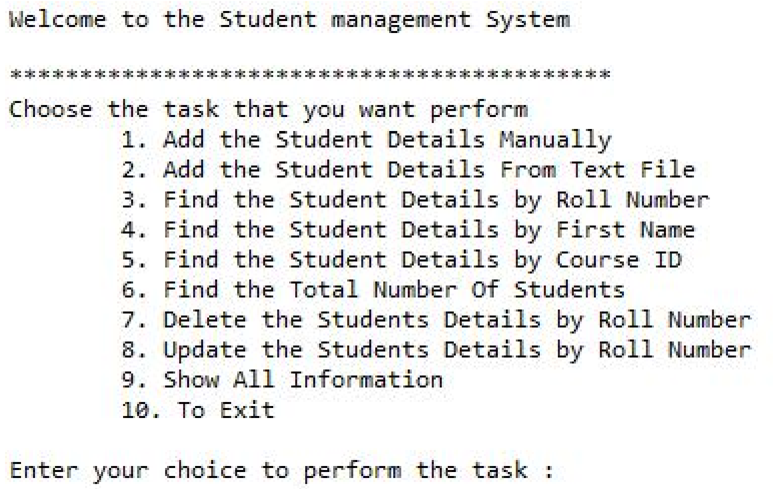
The main.c is just a simple file which consist of:

- Infinite loop while user enter option 10
- FIFO initialization Items
- Show all options in our system to the user
- Calling the program functions through switch cases.



```
/******************************************************************************************
		@file		: main.c
		@author		: Osama Youssef
		@brief		: program file contain the user interfacing functions
 ******************************************************************************************/
#include "student.h"

int main()
{
	FIFO_BUFF_t StudentData ;
	int choice ;
	FIFO_Init_Items(&StudentData ,Students,TotalNumber);
	Dprintf("Welcome to the Student management System \n");
	Dprintf("\n******************************************\n");
	while(1)
	{
		Dprintf("Choose the task that you want perform \n");
		Dprintf("\t1. Add the Student Details Manually \n");
		Dprintf("\t2. Add the Student Details From Text File \n");
		Dprintf("\t3. Find the Student Details by Roll Number \n");
		Dprintf("\t4. Find the Student Details by First Name \n");
		Dprintf("\t5. Find the Student Details by Course ID \n");
		Dprintf("\t6. Find the Total Number Of Students \n");
		Dprintf("\t7. Delete the Students Details by Roll Number \n");
		Dprintf("\t8. Update the Students Details by Roll Number \n");
		Dprintf("\t9. Show All Information \n");
		Dprintf("\t10. To Exit \n");
		Dprintf("\nEnter your choice to perform the task :");
		Dscanf(&choice,"%d");
		switch(choice)
		{
		case 1:Add_Student_Manually(&StudentData);break;
		case 2:Add_Student_From_File(&StudentData);break;
		case 3:Find_Student_By_Roll(&StudentData);break;
		case 4:Find_Student_By_First_Name(&StudentData);break;
		case 5:Find_NO_Of_Student_IN_Course(&StudentData);break;
		case 6:Find_Total_Number(&StudentData);break;
		case 7:Delete_Student(&StudentData);break;
		case 8:Update_Student_Info(&StudentData);break;
		case 9:Show_Student(&StudentData);break;
		case 10:Dprintf("Thank You For Using Our Student management System ");exit(1);break;
		default:Dprintf("\n You Entered a Wrong Option \n");break;
		}

	}

	return 0;
}
```

❖ Main.c Test :

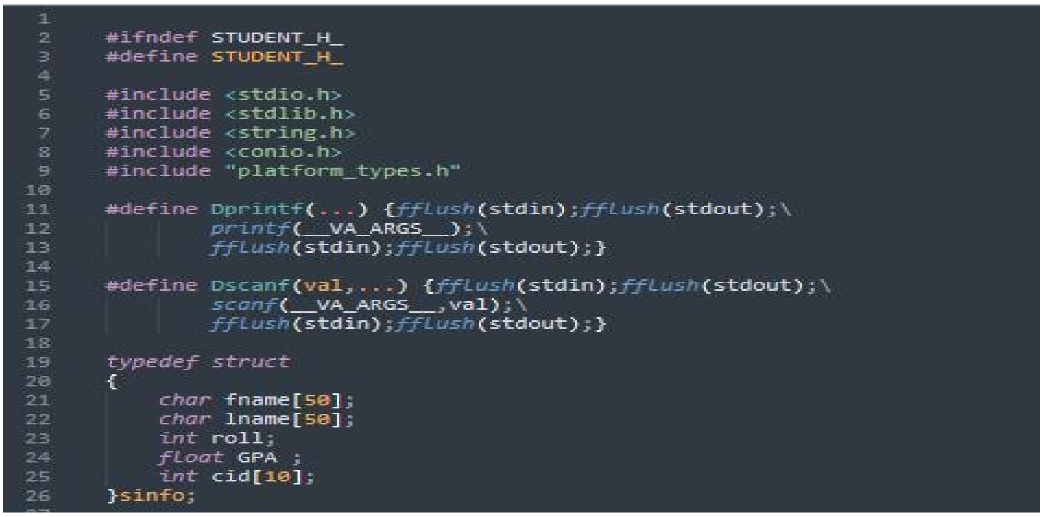


```
Welcome to the Student management System

******************************************
Choose the task that you want 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 Students Details by Roll Number
        8. Update the Students Details by Roll Number
        9. Show All Information
        10. To Exit

Enter your choice to perform the task :
```

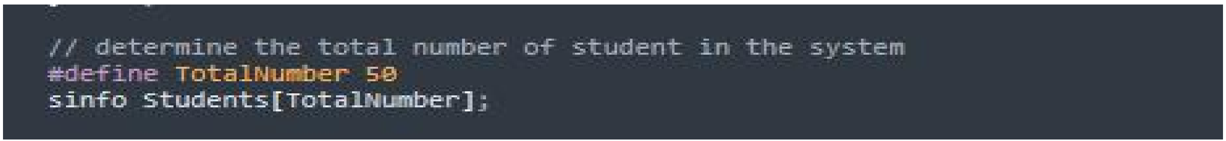
❖ Student.h Code :

The Student.h consist of:

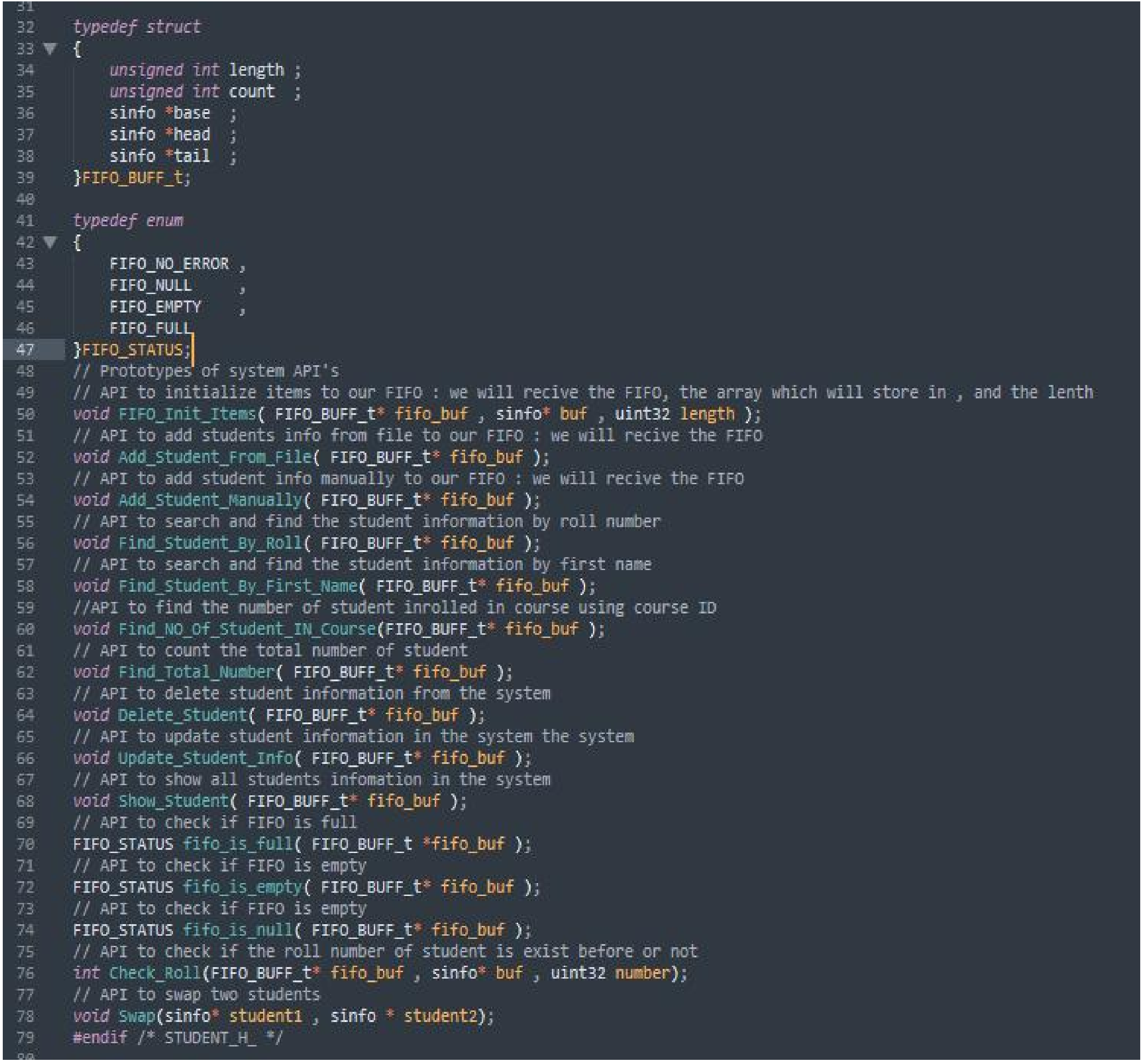
- Include all used libraries in projects
- Macro to manipulate the buffering problem in printf and scanf functions .
- Struct sinfo which contains the details of a student.
- Type definition of struct student info to use it easier with each student in the system.
- Type definition of FIFO Buffer.
- API prototypes with their functionality.



```c

#ifndef STUDENT_H_
#define STUDENT_H_

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <conio.h>
#include "platform_types.h"

#define Dprintf(...) {fflush(stdin);fflush(stdout);\
        printf(__VA_ARGS__);\
        fflush(stdin);fflush(stdout);}

#define Dscanf(val,...) {fflush(stdin);fflush(stdout);\
        scanf(__VA_ARGS__,val);\
        fflush(stdin);fflush(stdout);}

typedef struct
{
    char fname[50];
    char lname[50];
    int roll;
    float GPA ;
    int cid[10];
}sinfo;
```



```
// determine the total number of student in the system
#define TotalNumber 50
sinfo Students[TotalNumber];
```



```
typedef struct
{
    unsigned int length ;
    unsigned int count  ;
    sinfo *base  ;
    sinfo *head  ;
    sinfo *tail  ;
}FIFO_BUFF_t;

typedef enum
{
    FIFO_NO_ERROR ,
    FIFO_NULL     ,
    FIFO_EMPTY    ,
    FIFO_FULL
}FIFO_STATUS;
// Prototypes of system API's
// API to initialize items to our FIFO : we will recive the FIFO, the array which will store in , and the lenth
void FIFO_Init_Items( FIFO_BUFF_t* fifo_buf , sinfo* buf , uint32 length );
// API to add students info from file to our FIFO : we will recive the FIFO
void Add_Student_From_File( FIFO_BUFF_t* fifo_buf );
// API to add student info manually to our FIFO : we will recive the FIFO
void Add_Student_Manually( FIFO_BUFF_t* fifo_buf );
// API to search and find the student information by roll number
void Find_Student_By_Roll( FIFO_BUFF_t* fifo_buf );
// API to search and find the student information by first name
void Find_Student_By_First_Name( FIFO_BUFF_t* fifo_buf );
//API to find the number of student inrolled in course using course ID
void Find_NO_Of_Student_IN_Course(FIFO_BUFF_t* fifo_buf );
// API to count the total number of student
void Find_Total_Number( FIFO_BUFF_t* fifo_buf );
// API to delete student information from the system
void Delete_Student( FIFO_BUFF_t* fifo_buf );
// API to update student information in the system the system
void Update_Student_Info( FIFO_BUFF_t* fifo_buf );
// API to show all students infomation in the system
void Show_Student( FIFO_BUFF_t* fifo_buf );
// API to check if FIFO is full
FIFO_STATUS fifo_is_full( FIFO_BUFF_t *fifo_buf );
// API to check if FIFO is empty
FIFO_STATUS fifo_is_empty( FIFO_BUFF_t* fifo_buf );
// API to check if FIFO is empty
FIFO_STATUS fifo_is_null( FIFO_BUFF_t* fifo_buf );
// API to check if the roll number of student is exist before or not
int Check_Roll(FIFO_BUFF_t* fifo_buf , sinfo* buf , uint32 number);
// API to swap two students
void Swap(sinfo* student1 , sinfo * student2);
#endif /* STUDENT_H_ */
```

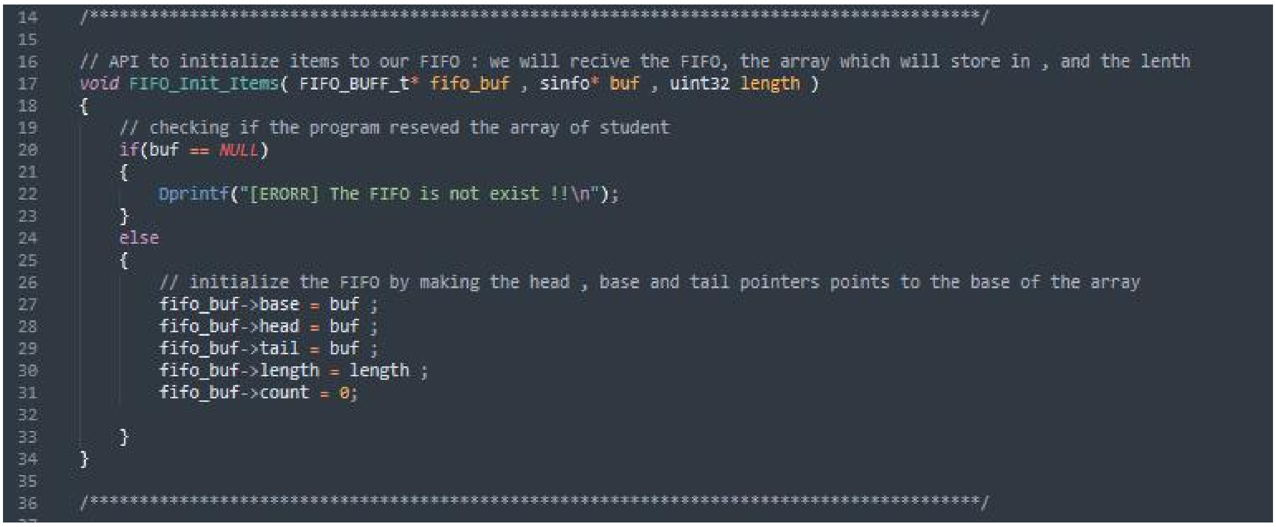
❖ Student.c :

he Student.c consist of :

- API Implementation

➢ FIFO initialization API :

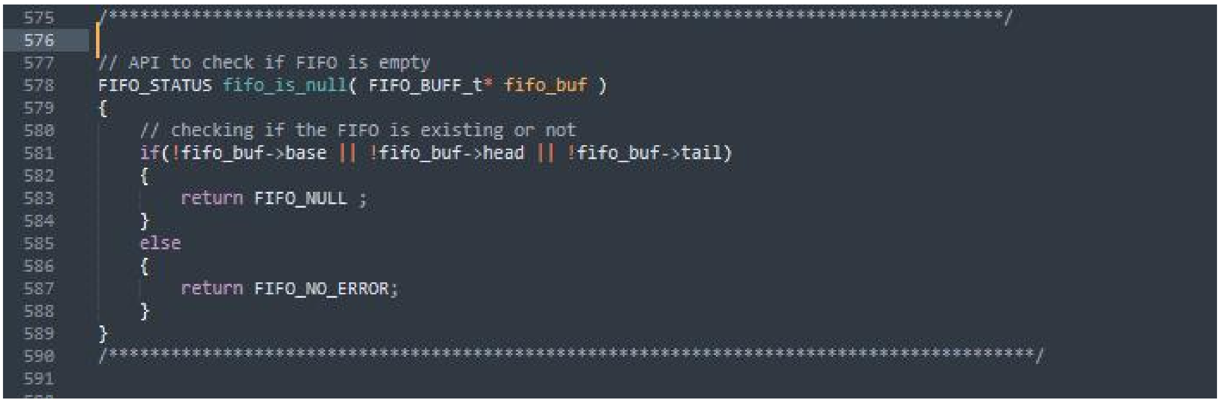
- Firstly , we check if the FIFO is null(exist).
- We start to link between the FIFO and the array.
- We make the head , base and tail pointers points to the base of the array .
- Set the length and counter .



```c
/**********************************************************************************************/

// API to initialize items to our FIFO : we will recive the FIFO, the array which will store in , and the lenth
void FIFO_Init_Items( FIFO_BUFF_t* fifo_buf , sinfo* buf , uint32 length )
{
    // checking if the program reseved the array of student
    if(buf == NULL)
    {
        Dprintf("[ERORR] The FIFO is not exist !!\n");
    }
    else
    {
        // initialize the FIFO by making the head , base and tail pointers points to the base of the array
        fifo_buf->base = buf ;
        fifo_buf->head = buf ;
        fifo_buf->tail = buf ;
        fifo_buf->length = length ;
        fifo_buf->count = 0;

    }
}

/**********************************************************************************************/
```

➢FIFO is null API :

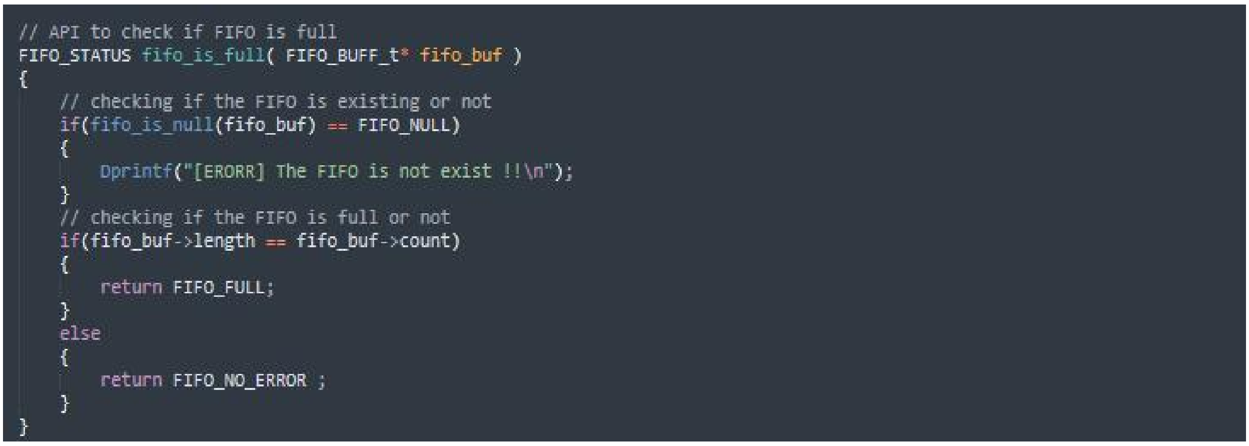
- Check if the FIFO is exist or not .
- Return the FIFO status.



```c
/**********************************************************************************************/

// API to check if FIFO is empty
FIFO_STATUS fifo_is_null( FIFO_BUFF_t* fifo_buf )
{
    // checking if the FIFO is existing or not
    if(!fifo_buf->base || !fifo_buf->head || !fifo_buf->tail)
    {
        return FIFO_NULL ;
    }
    else
    {
        return FIFO_NO_ERROR;
    }
}
/**********************************************************************************************/
```

➢FIFO is full API :

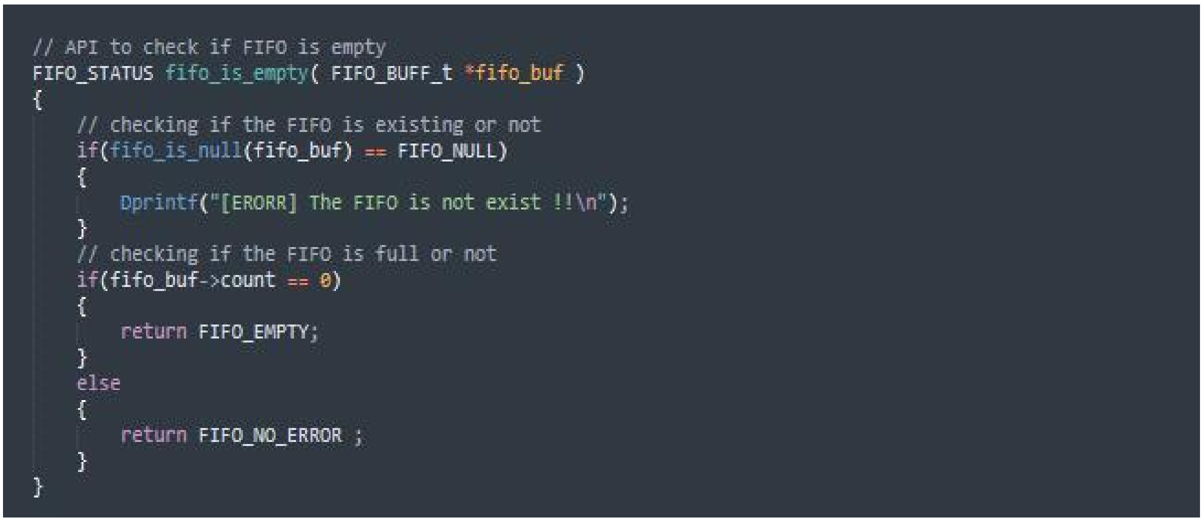
- Firstly , we call fifo_is_null function to check is FIFO null .
- Check if the FIFO is full or not.
- Return the FIFO status .



```
// API to check if FIFO is full
FIFO_STATUS fifo_is_full( FIFO_BUFF_t* fifo_buf )
{
    // checking if the FIFO is existing or not
    if(fifo_is_null(fifo_buf) == FIFO_NULL)
    {
        Dprintf("[ERORR] The FIFO is not exist !!\n");
    }
    // checking if the FIFO is full or not
    if(fifo_buf->length == fifo_buf->count)
    {
        return FIFO_FULL;
    }
    else
    {
        return FIFO_NO_ERROR ;
    }
}
```

➢FIFO is empty API :

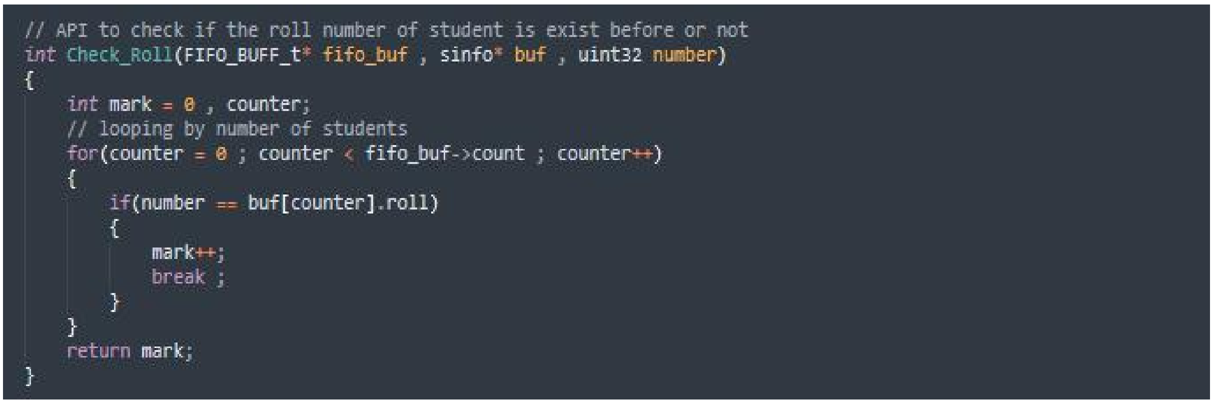
- Firstly , we call fifo_is_null function to check is FIFO null .
- Check if the FIFO is empty or not .
- Return the FIFO status.



```
// API to check if FIFO is empty
FIFO_STATUS fifo_is_empty( FIFO_BUFF_t *fifo_buf )
{
    // checking if the FIFO is existing or not
    if(fifo_is_null(fifo_buf) == FIFO_NULL)
    {
        Dprintf("[ERORR] The FIFO is not exist !!\n");
    }
    // checking if the FIFO is full or not
    if(fifo_buf->count == 0)
    {
        return FIFO_EMPTY;
    }
    else
    {
        return FIFO_NO_ERROR ;
    }
}
```

Check Roll API :

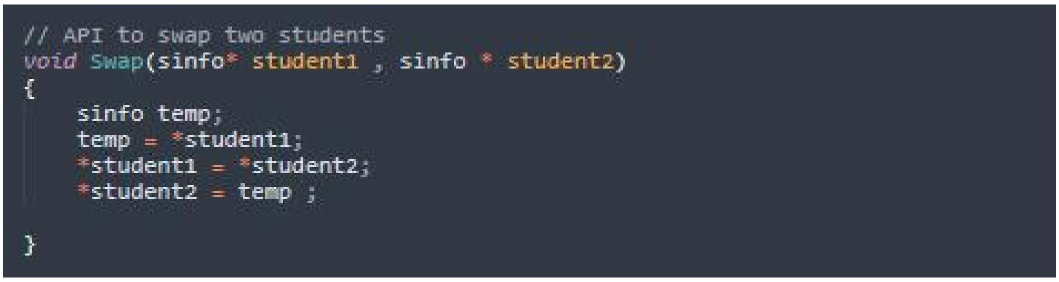
- Looping to all system student and check if roll of any student is identical to the given roll number.



```
// API to check if the roll number of student is exist before or not
int Check_Roll(FIFO_BUFF_t* fifo_buf , sinfo* buf , uint32 number)
{
    int mark = 0 , counter;
    // looping by number of students
    for(counter = 0 ; counter < fifo_buf->count ; counter++)
    {
        if(number == buf[counter].roll)
        {
            mark++;
            break ;
        }
    }
    return mark;
}
```

Swap Students API :

- Creating temp sinfo and then swap the two students locations .

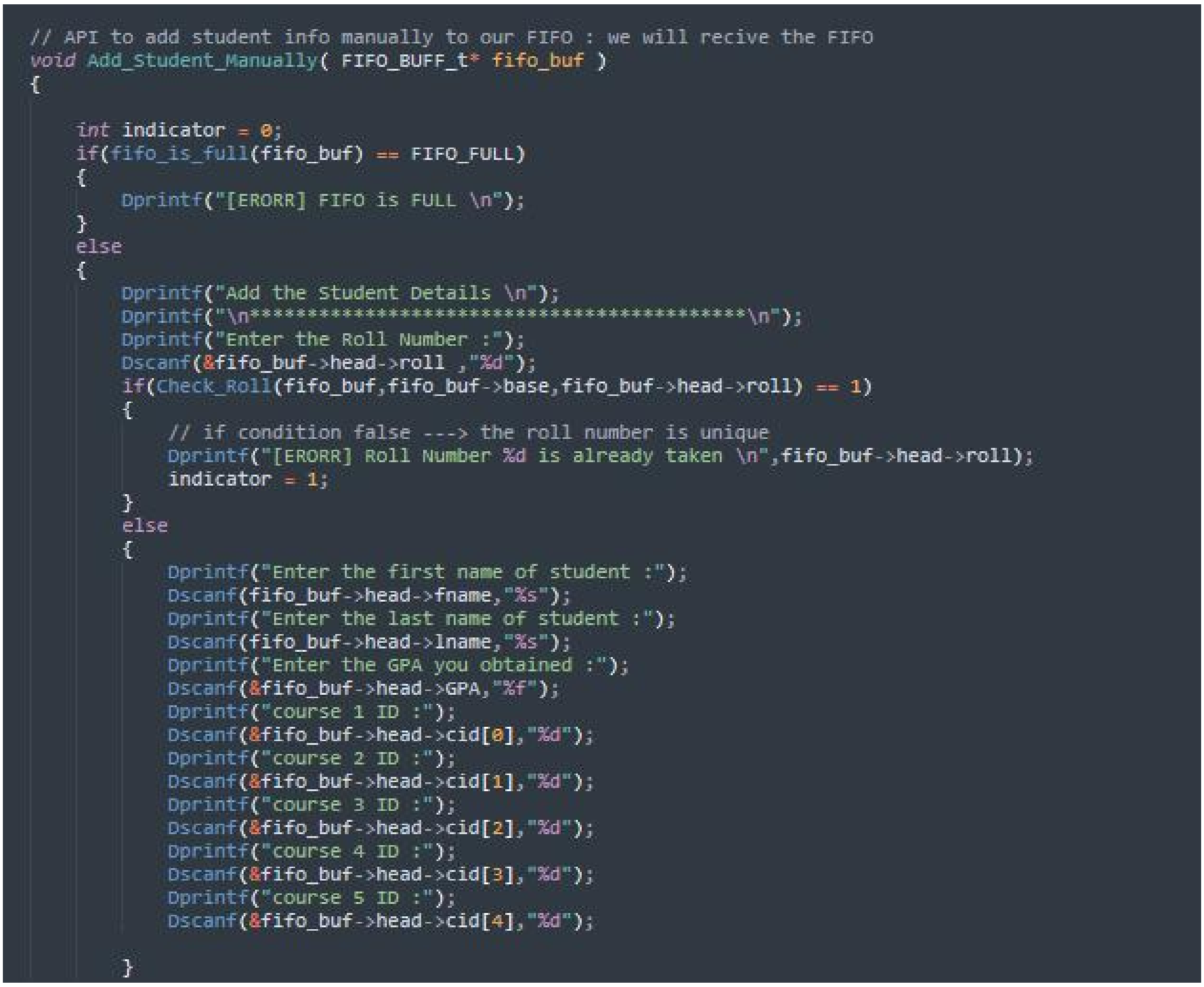


```
// API to swap two students
void Swap(sinfo* student1 , sinfo * student2)
{
    sinfo temp;
    temp = *student1;
    *student1 = *student2;
    *student2 = temp ;

}
```

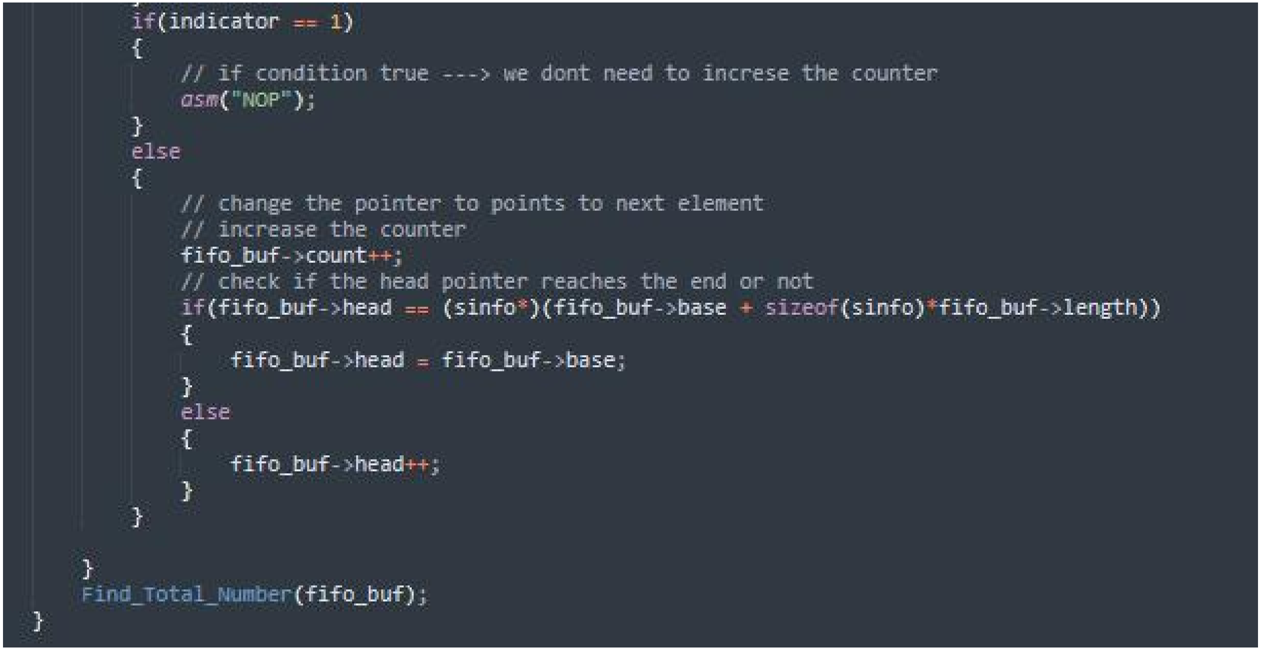
Add the Student Details Manually API :

- Firstly , we call fifo_is_null function to check is FIFO null .
- Call fifo_is_full function to check if the FIFO full
- Check if the roll number is token or not
- If roll number is token we terminate the process
- If roll number is unique we start to receive student info

API Code :

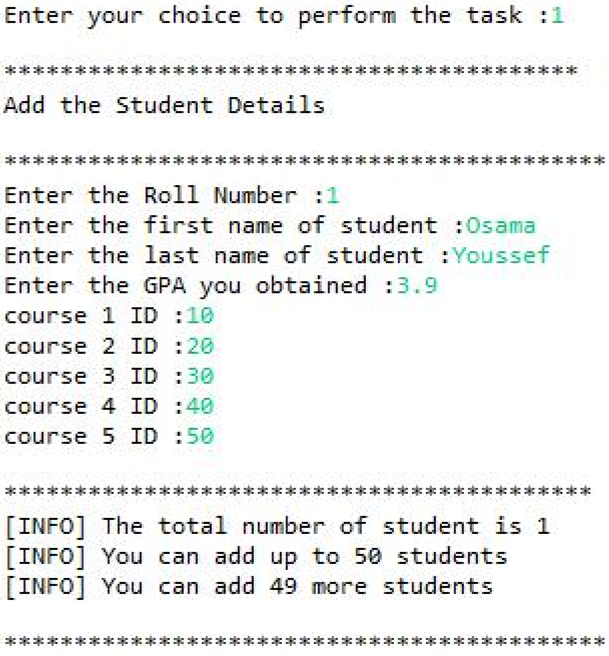


```c
// API to add student info manually to our FIFO : we will recive the FIFO
void Add_Student_Manually( FIFO_BUFF_t* fifo_buf )
{

    int indicator = 0;
    if(fifo_is_full(fifo_buf) == FIFO_FULL)
    {
        Dprintf("[ERORR] FIFO is FULL \n");
    }
    else
    {
        Dprintf("Add the Student Details \n");
        Dprintf("\n*******************************************\n");
        Dprintf("Enter the Roll Number :");
        Dscanf(&fifo_buf->head->roll ,"%d");
        if(Check_Roll(fifo_buf,fifo_buf->base,fifo_buf->head->roll) == 1)
        {
            // if condition false ---> the roll number is unique
            Dprintf("[ERORR] Roll Number %d is already taken \n",fifo_buf->head->roll);
            indicator = 1;
        }
        else
        {
            Dprintf("Enter the first name of student :");
            Dscanf(fifo_buf->head->fname,"%s");
            Dprintf("Enter the last name of student :");
            Dscanf(fifo_buf->head->lname,"%s");
            Dprintf("Enter the GPA you obtained :");
            Dscanf(&fifo_buf->head->GPA,"%f");
            Dprintf("course 1 ID :");
            Dscanf(&fifo_buf->head->cid[0],"%d");
            Dprintf("course 2 ID :");
            Dscanf(&fifo_buf->head->cid[1],"%d");
            Dprintf("course 3 ID :");
            Dscanf(&fifo_buf->head->cid[2],"%d");
            Dprintf("course 4 ID :");
            Dscanf(&fifo_buf->head->cid[3],"%d");
            Dprintf("course 5 ID :");
            Dscanf(&fifo_buf->head->cid[4],"%d");

        }
```



```
        if(indicator == 1)
        {
            // if condition true ---> we dont need to increse the counter
            asm("NOP");
        }
        else
        {
            // change the pointer to points to next element
            // increase the counter
            fifo_buf->count++;
            // check if the head pointer reaches the end or not
            if(fifo_buf->head == (sinfo*)(fifo_buf->base + sizeof(sinfo)*fifo_buf->length))
            {
                fifo_buf->head = fifo_buf->base;
            }
            else
            {
                fifo_buf->head++;
            }
        }

    }
    Find_Total_Number(fifo_buf);
}
```

- <u> API Test :</u>

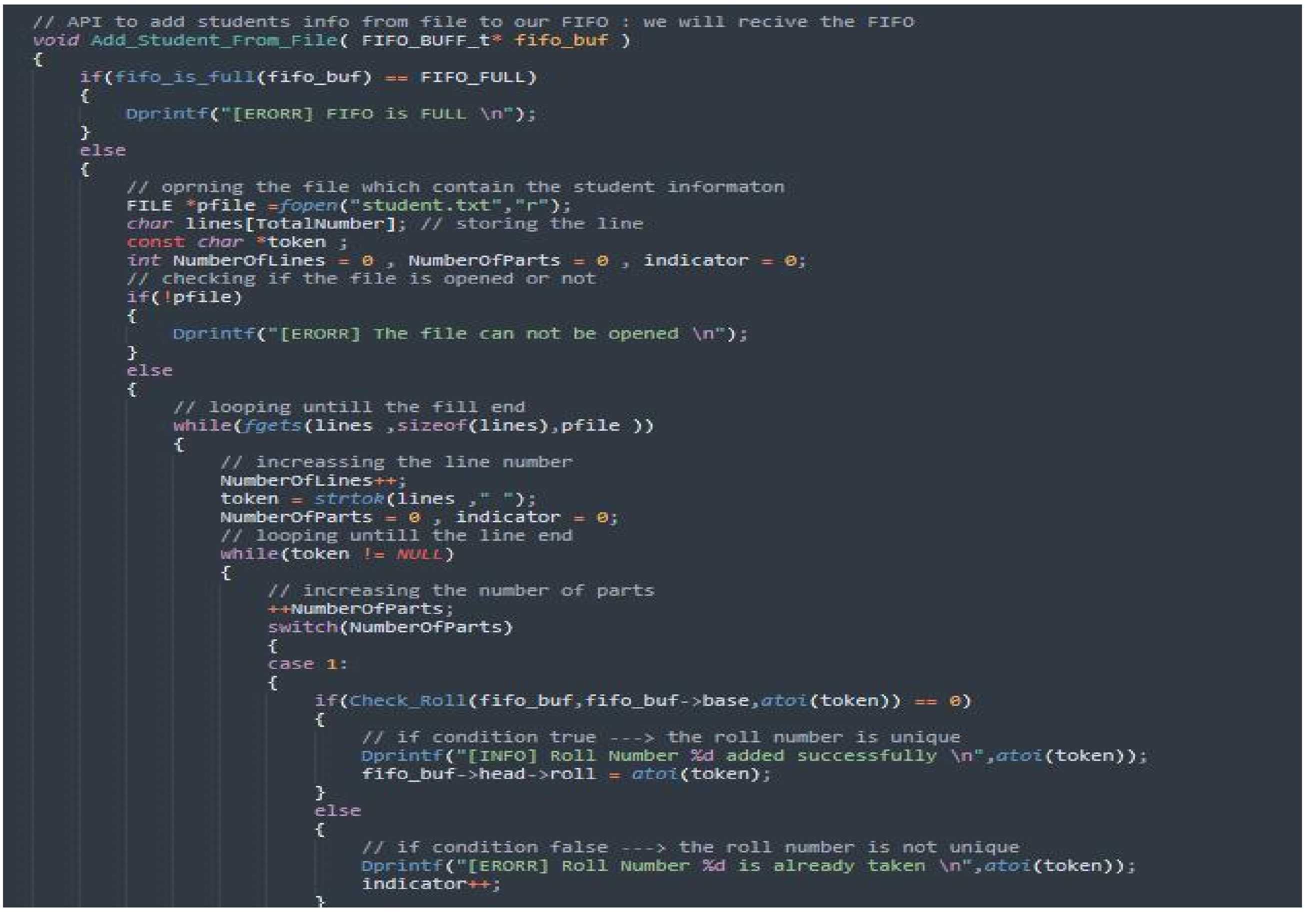


```
Enter your choice to perform the task :1

****************************************
Add the Student Details

******************************************
Enter the Roll Number :1
Enter the first name of student :Osama
Enter the last name of student :Youssef
Enter the GPA you obtained :3.9
course 1 ID :10
course 2 ID :20
course 3 ID :30
course 4 ID :40
course 5 ID :50

*****************************************
[INFO] The total number of student is 1
[INFO] You can add up to 50 students
[INFO] You can add 49 more students

******************************************
```

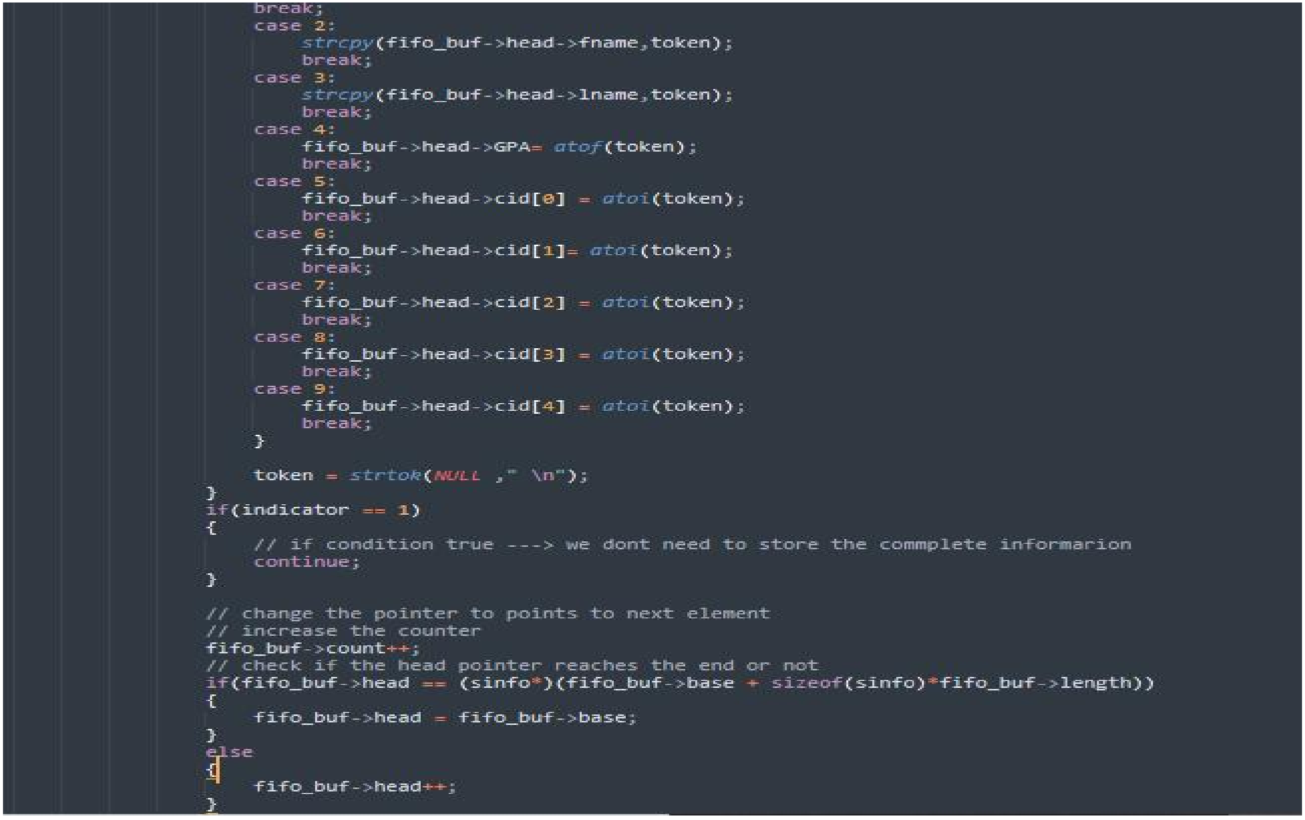
- <u>Add the Student Details Manually API:</u>
    - Firstly , we call fifo_is_null function to check is FIFO null .
    - Call fifo_is_full function to check if the FIFO full

- We create FILE pointer pfile to pointer to our file
- Start to read line by line and the tokan is an space
    - Read roll number
    - Read first name
    - Read last name
    - Read GPA
    - Read courses IDs
- Looping until reach the end of file
- Print if the the process executed successfully

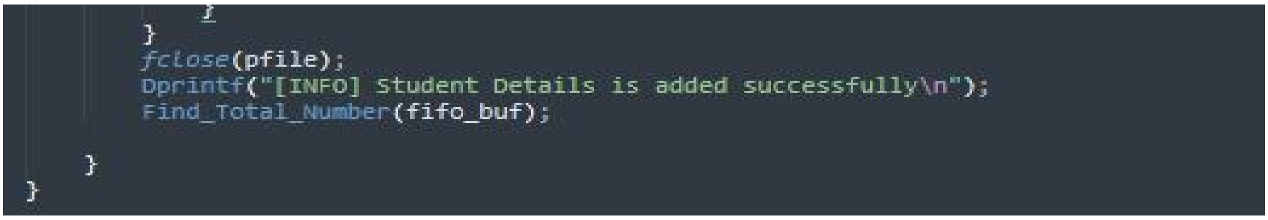
API Code :



```c
// API to add students info from file to our FIFO : we will recive the FIFO
void Add_Student_From_File( FIFO_BUFF_t* fifo_buf )
{
    if(fifo_is_full(fifo_buf) == FIFO_FULL)
    {
        Dprintf("[ERORR] FIFO is FULL \n");
    }
    else
    {
        // oprning the file which contain the student informaton
        FILE *pfile =fopen("student.txt","r");
        char lines[TotalNumber]; // storing the line
        const char *token ;
        int NumberOfLines = 0 , NumberOfParts = 0 , indicator = 0;
        // checking if the file is opened or not
        if(!pfile)
        {
            Dprintf("[ERORR] The file can not be opened \n");
        }
        else
        {
            // looping untill the fill end
            while(fgets(lines ,sizeof(lines),pfile ))
            {
                // increassing the line number
                NumberOfLines++;
                token = strtok(lines ," ");
                NumberOfParts = 0 , indicator = 0;
                // looping untill the line end
                while(token != NULL)
                {
                    // increasing the number of parts
                    ++NumberOfParts;
                    switch(NumberOfParts)
                    {
                    case 1:
                    {
                        if(Check_Roll(fifo_buf,fifo_buf->base,atoi(token)) == 0)
                        {
                            // if condition true ---> the roll number is unique
                            Dprintf("[INFO] Roll Number %d added successfully \n",atoi(token));
                            fifo_buf->head->roll = atoi(token);
                        }
                        else
                        {
                            // if condition false ---> the roll number is not unique
                            Dprintf("[ERORR] Roll Number %d is already taken \n",atoi(token));
                            indicator++;
                        }
```

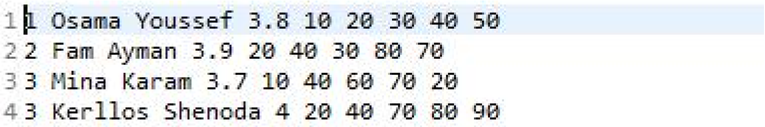


```
            break;
            case 2:
                strcpy(fifo_buf->head->fname,token);
                break;
            case 3:
                strcpy(fifo_buf->head->lname,token);
                break;
            case 4:
                fifo_buf->head->GPA= atof(token);
                break;
            case 5:
                fifo_buf->head->cid[0] = atoi(token);
                break;
            case 6:
                fifo_buf->head->cid[1]= atoi(token);
                break;
            case 7:
                fifo_buf->head->cid[2] = atoi(token);
                break;
            case 8:
                fifo_buf->head->cid[3] = atoi(token);
                break;
            case 9:
                fifo_buf->head->cid[4] = atoi(token);
                break;
            }

            token = strtok(NULL ," \n");
        }
        if(indicator == 1)
        {
            // if condition true ---> we dont need to store the commplete informarion
            continue;
        }

        // change the pointer to points to next element
        // increase the counter
        fifo_buf->count++;
        // check if the head pointer reaches the end or not
        if(fifo_buf->head == (sinfo*)(fifo_buf->base + sizeof(sinfo)*fifo_buf->length))
        {
            fifo_buf->head = fifo_buf->base;
        }
        else
        {
            fifo_buf->head++;
        }
```

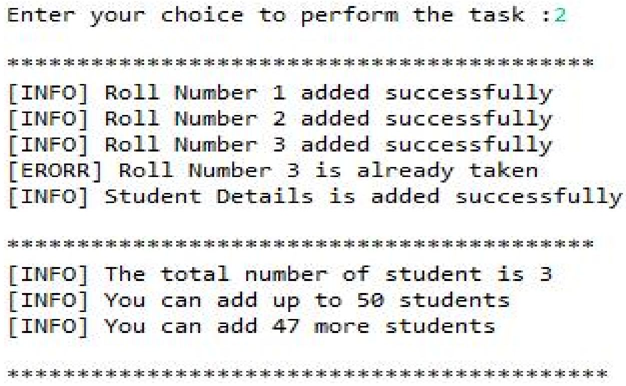


```
            }
        }
        fclose(pfile);
        Dprintf("[INFO] Student Details is added successfully\n");
        Find_Total_Number(fifo_buf);

    }
}
```

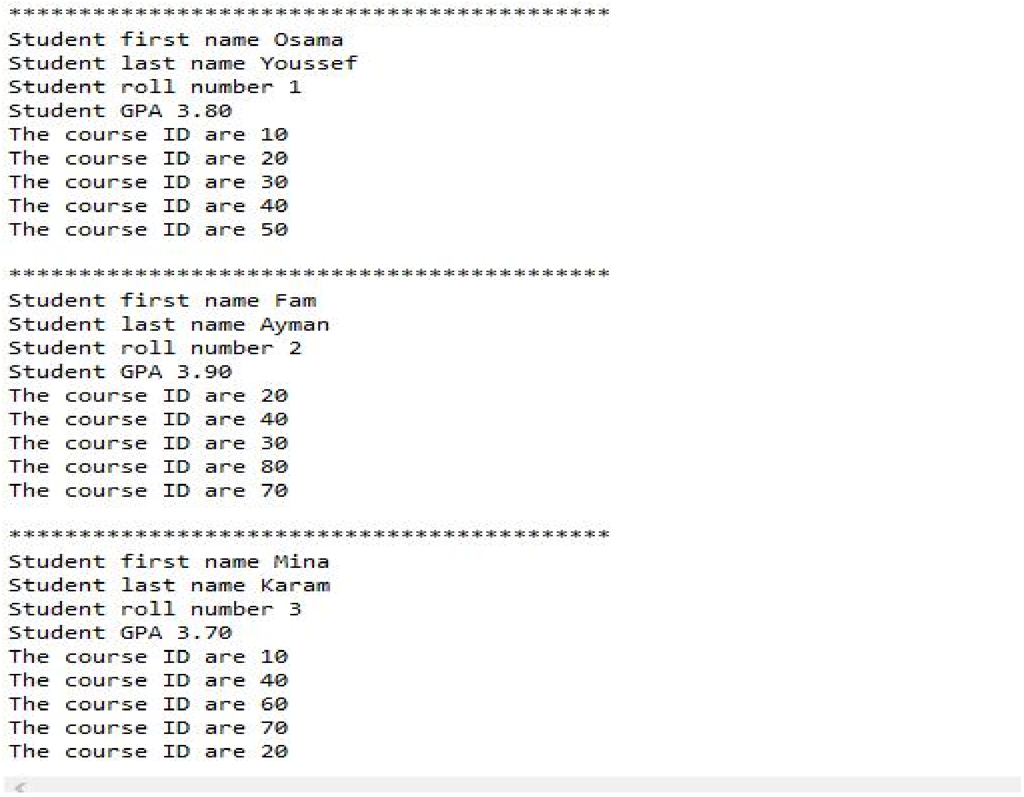
➢ API Test :



```
1 Osama Youssef 3.8 10 20 30 40 50
2 Fam Ayman 3.9 20 40 30 80 70
3 Mina Karam 3.7 10 40 60 70 20
3 Kerllos Shenoda 4 20 40 70 80 90
```



```
Enter your choice to perform the task :2

******************************************
[INFO] Roll Number 1 added successfully
[INFO] Roll Number 2 added successfully
[INFO] Roll Number 3 added successfully
[ERORR] Roll Number 3 is already taken
[INFO] Student Details is added successfully

******************************************
[INFO] The total number of student is 3
[INFO] You can add up to 50 students
[INFO] You can add 47 more students

*******************************************
```

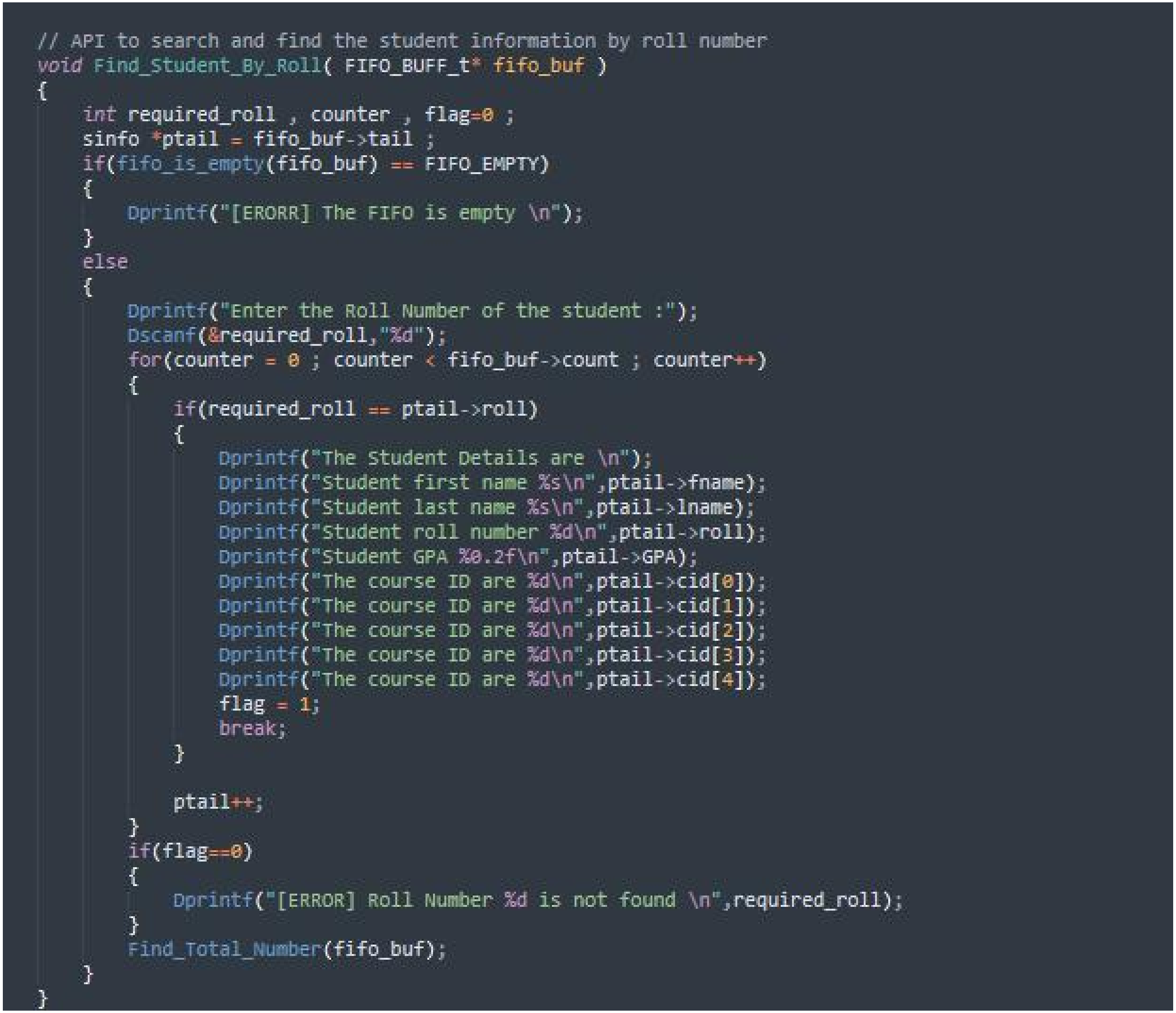


```
*******************************************
Student first name Osama
Student last name Youssef
Student roll number 1
Student GPA 3.80
The course ID are 10
The course ID are 20
The course ID are 30
The course ID are 40
The course ID are 50

*******************************************
Student first name Fam
Student last name Ayman
Student roll number 2
Student GPA 3.90
The course ID are 20
The course ID are 40
The course ID are 30
The course ID are 80
The course ID are 70

*******************************************
Student first name Mina
Student last name Karam
Student roll number 3
Student GPA 3.70
The course ID are 10
The course ID are 40
The course ID are 60
The course ID are 70
The course ID are 20
```

➢ Find the Student Details by Roll Number API:

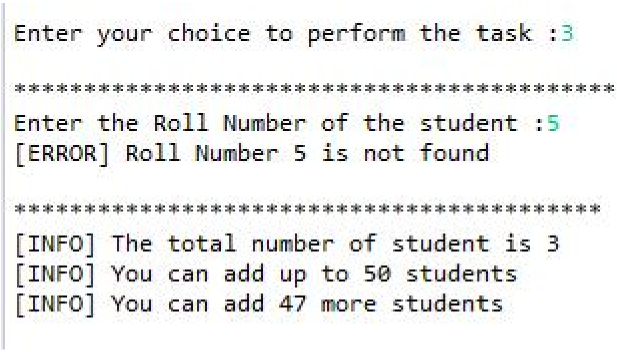
- Firstly , we call fifo_is_null function to check is FIFO null .
- Call fifo_is_empty function to check if the FIFO empty
- Ask user to enter the roll number
- If the user entered roll number is not found message the user by calling check_roll function
- If found , show informations of the student

➢ API Code :

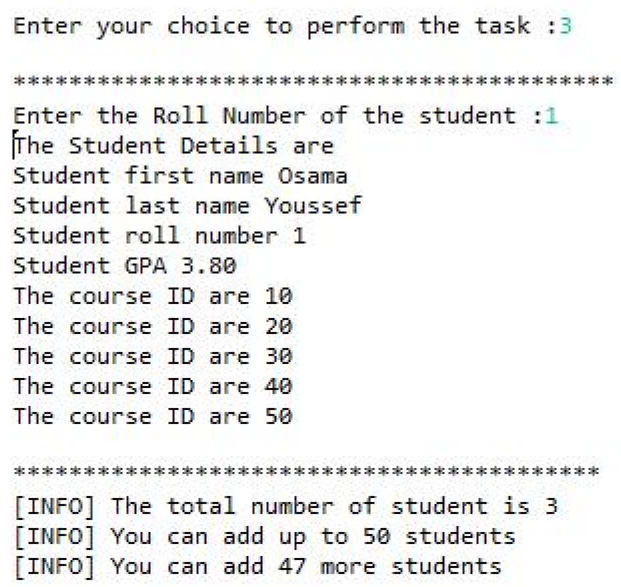


```c
// API to search and find the student information by roll number
void Find_Student_By_Roll( FIFO_BUFF_t* fifo_buf )
{
    int required_roll , counter , flag=0 ;
    sinfo *ptail = fifo_buf->tail ;
    if(fifo_is_empty(fifo_buf) == FIFO_EMPTY)
    {
        Dprintf("[ERORR] The FIFO is empty \n");
    }
    else
    {
        Dprintf("Enter the Roll Number of the student :");
        Dscanf(&required_roll,"%d");
        for(counter = 0 ; counter < fifo_buf->count ; counter++)
        {
            if(required_roll == ptail->roll)
            {
                Dprintf("The Student Details are \n");
                Dprintf("Student first name %s\n",ptail->fname);
                Dprintf("Student last name %s\n",ptail->lname);
                Dprintf("Student roll number %d\n",ptail->roll);
                Dprintf("Student GPA %0.2f\n",ptail->GPA);
                Dprintf("The course ID are %d\n",ptail->cid[0]);
                Dprintf("The course ID are %d\n",ptail->cid[1]);
                Dprintf("The course ID are %d\n",ptail->cid[2]);
                Dprintf("The course ID are %d\n",ptail->cid[3]);
                Dprintf("The course ID are %d\n",ptail->cid[4]);
                flag = 1;
                break;
            }

            ptail++;
        }
        if(flag==0)
        {
            Dprintf("[ERROR] Roll Number %d is not found \n",required_roll);
        }
        Find_Total_Number(fifo_buf);
    }
}
```

➢ API Test :



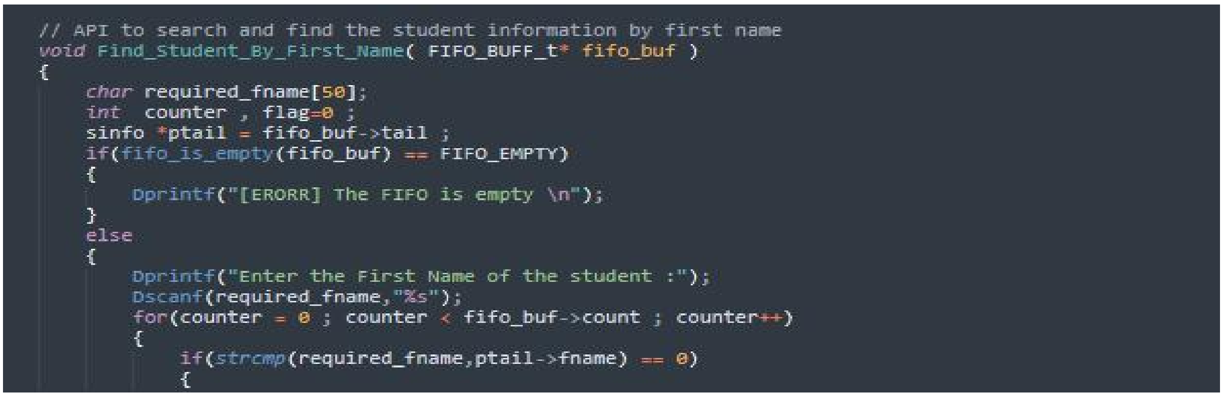
```
Enter your choice to perform the task :3

********************************************
Enter the Roll Number of the student :5
[ERROR] Roll Number 5 is not found

*******************************************
[INFO] The total number of student is 3
[INFO] You can add up to 50 students
[INFO] You can add 47 more students
```



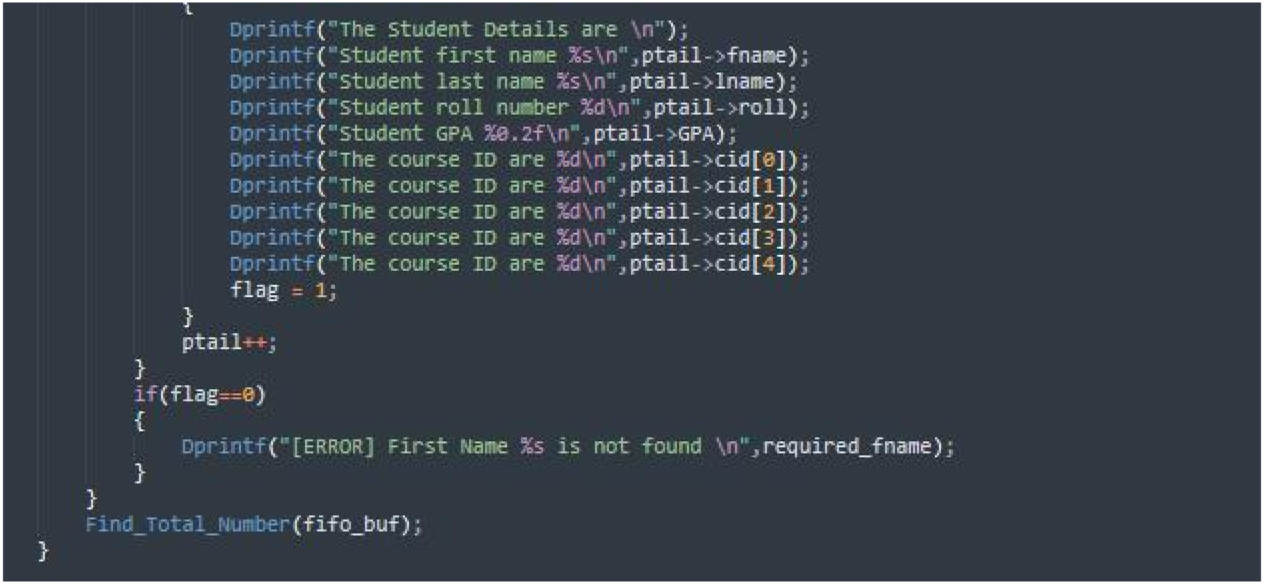
```
Enter your choice to perform the task :3

*******************************************
Enter the Roll Number of the student :1
The Student Details are
Student first name Osama
Student last name Youssef
Student roll number 1
Student GPA 3.80
The course ID are 10
The course ID are 20
The course ID are 30
The course ID are 40
The course ID are 50

******************************************
[INFO] The total number of student is 3
[INFO] You can add up to 50 students
[INFO] You can add 47 more students
```

➢ Find the Student Details by First Name API:

- Firstly , we call fifo_is_null function to check is FIFO null .
- Call fifo_is_empty function to check if the FIFO empty
- Ask user to enter the first name
- If found , show informations of the student

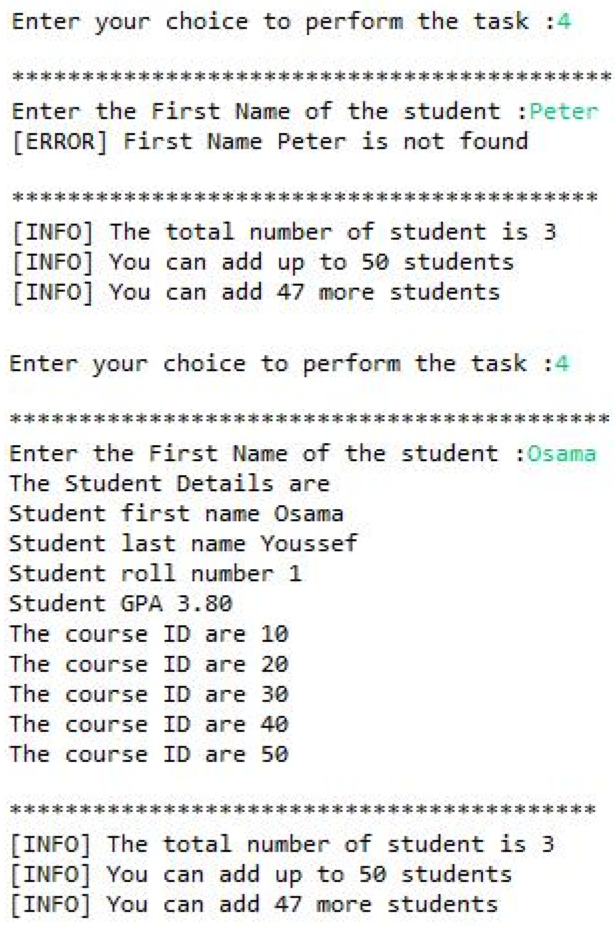


```c
// API to search and find the student information by first name
void Find_Student_By_First_Name( FIFO_BUFF_t* fifo_buf )
{
    char required_fname[50];
    int  counter , flag=0 ;
    sinfo *ptail = fifo_buf->tail ;
    if(fifo_is_empty(fifo_buf) == FIFO_EMPTY)
    {
        Dprintf("[ERORR] The FIFO is empty \n");
    }
    else
    {
        Dprintf("Enter the First Name of the student :");
        Dscanf(required_fname,"%s");
        for(counter = 0 ; counter < fifo_buf->count ; counter++)
        {
            if(strcmp(required_fname,ptail->fname) == 0)
            {
```



```
            {
                Dprintf("The Student Details are \n");
                Dprintf("Student first name %s\n",ptail->fname);
                Dprintf("Student last name %s\n",ptail->lname);
                Dprintf("Student roll number %d\n",ptail->roll);
                Dprintf("Student GPA %0.2f\n",ptail->GPA);
                Dprintf("The course ID are %d\n",ptail->cid[0]);
                Dprintf("The course ID are %d\n",ptail->cid[1]);
                Dprintf("The course ID are %d\n",ptail->cid[2]);
                Dprintf("The course ID are %d\n",ptail->cid[3]);
                Dprintf("The course ID are %d\n",ptail->cid[4]);
                flag = 1;
            }
            ptail++;
        }
        if(flag==0)
        {
            Dprintf("[ERROR] First Name %s is not found \n",required_fname);
        }
    }
    Find_Total_Number(fifo_buf);
}
```

➢ API Test :

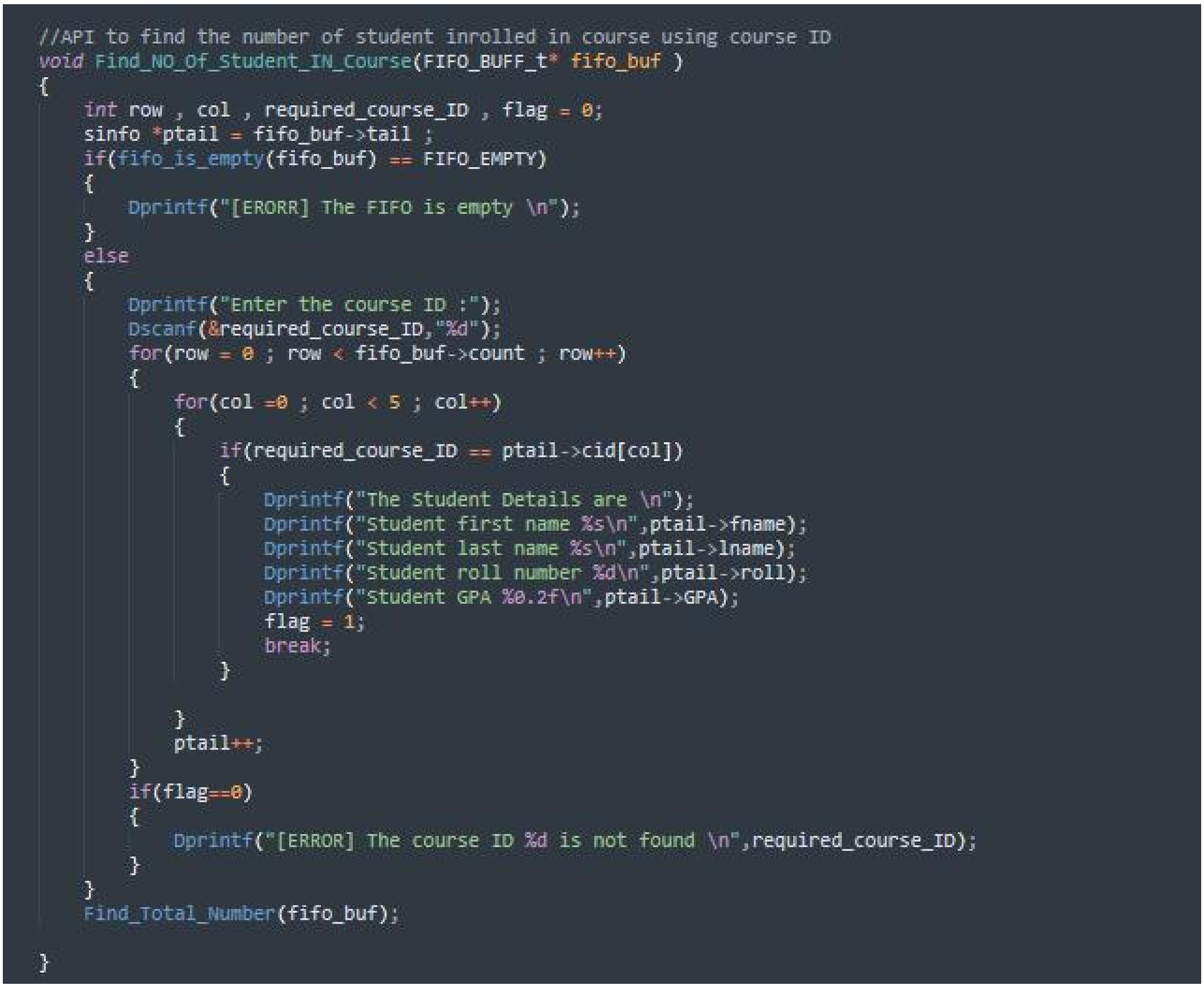


```
Enter your choice to perform the task :4

*******************************************
Enter the First Name of the student :Peter
[ERROR] First Name Peter is not found

*****************************************
[INFO] The total number of student is 3
[INFO] You can add up to 50 students
[INFO] You can add 47 more students


Enter your choice to perform the task :4

*******************************************
Enter the First Name of the student :Osama
The Student Details are
Student first name Osama
Student last name Youssef
Student roll number 1
Student GPA 3.80
The course ID are 10
The course ID are 20
The course ID are 30
The course ID are 40
The course ID are 50

*****************************************
[INFO] The total number of student is 3
[INFO] You can add up to 50 students
[INFO] You can add 47 more students
```

Find the Student Details by Course ID API:

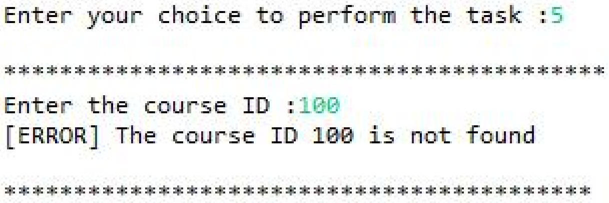
- Firstly , we call fifo_is_null function to check is FIFO null .
- Call fifo_is_empty function to check if the FIFO empty.
- Ask user to enter the course ID.
- If found , show informations of the student.

API Code :

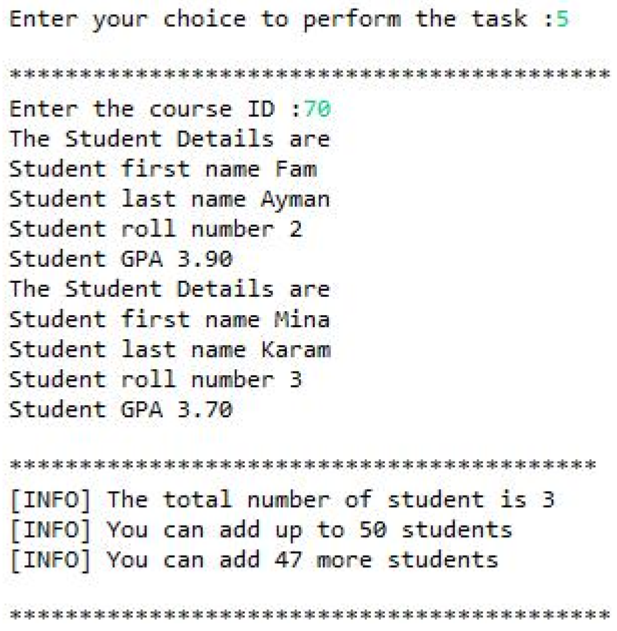


```c
//API to find the number of student inrolled in course using course ID
void Find_NO_Of_Student_IN_Course(FIFO_BUFF_t* fifo_buf )
{
    int row , col , required_course_ID , flag = 0;
    sinfo *ptail = fifo_buf->tail ;
    if(fifo_is_empty(fifo_buf) == FIFO_EMPTY)
    {
        Dprintf("[ERORR] The FIFO is empty \n");
    }
    else
    {
        Dprintf("Enter the course ID :");
        Dscanf(&required_course_ID,"%d");
        for(row = 0 ; row < fifo_buf->count ; row++)
        {
            for(col =0 ; col < 5 ; col++)
            {
                if(required_course_ID == ptail->cid[col])
                {
                    Dprintf("The Student Details are \n");
                    Dprintf("Student first name %s\n",ptail->fname);
                    Dprintf("Student last name %s\n",ptail->lname);
                    Dprintf("Student roll number %d\n",ptail->roll);
                    Dprintf("Student GPA %0.2f\n",ptail->GPA);
                    flag = 1;
                    break;
                }

            }
            ptail++;
        }
        if(flag==0)
        {
            Dprintf("[ERROR] The course ID %d is not found \n",required_course_ID);
        }
    }
    Find_Total_Number(fifo_buf);

}
```

API Test :



```
Enter your choice to perform the task :5

*******************************************
Enter the course ID :100
[ERROR] The course ID 100 is not found

*******************************************
```

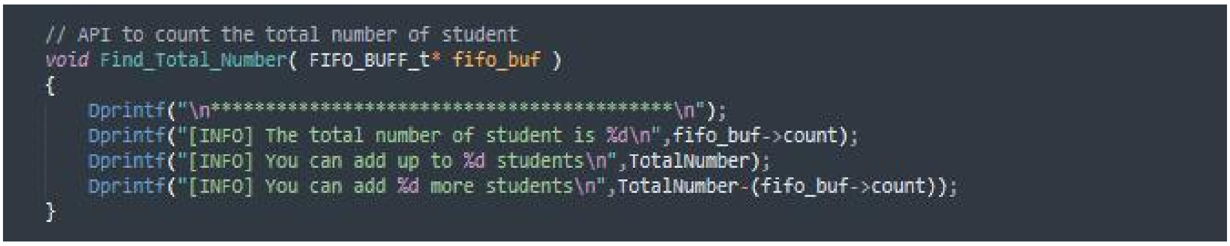


```
Enter your choice to perform the task :5

*******************************************
Enter the course ID :70
The Student Details are
Student first name Fam
Student last name Ayman
Student roll number 2
Student GPA 3.90
The Student Details are
Student first name Mina
Student last name Karam
Student roll number 3
Student GPA 3.70

******************************************
[INFO] The total number of student is 3
[INFO] You can add up to 50 students
[INFO] You can add 47 more students

*******************************************
```

➢ Find the Total Number Of Students API:

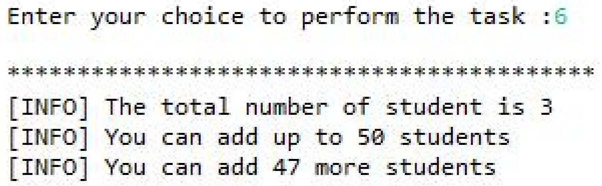
- Firstly , we call fifo_is_null function to check is FIFO null .
- Call fifo_is_empty function to check if the FIFO empty.
- Start to show number of student.

➢ API Code :



```
// API to count the total number of student
void Find_Total_Number( FIFO_BUFF_t* fifo_buf )
{
    Dprintf("\n******************************************\n");
    Dprintf("[INFO] The total number of student is %d\n",fifo_buf->count);
    Dprintf("[INFO] You can add up to %d students\n",TotalNumber);
    Dprintf("[INFO] You can add %d more students\n",TotalNumber-(fifo_buf->count));
}
```

➢ API Test :

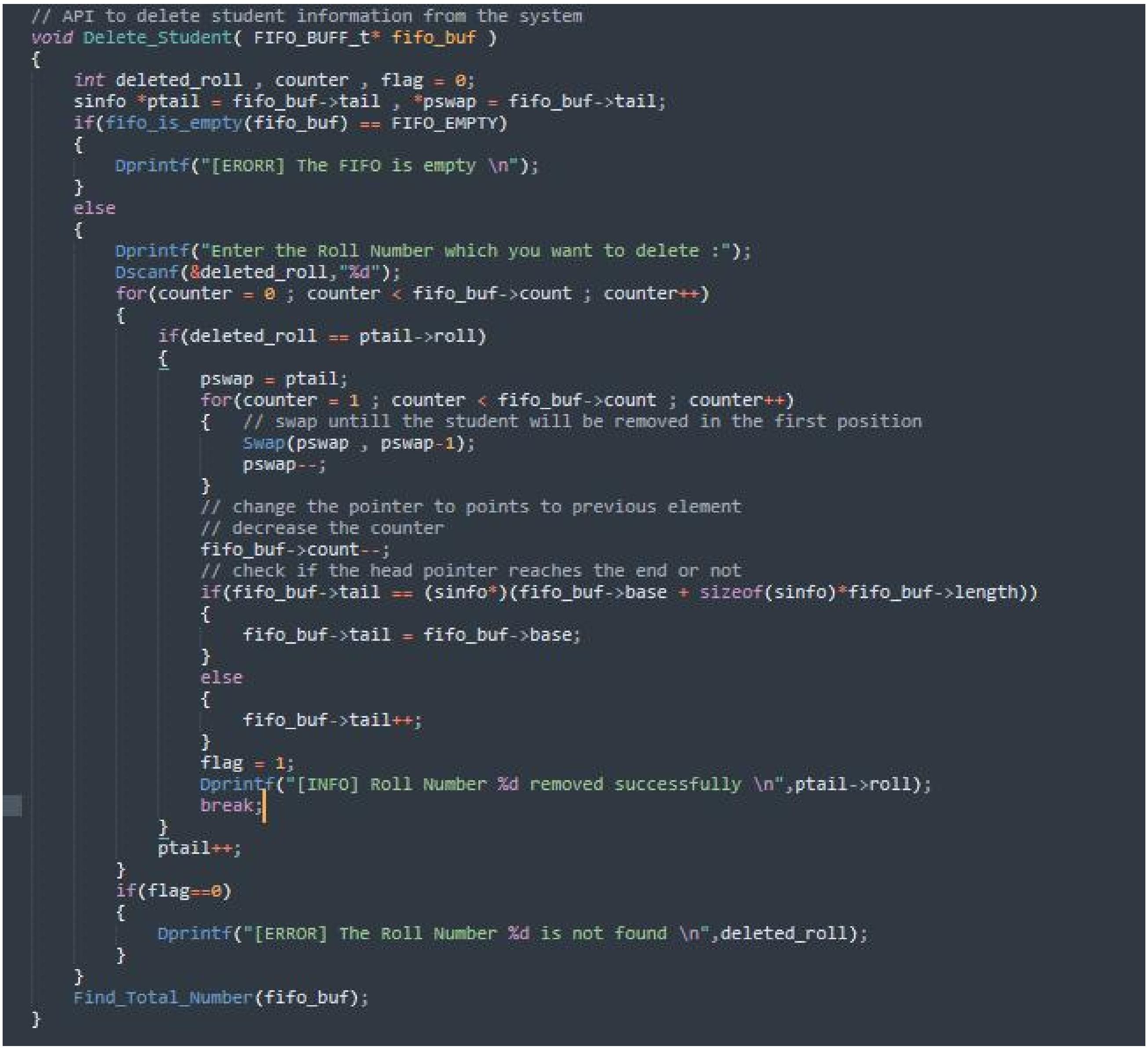


```
Enter your choice to perform the task :6

******************************************
[INFO] The total number of student is 3
[INFO] You can add up to 50 students
[INFO] You can add 47 more students
```

➢ Delete the Students Details by Roll Number API:

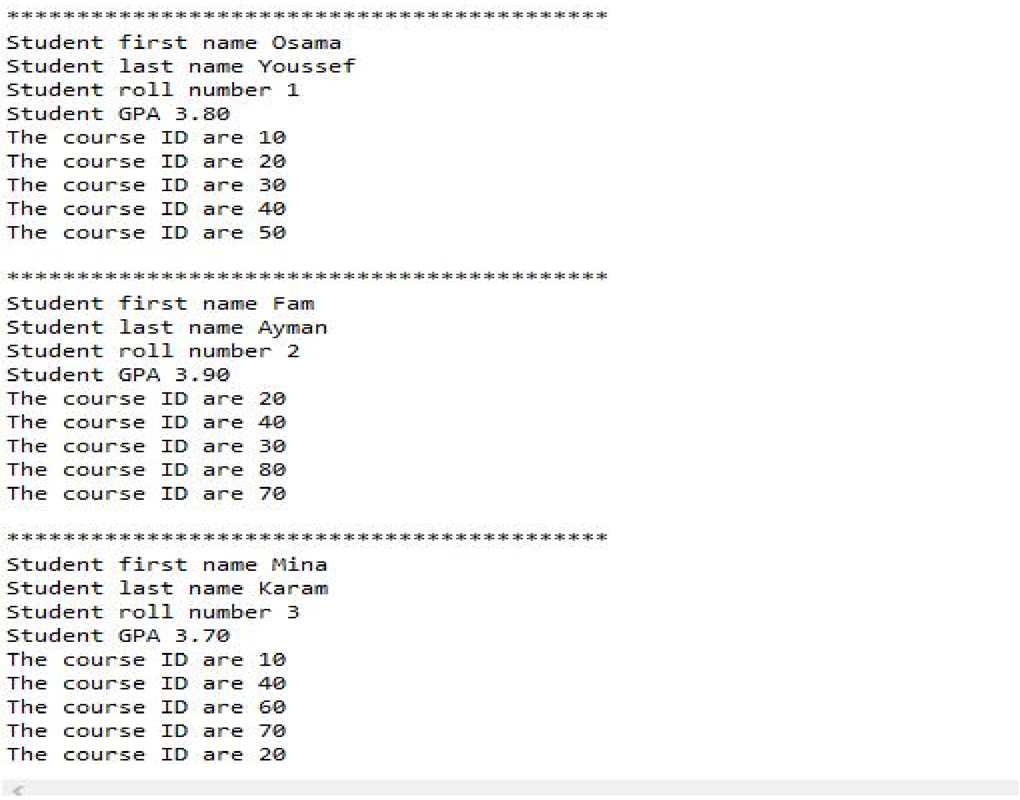
- Firstly , we call fifo_is_null function to check is FIFO null .
- Call fifo_is_empty function to check if the FIFO empty.
- Ask user to enter the roll number which will be deleted.
- If the user entered roll number is not found message the user.
- If found , start to call swap function to set the deleted student in start of enqueued student and delete him.
- Message the user that the deletion is executed successfully.

➢ API Code :

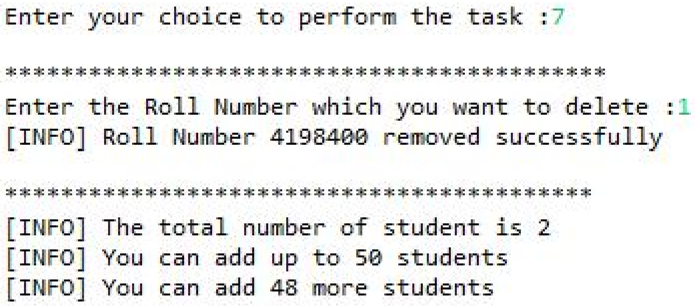


```c
// API to delete student information from the system
void Delete_Student( FIFO_BUFF_t* fifo_buf )
{
    int deleted_roll , counter , flag = 0;
    sinfo *ptail = fifo_buf->tail , *pswap = fifo_buf->tail;
    if(fifo_is_empty(fifo_buf) == FIFO_EMPTY)
    {
        Dprintf("[ERORR] The FIFO is empty \n");
    }
    else
    {
        Dprintf("Enter the Roll Number which you want to delete :");
        Dscanf(&deleted_roll,"%d");
        for(counter = 0 ; counter < fifo_buf->count ; counter++)
        {
            if(deleted_roll == ptail->roll)
            {
                pswap = ptail;
                for(counter = 1 ; counter < fifo_buf->count ; counter++)
                {   // swap untill the student will be removed in the first position
                    Swap(pswap , pswap-1);
                    pswap--;
                }
                // change the pointer to points to previous element
                // decrease the counter
                fifo_buf->count--;
                // check if the head pointer reaches the end or not
                if(fifo_buf->tail == (sinfo*)(fifo_buf->base + sizeof(sinfo)*fifo_buf->length))
                {
                    fifo_buf->tail = fifo_buf->base;
                }
                else
                {
                    fifo_buf->tail++;
                }
                flag = 1;
                Dprintf("[INFO] Roll Number %d removed successfully \n",ptail->roll);
                break;
            }
            ptail++;
        }
        if(flag==0)
        {
            Dprintf("[ERROR] The Roll Number %d is not found \n",deleted_roll);
        }
    }
    Find_Total_Number(fifo_buf);
}
```

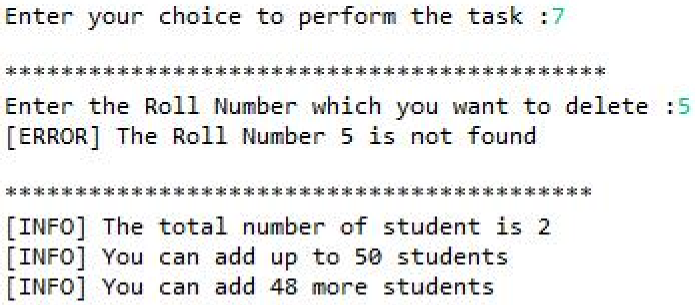
➢ API Test :



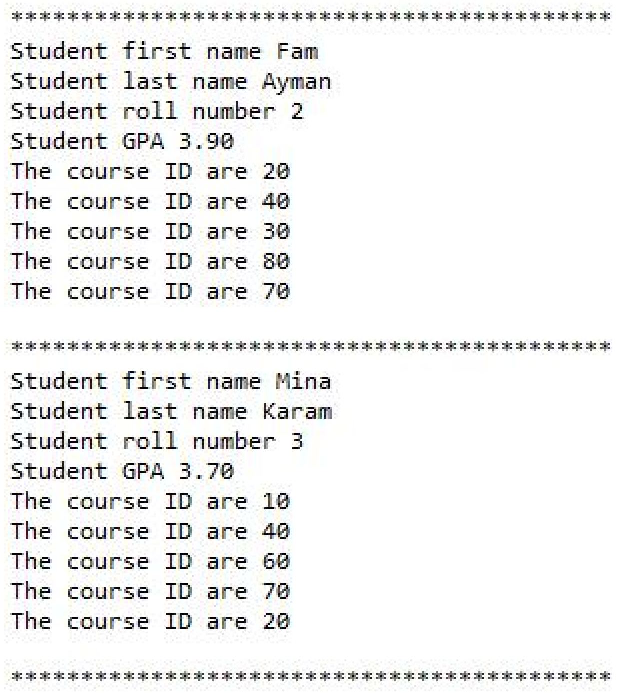
```
*******************************************
Student first name Osama
Student last name Youssef
Student roll number 1
Student GPA 3.80
The course ID are 10
The course ID are 20
The course ID are 30
The course ID are 40
The course ID are 50

*******************************************
Student first name Fam
Student last name Ayman
Student roll number 2
Student GPA 3.90
The course ID are 20
The course ID are 40
The course ID are 30
The course ID are 80
The course ID are 70

*******************************************
Student first name Mina
Student last name Karam
Student roll number 3
Student GPA 3.70
The course ID are 10
The course ID are 40
The course ID are 60
The course ID are 70
The course ID are 20
```



```
Enter your choice to perform the task :7

*******************************************
Enter the Roll Number which you want to delete :1
[INFO] Roll Number 4198400 removed successfully

******************************************
[INFO] The total number of student is 2
[INFO] You can add up to 50 students
[INFO] You can add 48 more students
```



```
Enter your choice to perform the task :7

*******************************************
Enter the Roll Number which you want to delete :5
[ERROR] The Roll Number 5 is not found

******************************************
[INFO] The total number of student is 2
[INFO] You can add up to 50 students
[INFO] You can add 48 more students
```

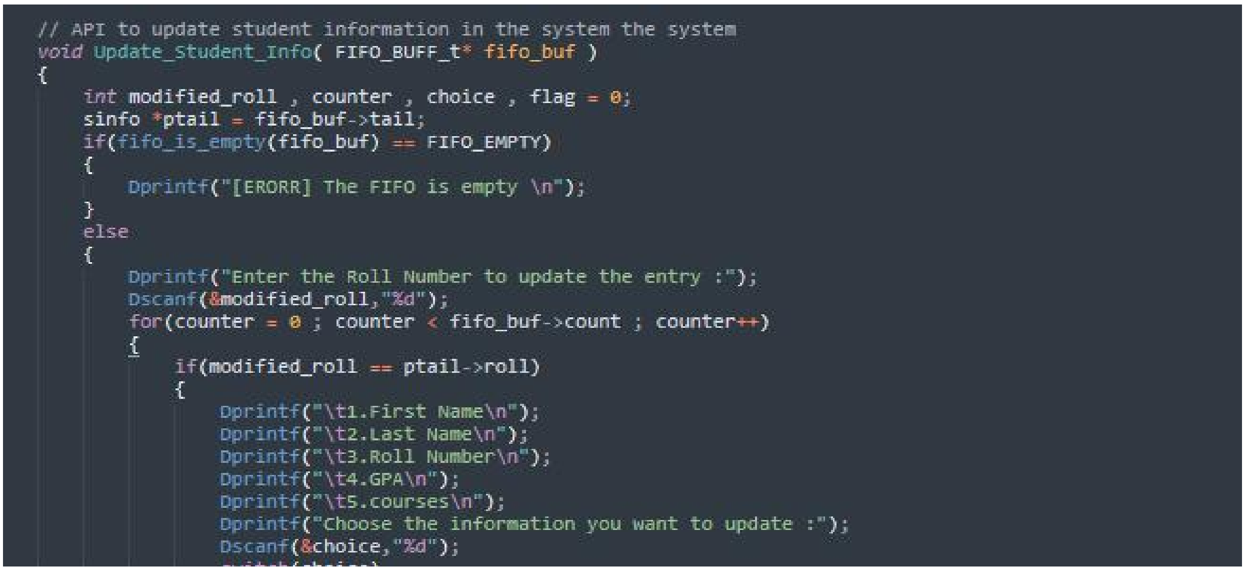


```
*******************************************
Student first name Fam
Student last name Ayman
Student roll number 2
Student GPA 3.90
The course ID are 20
The course ID are 40
The course ID are 30
The course ID are 80
The course ID are 70

*******************************************
Student first name Mina
Student last name Karam
Student roll number 3
Student GPA 3.70
The course ID are 10
The course ID are 40
The course ID are 60
The course ID are 70
The course ID are 20

*******************************************
```

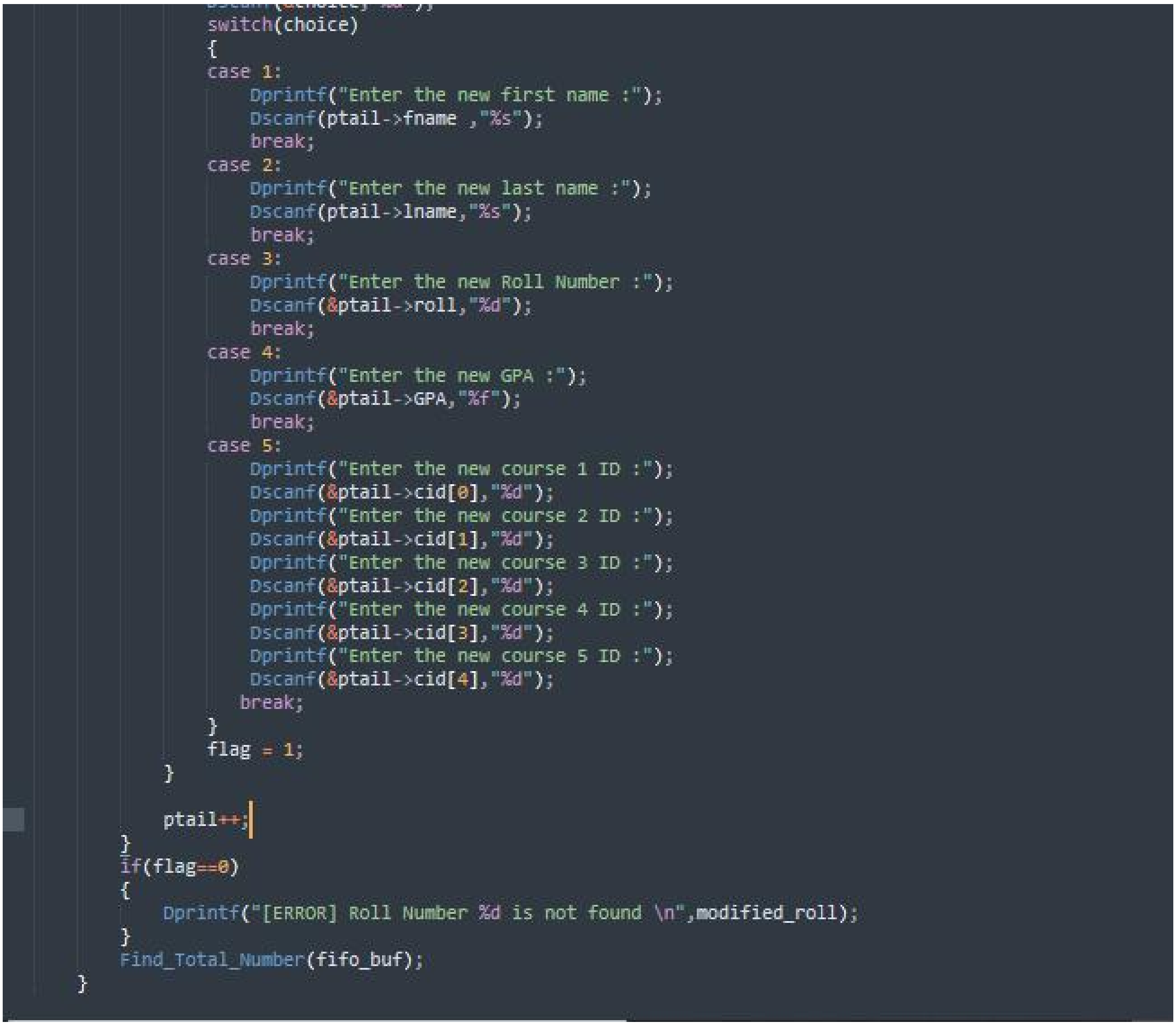
➢ Update Student Information by Roll Number API:

- Firstly , we call fifo_is_null function to check is FIFO null .
- Call fifo_is_empty function to check if the FIFO empty
- Ask user to enter the roll number which will be deleted
- If the user entered roll number is not found message the user.
- Start to switch in option of user about any information will be updated.
- Ask the user to enter the new data

➢ API Code :

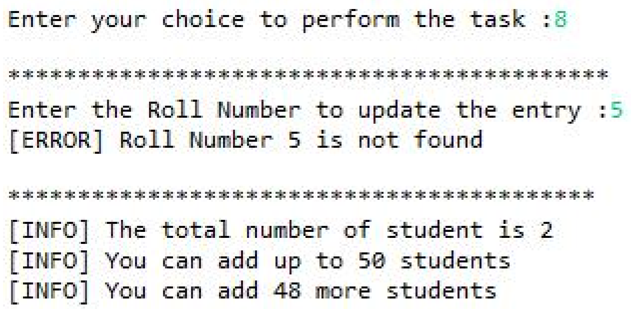


```
// API to update student information in the system the system
void Update_Student_Info( FIFO_BUFF_t* fifo_buf )
{
    int modified_roll , counter , choice , flag = 0;
    sinfo *ptail = fifo_buf->tail;
    if(fifo_is_empty(fifo_buf) == FIFO_EMPTY)
    {
        Dprintf("[ERORR] The FIFO is empty \n");
    }
    else
    {
        Dprintf("Enter the Roll Number to update the entry :");
        Dscanf(&modified_roll,"%d");
        for(counter = 0 ; counter < fifo_buf->count ; counter++)
        {
            if(modified_roll == ptail->roll)
            {
                Dprintf("\t1.First Name\n");
                Dprintf("\t2.Last Name\n");
                Dprintf("\t3.Roll Number\n");
                Dprintf("\t4.GPA\n");
                Dprintf("\t5.courses\n");
                Dprintf("Choose the information you want to update :");
                Dscanf(&choice,"%d");
```

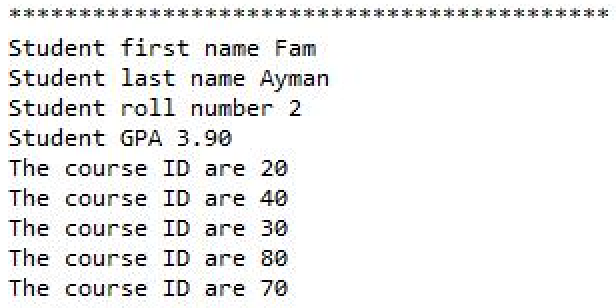


```
            switch(choice)
            {
            case 1:
                Dprintf("Enter the new first name :");
                Dscanf(ptail->fname ,"%s");
                break;
            case 2:
                Dprintf("Enter the new last name :");
                Dscanf(ptail->lname,"%s");
                break;
            case 3:
                Dprintf("Enter the new Roll Number :");
                Dscanf(&ptail->roll,"%d");
                break;
            case 4:
                Dprintf("Enter the new GPA :");
                Dscanf(&ptail->GPA,"%f");
                break;
            case 5:
                Dprintf("Enter the new course 1 ID :");
                Dscanf(&ptail->cid[0],"%d");
                Dprintf("Enter the new course 2 ID :");
                Dscanf(&ptail->cid[1],"%d");
                Dprintf("Enter the new course 3 ID :");
                Dscanf(&ptail->cid[2],"%d");
                Dprintf("Enter the new course 4 ID :");
                Dscanf(&ptail->cid[3],"%d");
                Dprintf("Enter the new course 5 ID :");
                Dscanf(&ptail->cid[4],"%d");
               break;
            }
            flag = 1;
        }

        ptail++;
    }
    if(flag==0)
    {
        Dprintf("[ERROR] Roll Number %d is not found \n",modified_roll);
    }
    Find_Total_Number(fifo_buf);
}
```

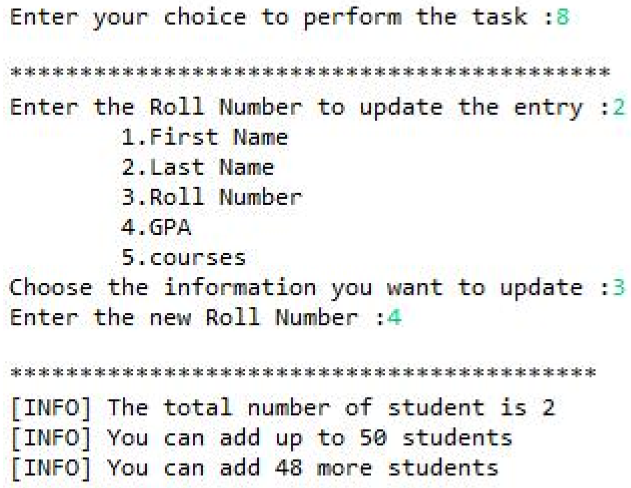
➢ API Test :



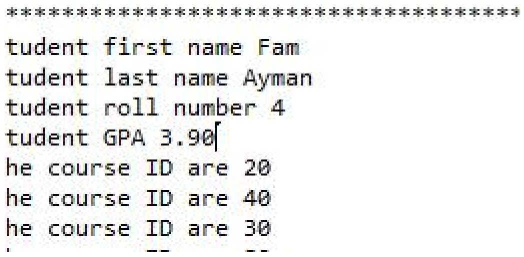
```
Enter your choice to perform the task :8

********************************************
Enter the Roll Number to update the entry :5
[ERROR] Roll Number 5 is not found

*******************************************
[INFO] The total number of student is 2
[INFO] You can add up to 50 students
[INFO] You can add 48 more students
```



```
*******************************************
Student first name Fam
Student last name Ayman
Student roll number 2
Student GPA 3.90
The course ID are 20
The course ID are 40
The course ID are 30
The course ID are 80
The course ID are 70
```



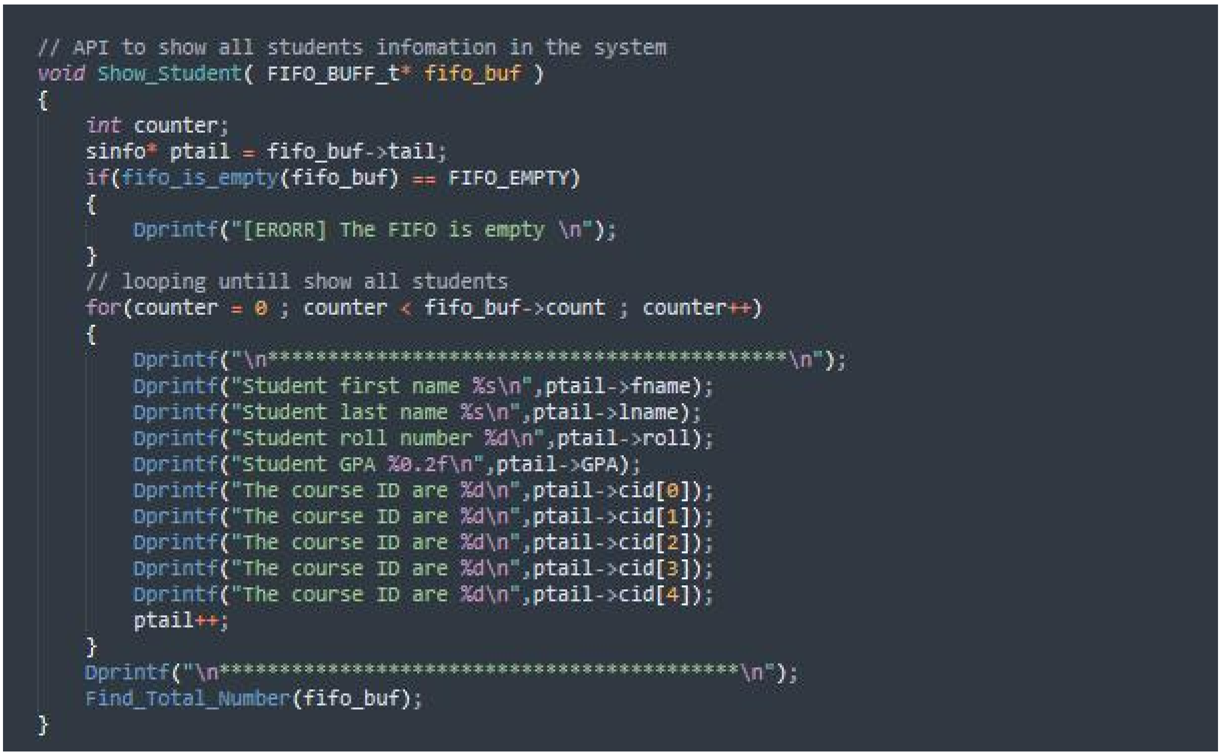
```
Enter your choice to perform the task :8

*******************************************
Enter the Roll Number to update the entry :2
        1.First Name
        2.Last Name
        3.Roll Number
        4.GPA
        5.courses
Choose the information you want to update :3
Enter the new Roll Number :4

******************************************
[INFO] The total number of student is 2
[INFO] You can add up to 50 students
[INFO] You can add 48 more students
```



```
*************************************
tudent first name Fam
tudent last name Ayman
tudent roll number 4
tudent GPA 3.90
he course ID are 20
he course ID are 40
he course ID are 30
```

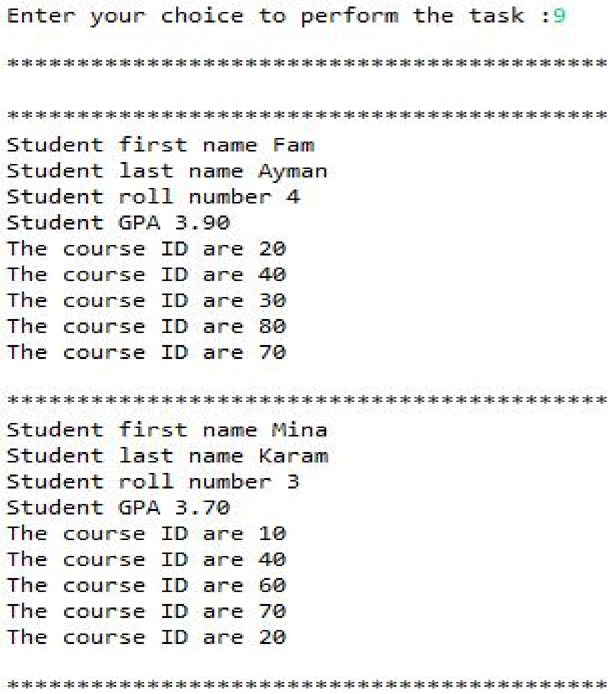
- Show All Information API:
  - Firstly , we call fifo_is_null function to check is FIFO null .
  - Call fifo_is_empty function to check if the FIFO empty
  - Start to show all students
  - Foe each student , we will show :
    - Read roll number
    - Read first name
    - Read last name
    - Read GPA
    - Read courses IDs

➢ API Code :



```
// API to show all students infomation in the system
void Show_Student( FIFO_BUFF_t* fifo_buf )
{
    int counter;
    sinfo* ptail = fifo_buf->tail;
    if(fifo_is_empty(fifo_buf) == FIFO_EMPTY)
    {
        Dprintf("[ERORR] The FIFO is empty \n");
    }
    // looping untill show all students
    for(counter = 0 ; counter < fifo_buf->count ; counter++)
    {
        Dprintf("\n********************************************\n");
        Dprintf("Student first name %s\n",ptail->fname);
        Dprintf("Student last name %s\n",ptail->lname);
        Dprintf("Student roll number %d\n",ptail->roll);
        Dprintf("Student GPA %0.2f\n",ptail->GPA);
        Dprintf("The course ID are %d\n",ptail->cid[0]);
        Dprintf("The course ID are %d\n",ptail->cid[1]);
        Dprintf("The course ID are %d\n",ptail->cid[2]);
        Dprintf("The course ID are %d\n",ptail->cid[3]);
        Dprintf("The course ID are %d\n",ptail->cid[4]);
        ptail++;
    }
    Dprintf("\n********************************************\n");
    Find_Total_Number(fifo_buf);
}
```

➢ API Test :



```
Enter your choice to perform the task :9

*******************************************

*******************************************
Student first name Fam
Student last name Ayman
Student roll number 4
Student GPA 3.90
The course ID are 20
The course ID are 40
The course ID are 30
The course ID are 80
The course ID are 70

*******************************************
Student first name Mina
Student last name Karam
Student roll number 3
Student GPA 3.70
The course ID are 10
The course ID are 40
The course ID are 60
The course ID are 70
The course ID are 20

*******************************************
```