

|  |
| --- |
| **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**
* **Approach**
* **Idea behind operation**
* **Codes Implementation and Testing**
* **Main.c**
* **Student.h**
* **Student.c**
* **Add the Student Details Manually**
* **Add the Student Details From Text File**
* **Find the Student Details by Roll Number**
* **Find the Student Details by First Name**
* **Find the Student Details by Course ID**
* **Find the Total Number Of Students**
* **Delete the Students Details by Roll Number**
* **Update the Students Details by Roll Number**
* **Show All Information**
* **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 .

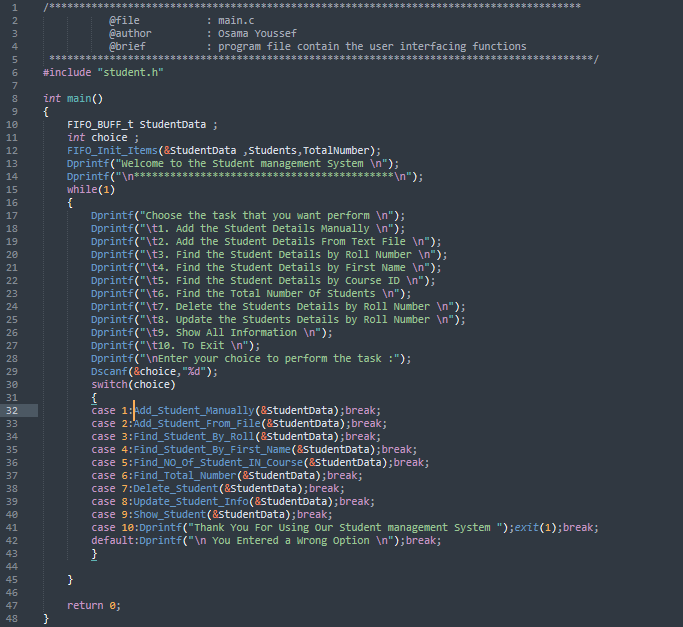
1. We will have array of struct of configurable variable TotalNumber which is the max number of students can system manipulate .
2. Each struct contains the student details.
3. We will have a global index which refers to the number of students.
4. The index is initialized with zero and can go up to 50.

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 .

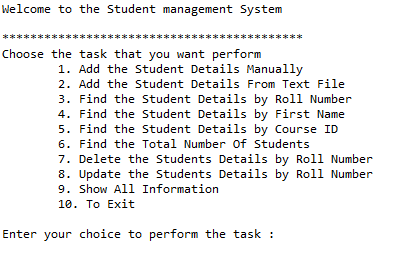
* **Codes Implementation and Testing :**
* **Main.c Code :**

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.

****

* **Main.c Test :**

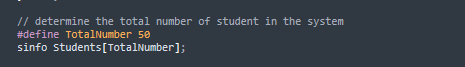


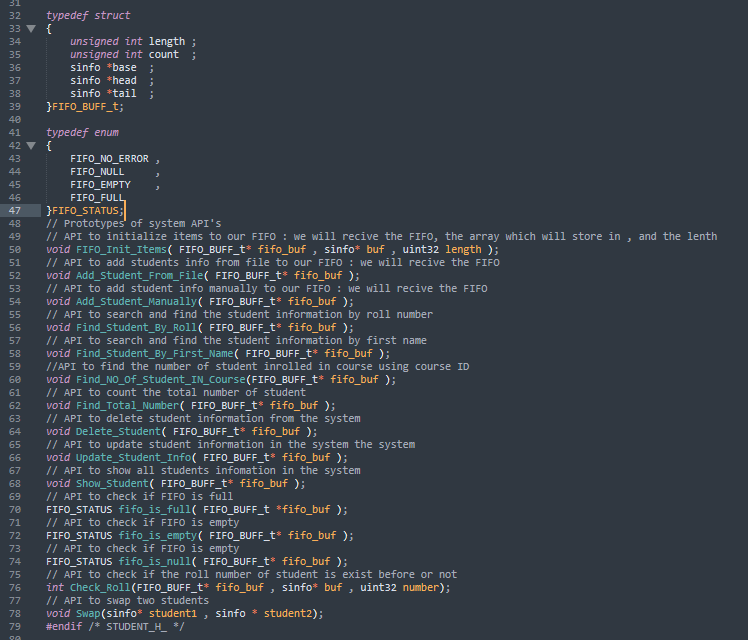
* **Student.h\_Code :**

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.

****





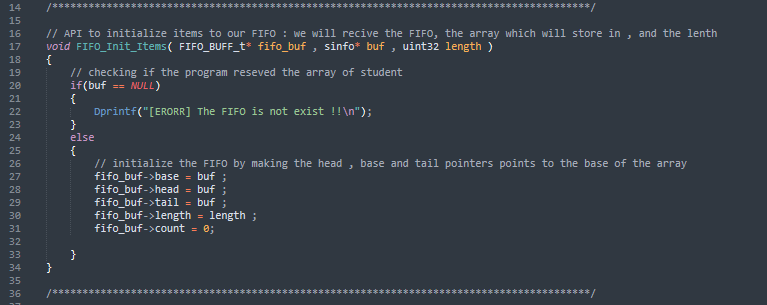
* **Student.c :**

he Student.c consist of :

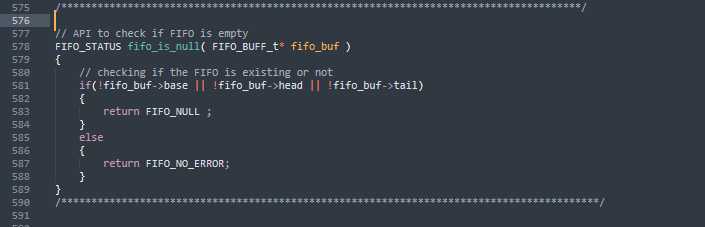
* API Implementation
* FIFO initialization API :
* Firstly , we check if the FIFO is null(exist).
* We start to link between the FIFO and the array.
* We make the head , base and tail pointers points to the base

of the array .

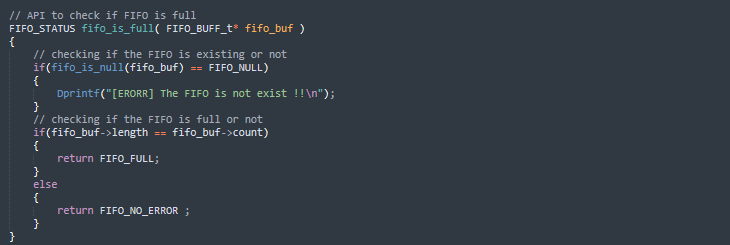
* Set the length and counter .



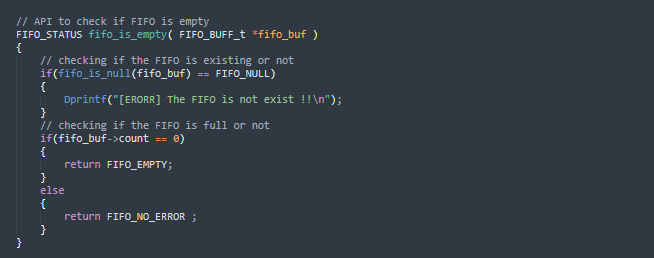
* FIFO is null API :
* Check if the FIFO is exist or not .
* Return the FIFO status.



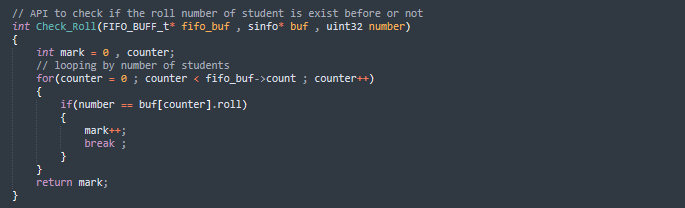
* FIFO is full API :
* Firstly , we call fifo\_is\_null function to check is FIFO null .
* Check if the FIFO is full or not.
* Return the FIFO status .



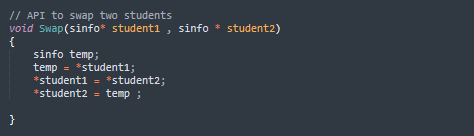
* FIFO is empty API :
* Firstly , we call fifo\_is\_null function to check is FIFO null .
* Check if the FIFO is empty or not .
* Return the FIFO status.



* Check Roll API :
* Looping to all system student and check if roll of any student is identical to the given roll number.

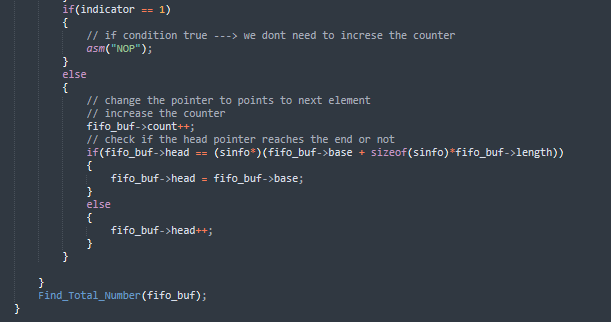


* Swap Students API :
* Creating temp sinfo and then swap the two students locations .

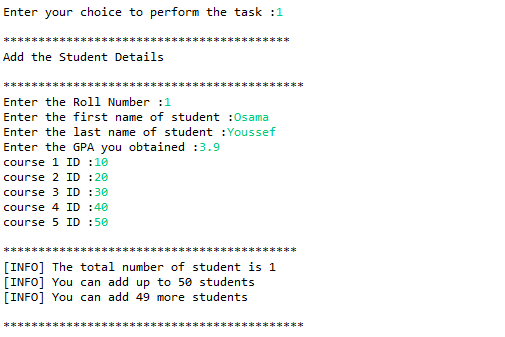


* Add the Student Details Manually API :
* Firstly , we call fifo\_is\_null function to check is FIFO null .
* Call fifo\_is\_full function to check if the FIFO full
* Check if the roll number is token or not
* If roll number is token we terminate the process
* If roll number is unique we start to receive student info
* API Code :



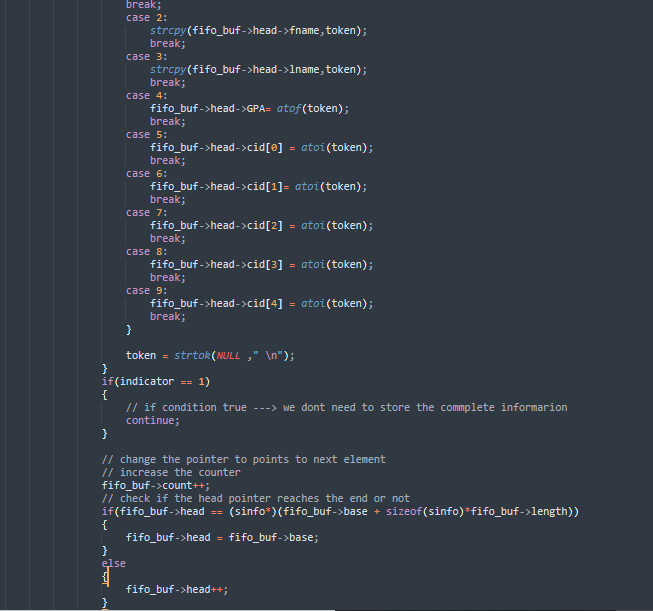


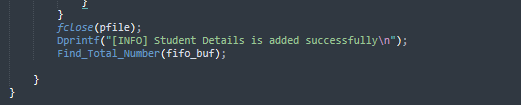
* API Test :



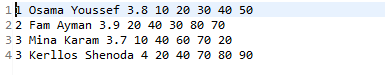
* Add the Student Details Manually API:
* Firstly , we call fifo\_is\_null function to check is FIFO null .
* Call fifo\_is\_full function to check if the FIFO full
* We create FILE pointer pfile to pointer to our file
* Start to read line by line and the tokan is an space
  + Read roll number
  + Read 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 :

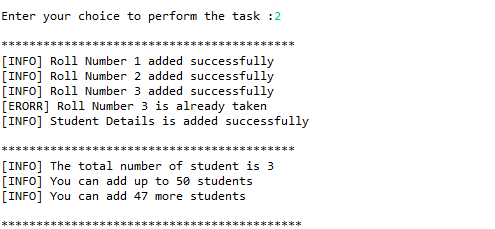


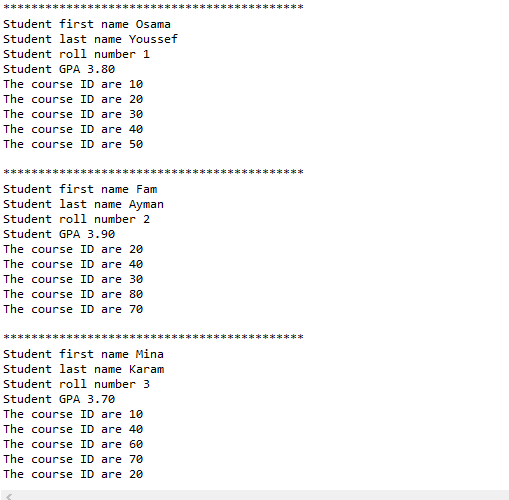




* API Test :



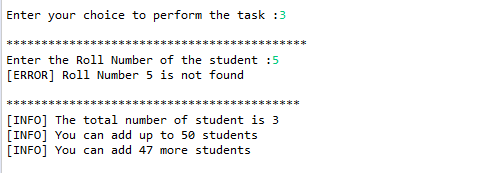


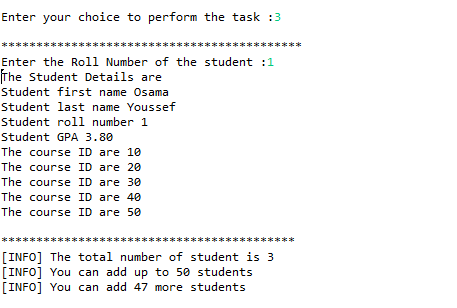


* Find the Student Details by Roll Number API:
* Firstly , we call fifo\_is\_null function to check is FIFO null .
* Call fifo\_is\_empty function to check if the FIFO empty
* Ask user to enter the roll number
* If the user entered roll number is not found message the user by calling check\_roll function
* If found , show informations of the student
* API Code :

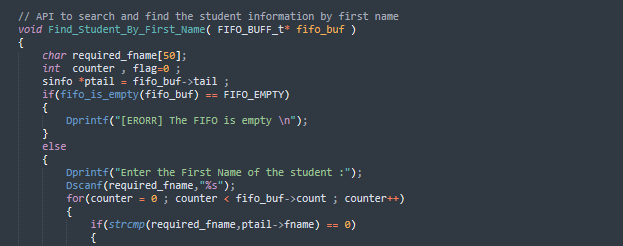


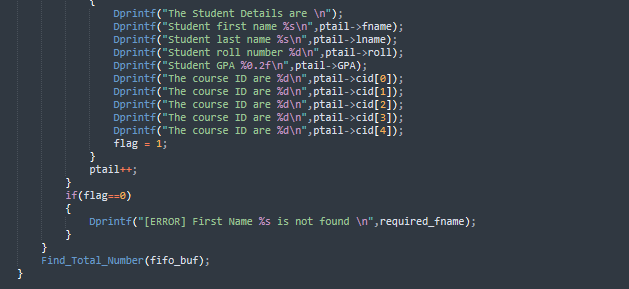
* API Test :



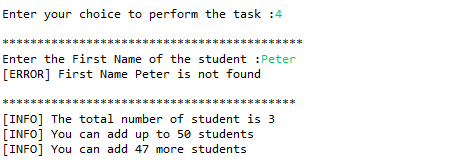


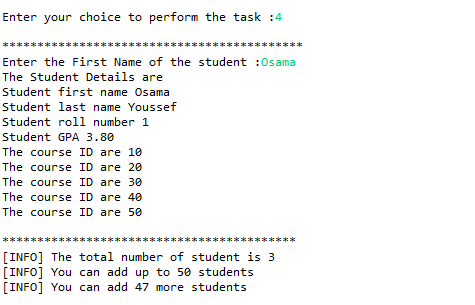
* Find the Student Details by First Name API:
* Firstly , we call fifo\_is\_null function to check is FIFO null .
* Call fifo\_is\_empty function to check if the FIFO empty
* Ask user to enter the first name
* If found , show informations of the student





* API Test :

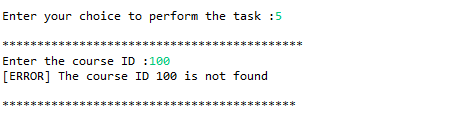


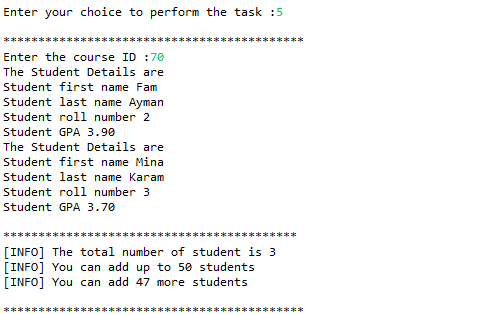


* Find the Student Details by Course ID API:
* Firstly , we call fifo\_is\_null function to check is FIFO null .
* Call fifo\_is\_empty function to check if the FIFO empty.
* Ask user to enter the course ID.
* If found , show informations of the student.
* API Code :

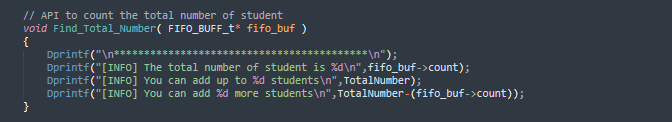


* API Test :

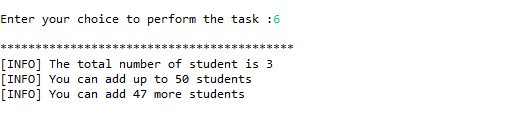




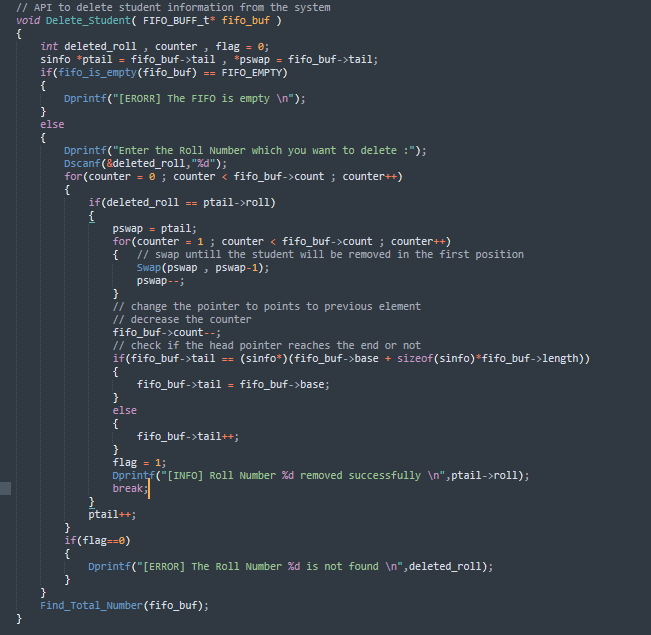
* Find the Total Number Of Students API:
* Firstly , we call fifo\_is\_null function to check is FIFO null .
* Call fifo\_is\_empty function to check if the FIFO empty.
* Start to show number of student.
* API Code :



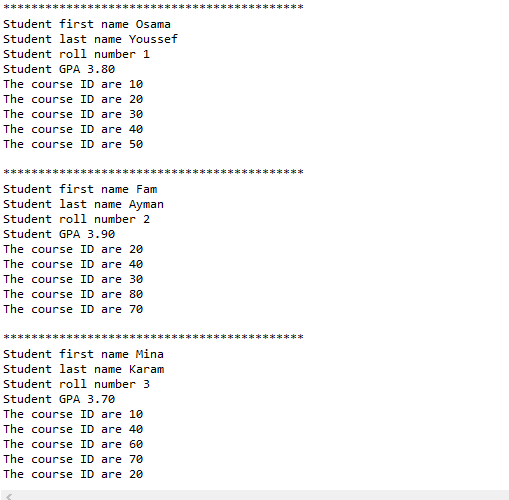
* API Test :

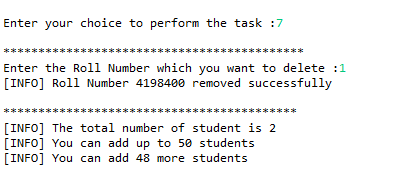


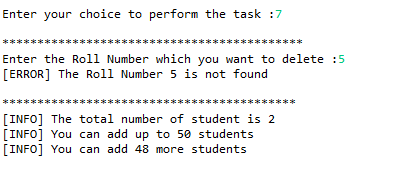
* Delete the Students Details by Roll Number API:
* Firstly , we call fifo\_is\_null function to check is FIFO null .
* Call fifo\_is\_empty function to check if the FIFO empty.
* Ask user to enter the roll number which will be deleted.
* If the user entered roll number is not found message the user.
* 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 :

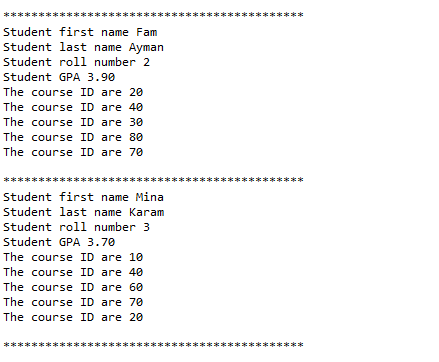


* API Test :



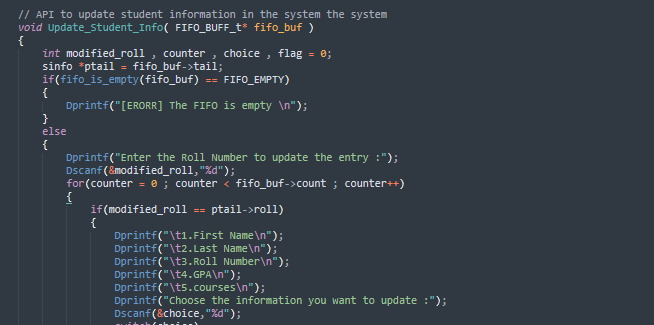


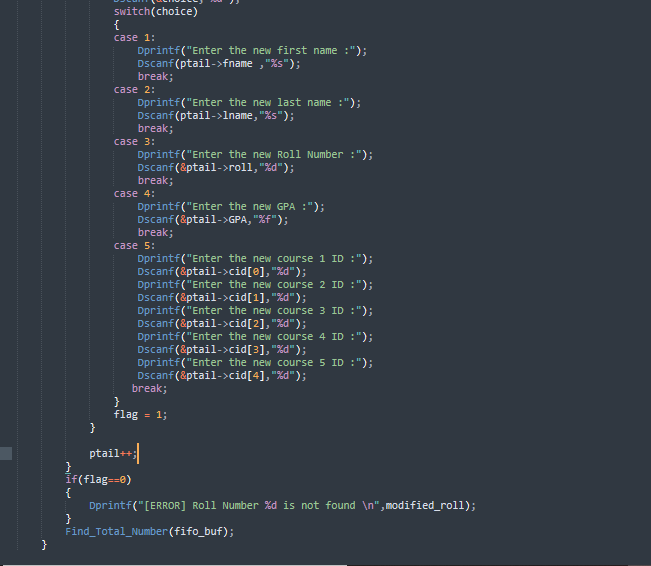




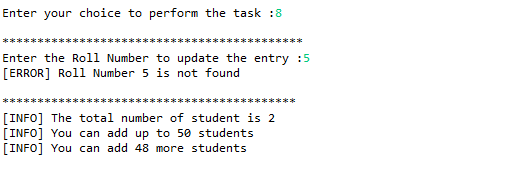
* Update Student Information by Roll Number API:
* Firstly , we call fifo\_is\_null function to check is FIFO null .
* Call fifo\_is\_empty function to check if the FIFO empty
* Ask user to enter the roll number which will be deleted
* If the user entered roll number is not found message the user.
* Start to switch in option of user about any information will be updated.
* Ask the user to enter the new data

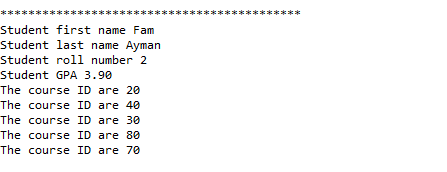
* API Code :

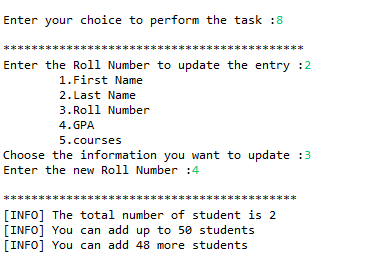


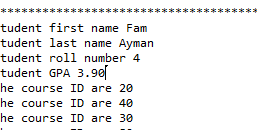


* API Test :

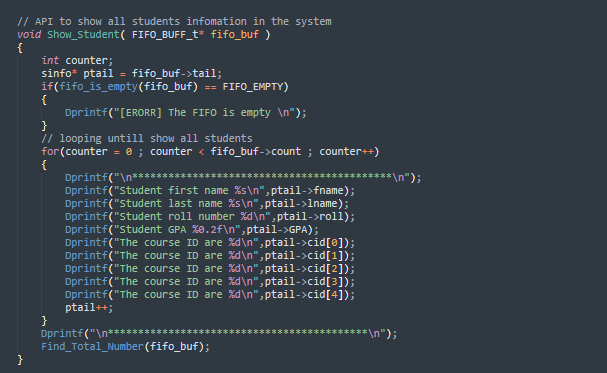








* Show All Information API:
* Firstly , we call fifo\_is\_null function to check is FIFO null .
* Call fifo\_is\_empty function to check if the FIFO empty
* Start to show all students
* Foe each student , we will show :
  + Read roll number
  + Read first name
  + Read last name
  + Read GPA
* Read courses IDs
* API Code :



* API Test :

