Report

Student management system

Using queue

Eng. Mostafa Rashed Kamel Mohamed

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 records.

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

Let’s see code

1-Structures, Enum and buffer:

as we mentioned before

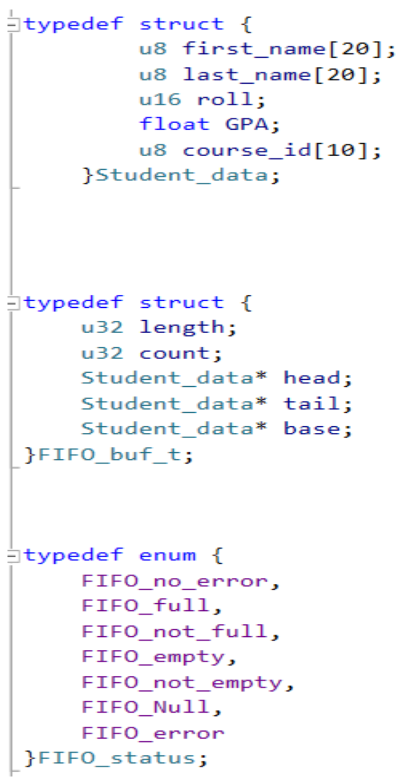
-structure to represent student

object.

- Buffer to contain the whole database.

- structure to control buffer.

- Enum to represent status.



```c
typedef struct {
            u8 first_name[20];
            u8 last_name[20];
            u16 roll;
            float GPA;
            u8 course_id[10];
        }Student_data;



typedef struct {
        u32 length;
        u32 count;
        Student_data* head;
        Student_data* tail;
        Student_data* base;
}FIFO_buf_t;


typedef enum {
        FIFO_no_error,
        FIFO_full,
        FIFO_not_full,
        FIFO_empty,
        FIFO_not_empty,
        FIFO_Null,
        FIFO_error
}FIFO_status;
```

2- initialization:

this function takes pointer to structure

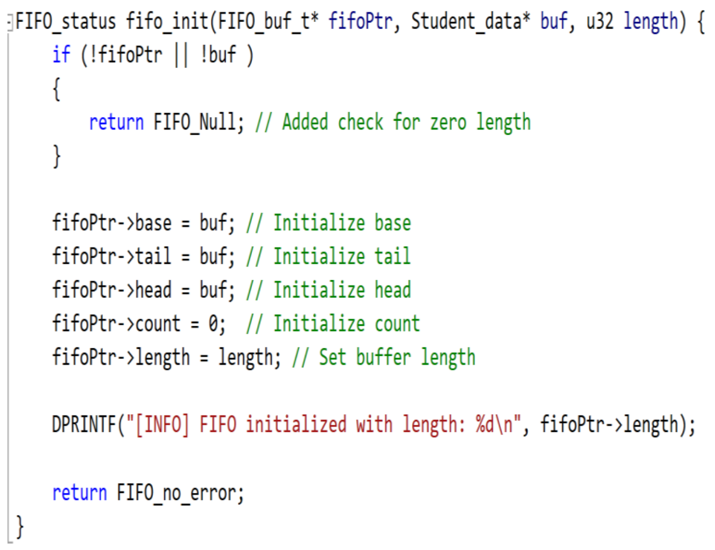
(controller buffer) and pointer to

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.



```c
FIFO_status fifo_init(FIFO_buf_t* fifoPtr, Student_data* buf, u32 length) {
    if (!fifoPtr || !buf )
    {
        return FIFO_Null; // Added check for zero length
    }

    fifoPtr->base = buf; // Initialize base
    fifoPtr->tail = buf; // Initialize tail
    fifoPtr->head = buf; // Initialize head
    fifoPtr->count = 0;  // Initialize count
    fifoPtr->length = length; // Set buffer length

    DPRINTF("[INFO] FIFO initialized with length: %d\n", fifoPtr->length);

    return FIFO_no_error;
}
```

3- Check for Roll number

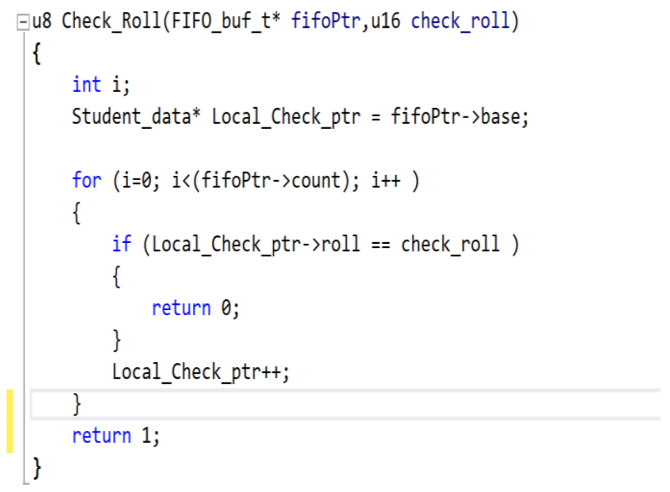
this function takes pointer to buffer controller and take integer number represents Roll number wanted to be check if exists before or not because the

roll number must be a unique number for

a student.

It returns 0 if roll number found and 1

If not found.



```c
u8 Check_Roll(FIFO_buf_t* fifoPtr,u16 check_roll)
{
    int i;
    Student_data* Local_Check_ptr = fifoPtr->base;

    for (i=0; i<(fifoPtr->count); i++ )
    {
        if (Local_Check_ptr->roll == check_roll )
        {
            return 0;
        }
        Local_Check_ptr++;
    }
    return 1;
}
```

4- add student manually

- this function to add students details manually takes pointer to

Buffer controller.

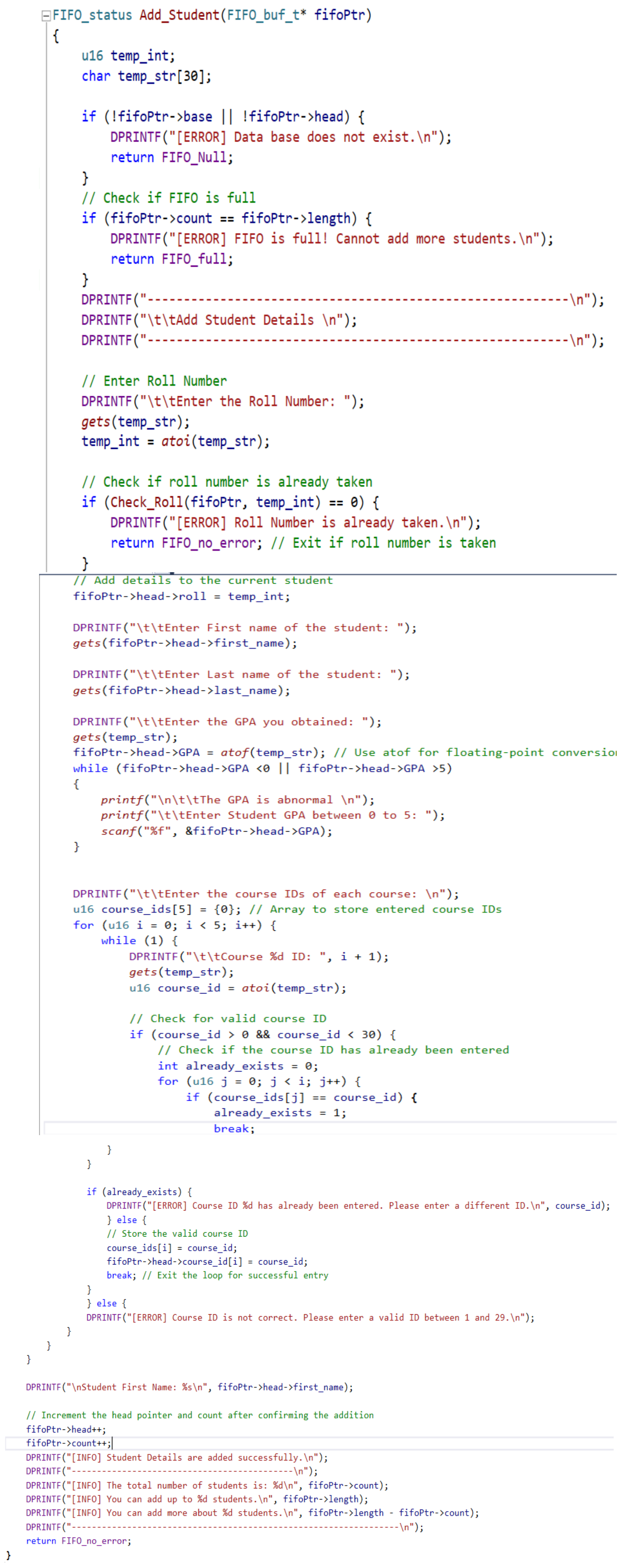
-starts with checking if the buffer is existing or not then checks if the buffer is full or not.

- 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

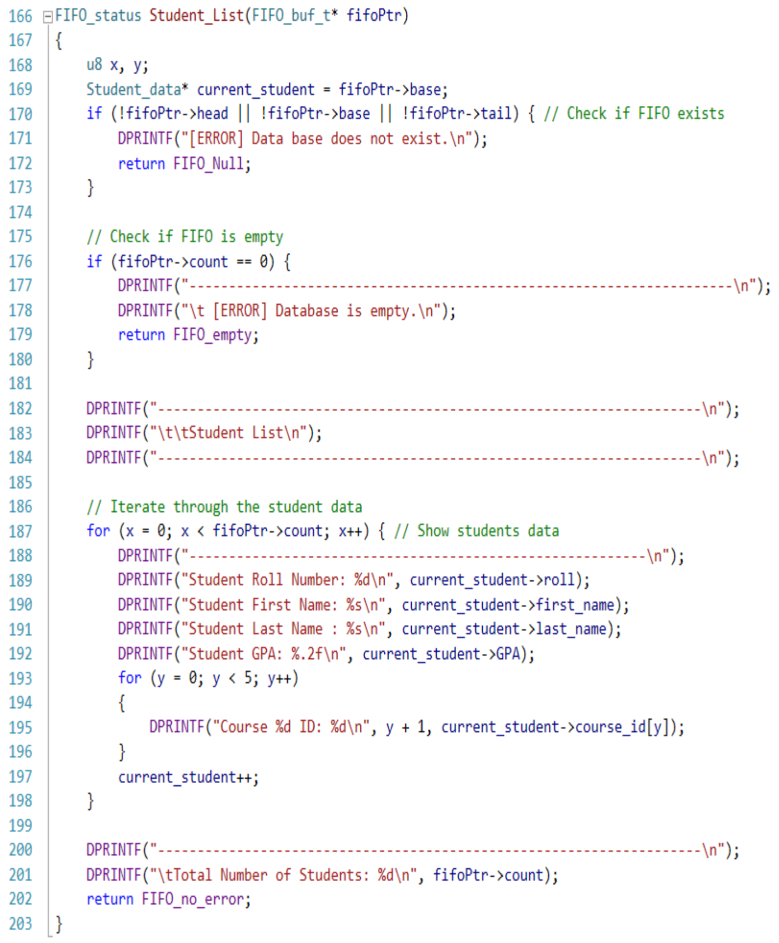
- if everything is going well, the function finally prints the max number of students that the buffer can hold and the number of empty places for new records.



```c
FIFO_status Add_Student(FIFO_buf_t* fifoPtr)
{
    u16 temp_int;
    char temp_str[30];

    if (!fifoPtr->base || !fifoPtr->head) {
        DPRINTF("[ERROR] Data base does not exist.\n");
        return FIFO_Null;
    }
    // Check if FIFO is full
    if (fifoPtr->count == fifoPtr->length) {
        DPRINTF("[ERROR] FIFO is full! Cannot add more students.\n");
        return FIFO_full;
    }
    DPRINTF("------------------------------------------------------------\n");
    DPRINTF("\t\tAdd Student Details \n");
    DPRINTF("------------------------------------------------------------\n");

    // Enter Roll Number
    DPRINTF("\t\tEnter the Roll Number: ");
    gets(temp_str);
    temp_int = atoi(temp_str);

    // Check if roll number is already taken
    if (Check_Roll(fifoPtr, temp_int) == 0) {
        DPRINTF("[ERROR] Roll Number is already taken.\n");
        return FIFO_no_error; // Exit if roll number is taken
    }
    // Add details to the current student
    fifoPtr->head->roll = temp_int;

    DPRINTF("\t\tEnter First name of the student: ");
    gets(fifoPtr->head->first_name);

    DPRINTF("\t\tEnter Last name of the student: ");
    gets(fifoPtr->head->last_name);

    DPRINTF("\t\tEnter the GPA you obtained: ");
    gets(temp_str);
    fifoPtr->head->GPA = atof(temp_str); // Use atof for floating-point conversio
    while (fifoPtr->head->GPA <0 || fifoPtr->head->GPA >5)
    {
        printf("\n\t\tThe GPA is abnormal \n");
        printf("\t\tEnter Student GPA between 0 to 5: ");
        scanf("%f", &fifoPtr->head->GPA);
    }


    DPRINTF("\t\tEnter the course IDs of each course: \n");
    u16 course_ids[5] = {0}; // Array to store entered course IDs
    for (u16 i = 0; i < 5; i++) {
        while (1) {
            DPRINTF("\t\tCourse %d ID: ", i + 1);
            gets(temp_str);
            u16 course_id = atoi(temp_str);

            // Check for valid course ID
            if (course_id > 0 && course_id < 30) {
                // Check if the course ID has already been entered
                int already_exists = 0;
                for (u16 j = 0; j < i; j++) {
                    if (course_ids[j] == course_id) {
                        already_exists = 1;
                        break;
                    }
                }

                if (already_exists) {
                    DPRINTF("[ERROR] Course ID %d has already been entered. Please enter a different ID.\n", course_id);
                    } else {
                    // Store the valid course ID
                    course_ids[i] = course_id;
                    fifoPtr->head->course_id[i] = course_id;
                    break; // Exit the loop for successful entry
                }
                } else {
                DPRINTF("[ERROR] Course ID is not correct. Please enter a valid ID between 1 and 29.\n");
            }
        }
    }

    DPRINTF("\nStudent First Name: %s\n", fifoPtr->head->first_name);

    // Increment the head pointer and count after confirming the addition
    fifoPtr->head++;
    fifoPtr->count++;
    DPRINTF("[INFO] Student Details are added successfully.\n");
    DPRINTF("---------------------------------------------\n");
    DPRINTF("[INFO] The total number of students is: %d\n", fifoPtr->count);
    DPRINTF("[INFO] You can add up to %d students.\n", fifoPtr->length);
    DPRINTF("[INFO] You can add more about %d students.\n", fifoPtr->length - fifoPtr->count);
    DPRINTF("--------------------------------------------------------------\n");
    return FIFO_no_error;
}
```

5-Show all database

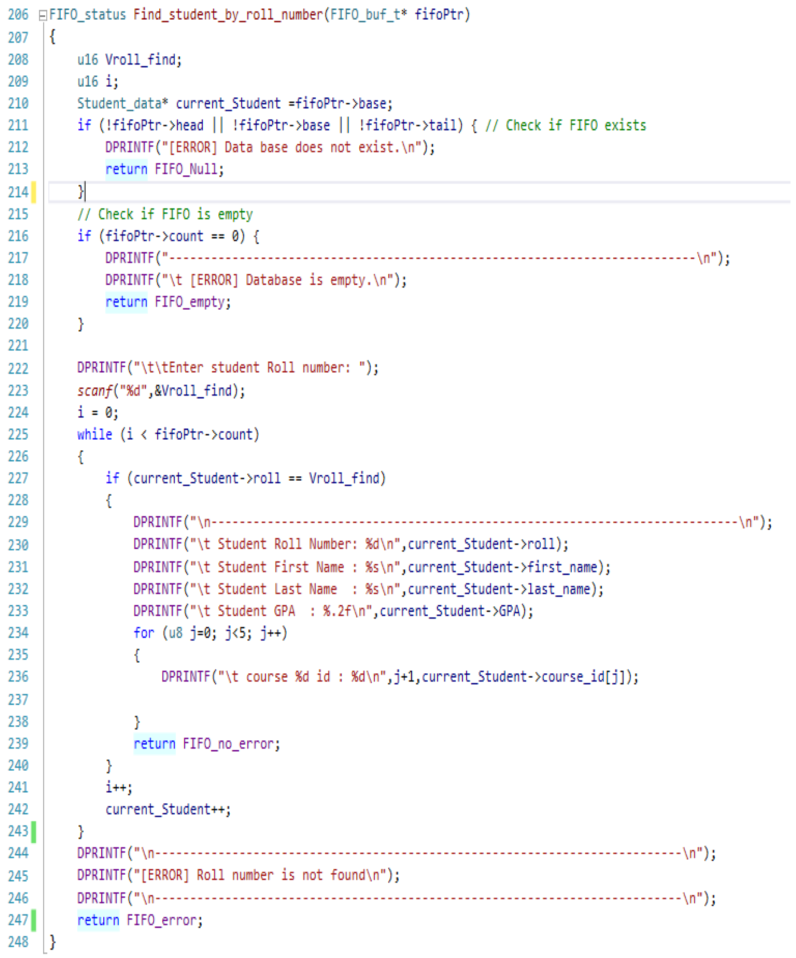
- this function responsible for display all data for all students registered.
- It starts with checking if buffer is existing or not and check if the buffer is empty or not.
- It loops on the buffer and extract all data
- Finally, it informs us the total number of students



```c
FIFO_status Student_List(FIFO_buf_t* fifoPtr)
{
    u8 x, y;
    Student_data* current_student = fifoPtr->base;
    if (!fifoPtr->head || !fifoPtr->base || !fifoPtr->tail) { // Check if FIFO exists
        DPRINTF("[ERROR] Data base does not exist.\n");
        return FIFO_Null;
    }

    // Check if FIFO is empty
    if (fifoPtr->count == 0) {
        DPRINTF("--------------------------------------------------------------------\n");
        DPRINTF("\t [ERROR] Database is empty.\n");
        return FIFO_empty;
    }

    DPRINTF("--------------------------------------------------------------------\n");
    DPRINTF("\t\tStudent List\n");
    DPRINTF("--------------------------------------------------------------------\n");

    // Iterate through the student data
    for (x = 0; x < fifoPtr->count; x++) { // Show students data
        DPRINTF("------------------------------------------------------------\n");
        DPRINTF("Student Roll Number: %d\n", current_student->roll);
        DPRINTF("Student First Name: %s\n", current_student->first_name);
        DPRINTF("Student Last Name : %s\n", current_student->last_name);
        DPRINTF("Student GPA: %.2f\n", current_student->GPA);
        for (y = 0; y < 5; y++)
        {
            DPRINTF("Course %d ID: %d\n", y + 1, current_student->course_id[y]);
        }
        current_student++;
    }

    DPRINTF("--------------------------------------------------------------------\n");
    DPRINTF("\tTotal Number of Students: %d\n", fifoPtr->count);
    return FIFO_no_error;
}
```

6-find by roll number

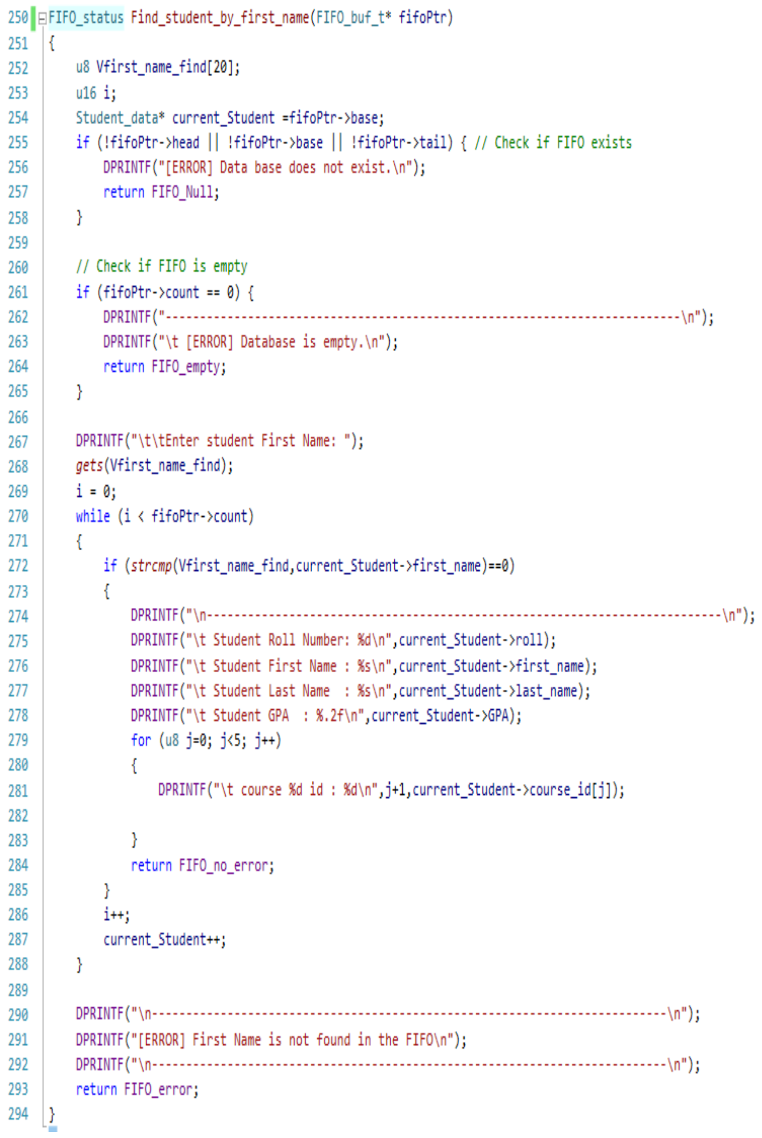
- this function responsible for find the student data by roll number
- It starts with checking if buffer is existing or not and check if the buffer is empty or not.
- Then loop on all database to get student information by roll number.
- If roll number is not found function will inform us.



```c
FIFO_status Find_student_by_roll_number(FIFO_buf_t* fifoPtr)
{
    u16 Vroll_find;
    u16 i;
    Student_data* current_Student =fifoPtr->base;
    if (!fifoPtr->head || !fifoPtr->base || !fifoPtr->tail) { // Check if FIFO exists
        DPRINTF("[ERROR] Data base does not exist.\n");
        return FIFO_Null;
    }
    // Check if FIFO is empty
    if (fifoPtr->count == 0) {
        DPRINTF("---------------------------------------------------------------------\n");
        DPRINTF("\t [ERROR] Database is empty.\n");
        return FIFO_empty;
    }

    DPRINTF("\t\tEnter student Roll number: ");
    scanf("%d",&Vroll_find);
    i = 0;
    while (i < fifoPtr->count)
    {
        if (current_Student->roll == Vroll_find)
        {
            DPRINTF("\n---------------------------------------------------------------------\n");
            DPRINTF("\t Student Roll Number: %d\n",current_Student->roll);
            DPRINTF("\t Student First Name : %s\n",current_Student->first_name);
            DPRINTF("\t Student Last Name  : %s\n",current_Student->last_name);
            DPRINTF("\t Student GPA  : %.2f\n",current_Student->GPA);
            for (u8 j=0; j<5; j++)
            {
                DPRINTF("\t course %d id : %d\n",j+1,current_Student->course_id[j]);

            }
            return FIFO_no_error;
        }
        i++;
        current_Student++;
    }
    DPRINTF("\n---------------------------------------------------------------------\n");
    DPRINTF("[ERROR] Roll number is not found\n");
    DPRINTF("\n---------------------------------------------------------------------\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.



```c
FIFO_status Find_student_by_first_name(FIFO_buf_t* fifoPtr)
{
    u8 Vfirst_name_find[20];
    u16 i;
    Student_data* current_Student =fifoPtr->base;
    if (!fifoPtr->head || !fifoPtr->base || !fifoPtr->tail) { // Check if FIFO exists
        DPRINTF("[ERROR] Data base does not exist.\n");
        return FIFO_Null;
    }

    // Check if FIFO is empty
    if (fifoPtr->count == 0) {
        DPRINTF("-----------------------------------------------------------------------\n");
        DPRINTF("\t [ERROR] Database is empty.\n");
        return FIFO_empty;
    }

    DPRINTF("\t\tEnter student First Name: ");
    gets(Vfirst_name_find);
    i = 0;
    while (i < fifoPtr->count)
    {
        if (strcmp(Vfirst_name_find,current_Student->first_name)==0)
        {
            DPRINTF("\n-----------------------------------------------------------------------\n");
            DPRINTF("\t Student Roll Number: %d\n",current_Student->roll);
            DPRINTF("\t Student First Name : %s\n",current_Student->first_name);
            DPRINTF("\t Student Last Name  : %s\n",current_Student->last_name);
            DPRINTF("\t Student GPA  : %.2f\n",current_Student->GPA);
            for (u8 j=0; j<5; j++)
            {
                DPRINTF("\t course %d id : %d\n",j+1,current_Student->course_id[j]);

            }
            return FIFO_no_error;
        }
        i++;
        current_Student++;
    }

    DPRINTF("\n-----------------------------------------------------------------------\n");
    DPRINTF("[ERROR] First Name is not found in the FIFO\n");
    DPRINTF("\n-----------------------------------------------------------------------\n");
    return FIFO_error;
}
```

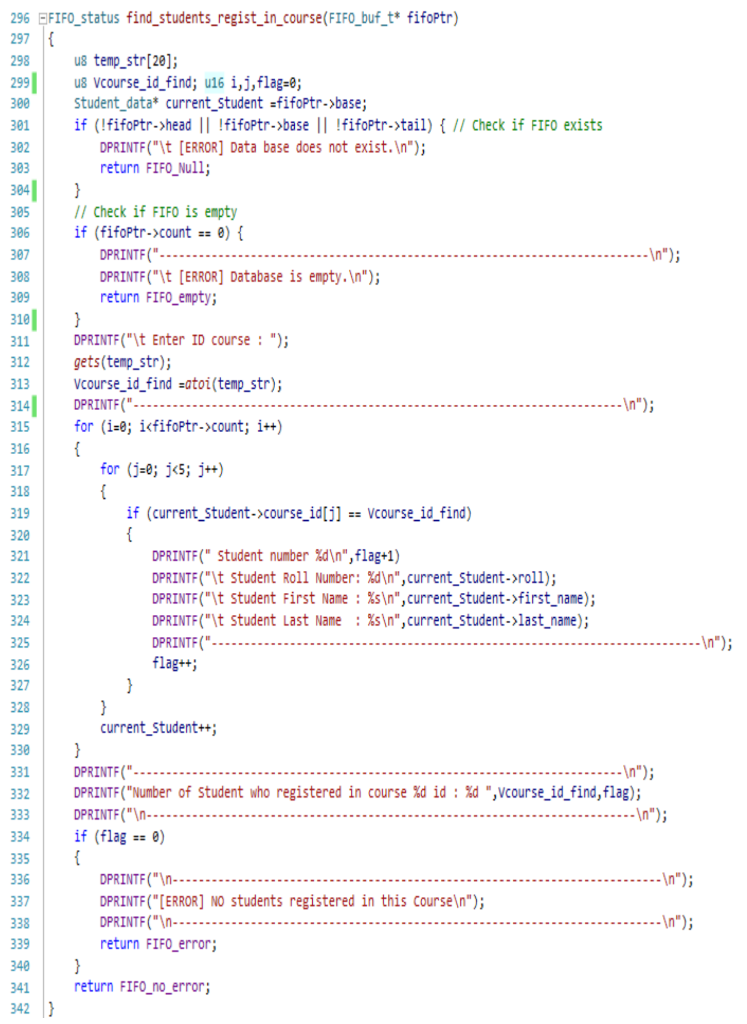
8-find all students registered in a specific course using course id

-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.

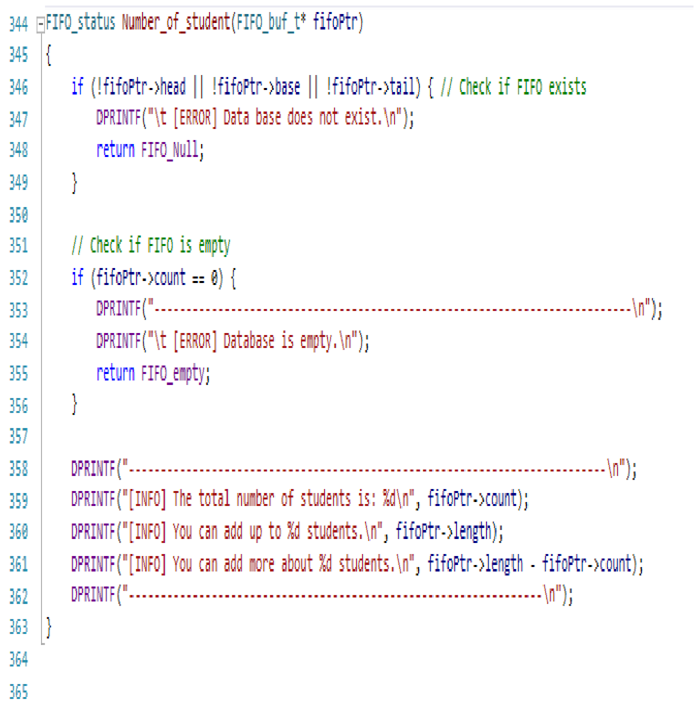
- if the loop finished and there are no students registered this course the function will print message to inform us.



```c
FIFO_status find_students_regist_in_course(FIFO_buf_t* fifoPtr)
{
    u8 temp_str[20];
    u8 Vcourse_id_find; u16 i,j,flag=0;
    Student_data* current_Student =fifoPtr->base;
    if (!fifoPtr->head || !fifoPtr->base || !fifoPtr->tail) { // Check if FIFO exists
        DPRINTF("\t [ERROR] Data base does not exist.\n");
        return FIFO_Null;
    }
    // Check if FIFO is empty
    if (fifoPtr->count == 0) {
        DPRINTF("----------------------------------------------------------------------------\n");
        DPRINTF("\t [ERROR] Database is empty.\n");
        return FIFO_empty;
    }
    DPRINTF("\t Enter ID course : ");
    gets(temp_str);
    Vcourse_id_find =atoi(temp_str);
    DPRINTF("----------------------------------------------------------------------------\n");
    for (i=0; i<fifoPtr->count; i++)
    {
        for (j=0; j<5; j++)
        {
            if (current_Student->course_id[j] == Vcourse_id_find)
            {
                DPRINTF(" Student number %d\n",flag+1)
                DPRINTF("\t Student Roll Number: %d\n",current_Student->roll);
                DPRINTF("\t Student First Name : %s\n",current_Student->first_name);
                DPRINTF("\t Student Last Name  : %s\n",current_Student->last_name);
                DPRINTF("----------------------------------------------------------------------------\n");
                flag++;
            }
        }
        current_Student++;
    }
    DPRINTF("----------------------------------------------------------------------------\n");
    DPRINTF("Number of Student who registered in course %d id : %d ",Vcourse_id_find,flag);
    DPRINTF("\n----------------------------------------------------------------------------\n");
    if (flag == 0)
    {
        DPRINTF("\n----------------------------------------------------------------------------\n");
        DPRINTF("[ERROR] NO students registered in this Course\n");
        DPRINTF("\n----------------------------------------------------------------------------\n");
        return FIFO_error;
    }
    return FIFO_no_error;
}
```

9-Total number of students

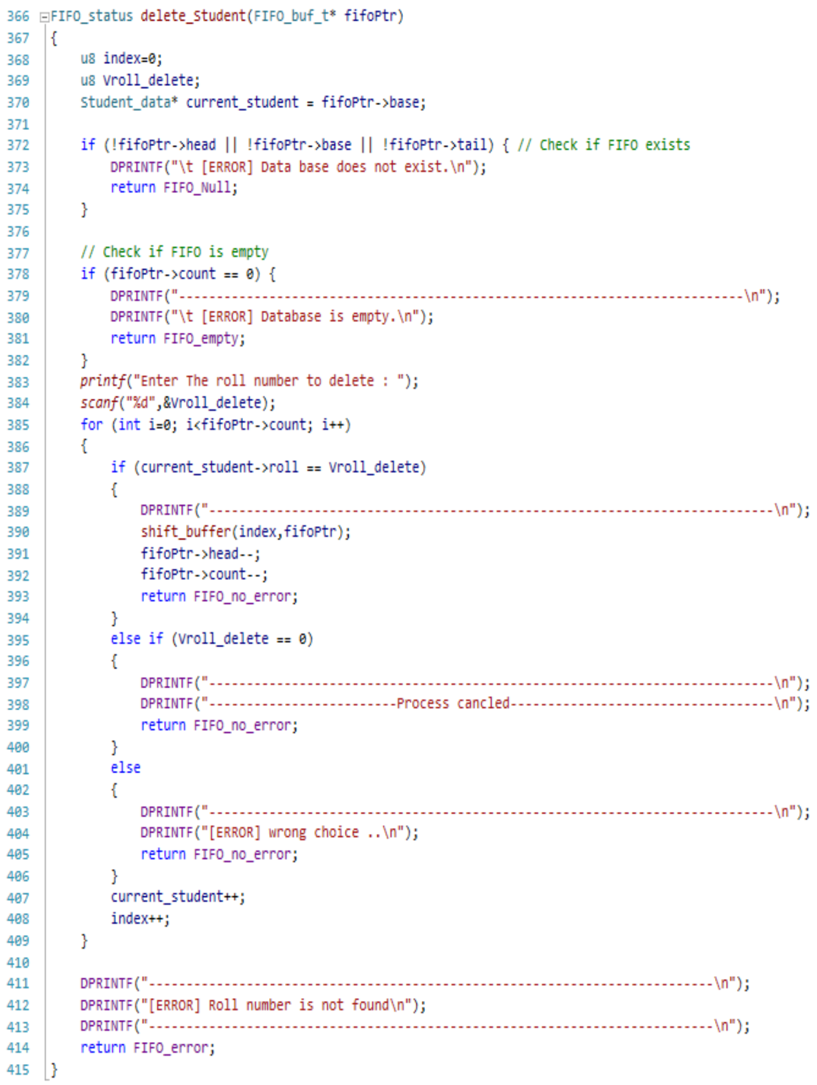
This function gets the total number of students in the database



```c
FIFO_status Number_of_student(FIFO_buf_t* fifoPtr)
{
    if (!fifoPtr->head || !fifoPtr->base || !fifoPtr->tail) { // Check if FIFO exists
        DPRINTF("\t [ERROR] Data base does not exist.\n");
        return FIFO_Null;
    }

    // Check if FIFO is empty
    if (fifoPtr->count == 0) {
        DPRINTF("----------------------------------------------------------------------------\n");
        DPRINTF("\t [ERROR] Database is empty.\n");
        return FIFO_empty;
    }

    DPRINTF("----------------------------------------------------------------------------\n");
    DPRINTF("[INFO] The total number of students is: %d\n", fifoPtr->count);
    DPRINTF("[INFO] You can add up to %d students.\n", fifoPtr->length);
    DPRINTF("[INFO] You can add more about %d students.\n", fifoPtr->length - fifoPtr->count);
    DPRINTF("----------------------------------------------------------------------------\n");
}
```

10- Delete student using roll number

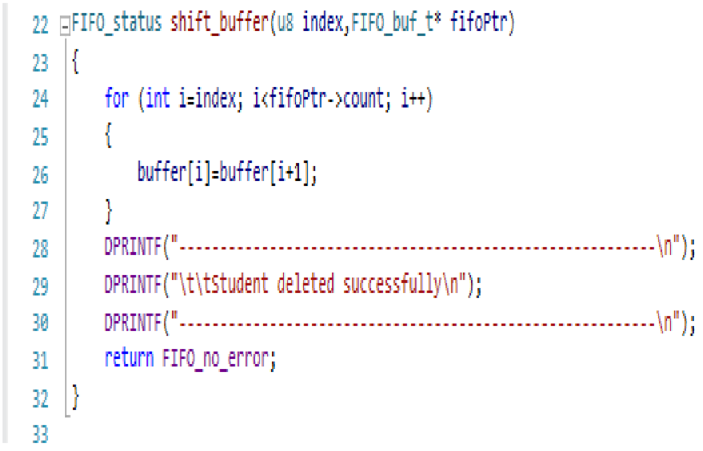
- this function deletes student registration by roll number.
- First it loops on database to find the roll number
- It displays student information before delete
- Then asks us to confirm deleting process
- If roll number is not found it will inform us
- If we enter wrong option It will inform and back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



```c
FIFO_status delete_Student(FIFO_buf_t* fifoPtr)
{
    u8 index=0;
    u8 Vroll_delete;
    Student_data* current_student = fifoPtr->base;

    if (!fifoPtr->head || !fifoPtr->base || !fifoPtr->tail) { // Check if FIFO exists
        DPRINTF("\t [ERROR] Data base does not exist.\n");
        return FIFO_Null;
    }

    // Check if FIFO is empty
    if (fifoPtr->count == 0) {
        DPRINTF("-----------------------------------------------------------------------\n");
        DPRINTF("\t [ERROR] Database is empty.\n");
        return FIFO_empty;
    }
    printf("Enter The roll number to delete : ");
    scanf("%d",&Vroll_delete);
    for (int i=0; i<fifoPtr->count; i++)
    {
        if (current_student->roll == Vroll_delete)
        {
            DPRINTF("-----------------------------------------------------------------------\n");
            shift_buffer(index,fifoPtr);
            fifoPtr->head--;
            fifoPtr->count--;
            return FIFO_no_error;
        }
        else if (Vroll_delete == 0)
        {
            DPRINTF("-----------------------------------------------------------------------\n");
            DPRINTF("-----------------------Process cancled---------------------------------\n");
            return FIFO_no_error;
        }
        else
        {
            DPRINTF("-----------------------------------------------------------------------\n");
            DPRINTF("[ERROR] wrong choice ..\n");
            return FIFO_no_error;
        }
        current_student++;
        index++;
    }

    DPRINTF("-----------------------------------------------------------------------\n");
    DPRINTF("[ERROR] Roll number is not found\n");
    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.



```c
FIFO_status shift_buffer(u8 index,FIFO_buf_t* fifoPtr)
{
    for (int i=index; i<fifoPtr->count; i++)
    {
        buffer[i]=buffer[i+1];
    }
    DPRINTF("-----------------------------------------------------\n");
    DPRINTF("\t\tStudent 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.

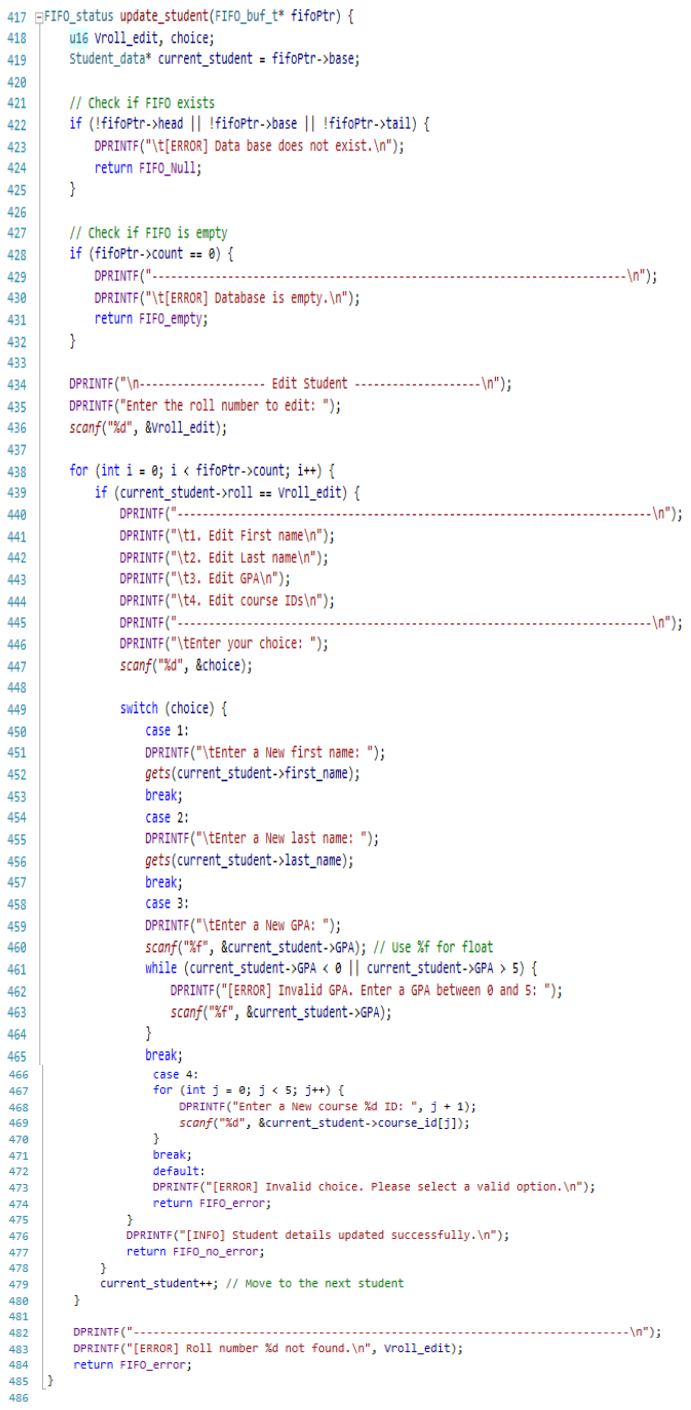
- it navigates on database based on studentroll number.

- 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 afterupdating.

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



```c
FIFO_status update_student(FIFO_buf_t* fifoPtr) {
    u16 Vroll_edit, choice;
    Student_data* current_student = fifoPtr->base;

    // Check if FIFO exists
    if (!fifoPtr->head || !fifoPtr->base || !fifoPtr->tail) {
        DPRINTF("\t[ERROR] Data base does not exist.\n");
        return FIFO_Null;
    }

    // Check if FIFO is empty
    if (fifoPtr->count == 0) {
        DPRINTF("--------------------------------------------------------------------------\n");
        DPRINTF("\t[ERROR] Database is empty.\n");
        return FIFO_empty;
    }

    DPRINTF("\n------------------- Edit Student -------------------\n");
    DPRINTF("Enter the roll number to edit: ");
    scanf("%d", &Vroll_edit);

    for (int i = 0; i < fifoPtr->count; i++) {
        if (current_student->roll == Vroll_edit) {
            DPRINTF("--------------------------------------------------------------------------\n");
            DPRINTF("\t1. Edit First name\n");
            DPRINTF("\t2. Edit Last name\n");
            DPRINTF("\t3. Edit GPA\n");
            DPRINTF("\t4. Edit course IDs\n");
            DPRINTF("--------------------------------------------------------------------------\n");
            DPRINTF("\tEnter your choice: ");
            scanf("%d", &choice);

            switch (choice) {
                case 1:
                DPRINTF("\tEnter a New first name: ");
                gets(current_student->first_name);
                break;
                case 2:
                DPRINTF("\tEnter a New last name: ");
                gets(current_student->last_name);
                break;
                case 3:
                DPRINTF("\tEnter a New GPA: ");
                scanf("%f", &current_student->GPA); // Use %f for float
                while (current_student->GPA < 0 || current_student->GPA > 5) {
                    DPRINTF("[ERROR] Invalid GPA. Enter a GPA between 0 and 5: ");
                    scanf("%f", &current_student->GPA);
                }
                break;
                 case 4:
                 for (int j = 0; j < 5; j++) {
                     DPRINTF("Enter a New course %d ID: ", j + 1);
                     scanf("%d", &current_student->course_id[j]);
                 }
                 break;
                 default:
                 DPRINTF("[ERROR] Invalid choice. Please select a valid option.\n");
                 return FIFO_error;
            }
            DPRINTF("[INFO] Student details updated successfully.\n");
            return FIFO_no_error;
        }
        current_student++; // Move to the next student
    }

    DPRINTF("--------------------------------------------------------------------------\n");
    DPRINTF("[ERROR] Roll number %d not found.\n", Vroll_edit);
    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.

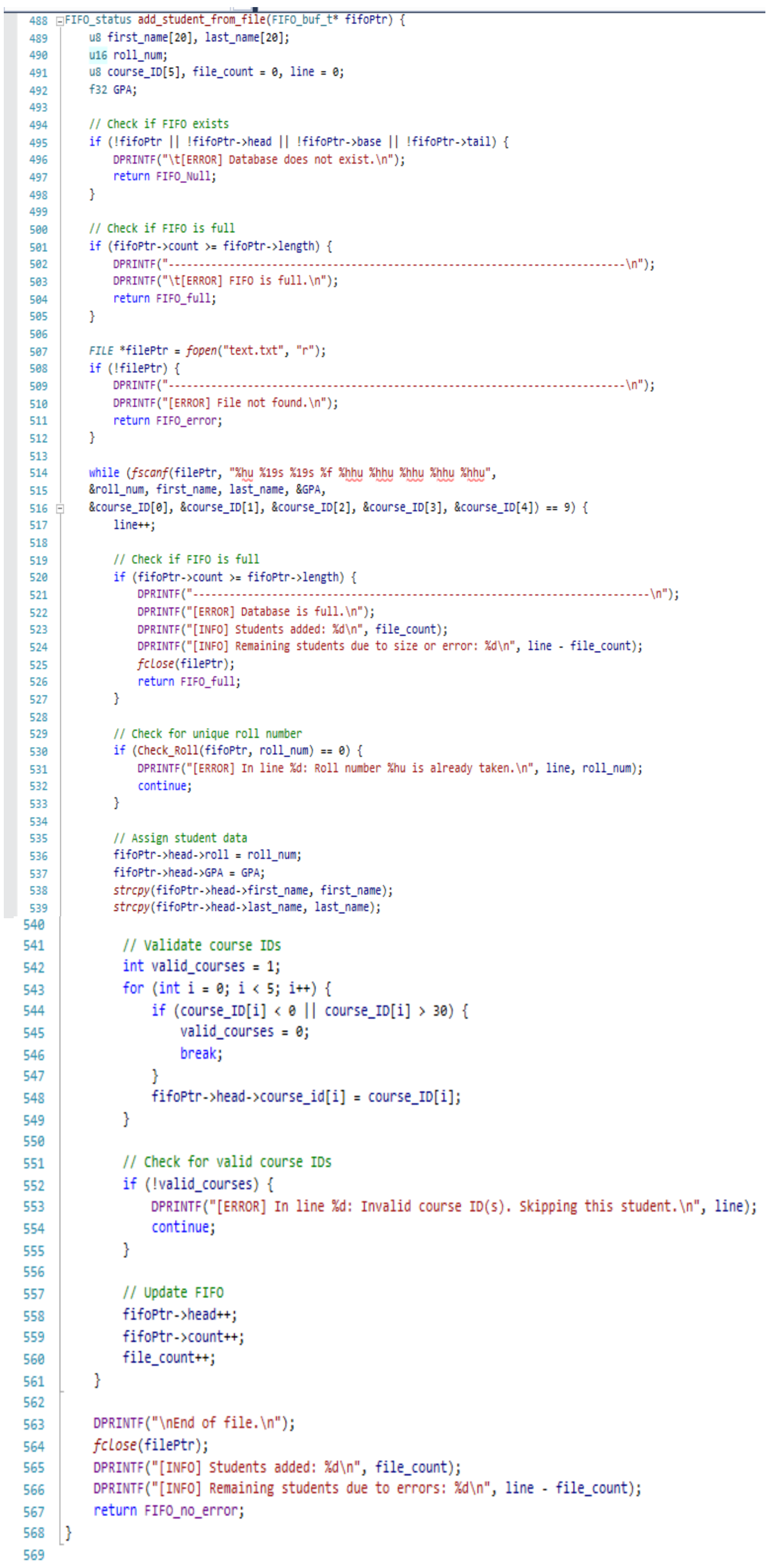
- if any student has non-valid course id, a message gets printed and this student is skipped.

-if buffer size reaches the full size

The function will stop adding and

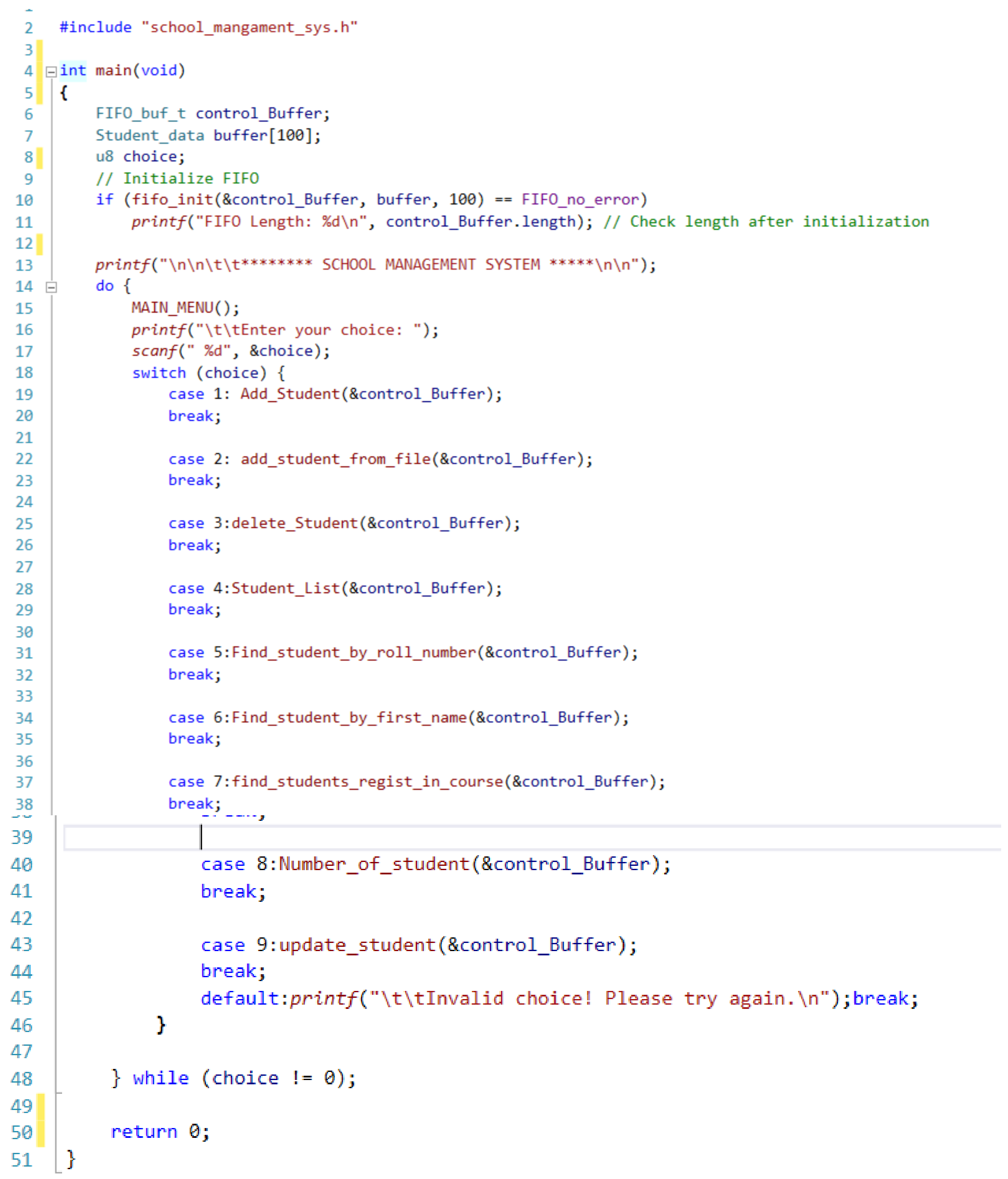
A message gets printed with the number of students added and the remaining students.

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



```c
FIFO_status add_student_from_file(FIFO_buf_t* fifoPtr) {
    u8 first_name[20], last_name[20];
    u16 roll_num;
    u8 course_ID[5], file_count = 0, line = 0;
    f32 GPA;

    // Check if FIFO exists
    if (!fifoPtr || !fifoPtr->head || !fifoPtr->base || !fifoPtr->tail) {
        DPRINTF("\t[ERROR] Database does not exist.\n");
        return FIFO_Null;
    }

    // Check if FIFO is full
    if (fifoPtr->count >= fifoPtr->length) {
        DPRINTF("------------------------------------------------------------------------\n");
        DPRINTF("\t[ERROR] FIFO is full.\n");
        return FIFO_full;
    }

    FILE *filePtr = fopen("text.txt", "r");
    if (!filePtr) {
        DPRINTF("------------------------------------------------------------------------\n");
        DPRINTF("[ERROR] File not found.\n");
        return FIFO_error;
    }

    while (fscanf(filePtr, "%hu %19s %19s %f %hhu %hhu %hhu %hhu %hhu",
    &roll_num, first_name, last_name, &GPA,
    &course_ID[0], &course_ID[1], &course_ID[2], &course_ID[3], &course_ID[4]) == 9) {
        line++;

        // Check if FIFO is full
        if (fifoPtr->count >= fifoPtr->length) {
            DPRINTF("------------------------------------------------------------------------\n");
            DPRINTF("[ERROR] Database is full.\n");
            DPRINTF("[INFO] Students added: %d\n", file_count);
            DPRINTF("[INFO] Remaining students due to size or error: %d\n", line - file_count);
            fclose(filePtr);
            return FIFO_full;
        }

        // Check for unique roll number
        if (Check_Roll(fifoPtr, roll_num) == 0) {
            DPRINTF("[ERROR] In line %d: Roll number %hu is already taken.\n", line, roll_num);
            continue;
        }

        // Assign student data
        fifoPtr->head->roll = roll_num;
        fifoPtr->head->GPA = GPA;
        strcpy(fifoPtr->head->first_name, first_name);
        strcpy(fifoPtr->head->last_name, last_name);

        // Validate course IDs
        int valid_courses = 1;
        for (int i = 0; i < 5; i++) {
            if (course_ID[i] < 0 || course_ID[i] > 30) {
                valid_courses = 0;
                break;
            }
            fifoPtr->head->course_id[i] = course_ID[i];
        }

        // Check for valid course IDs
        if (!valid_courses) {
            DPRINTF("[ERROR] In line %d: Invalid course ID(s). Skipping this student.\n", line);
            continue;
        }

        // Update FIFO
        fifoPtr->head++;
        fifoPtr->count++;
        file_count++;
    }

    DPRINTF("\nEnd of file.\n");
    fclose(filePtr);
    DPRINTF("[INFO] Students added: %d\n", file_count);
    DPRINTF("[INFO] Remaining students due to errors: %d\n", line - file_count);
    return FIFO_no_error;
}
```

14- main function

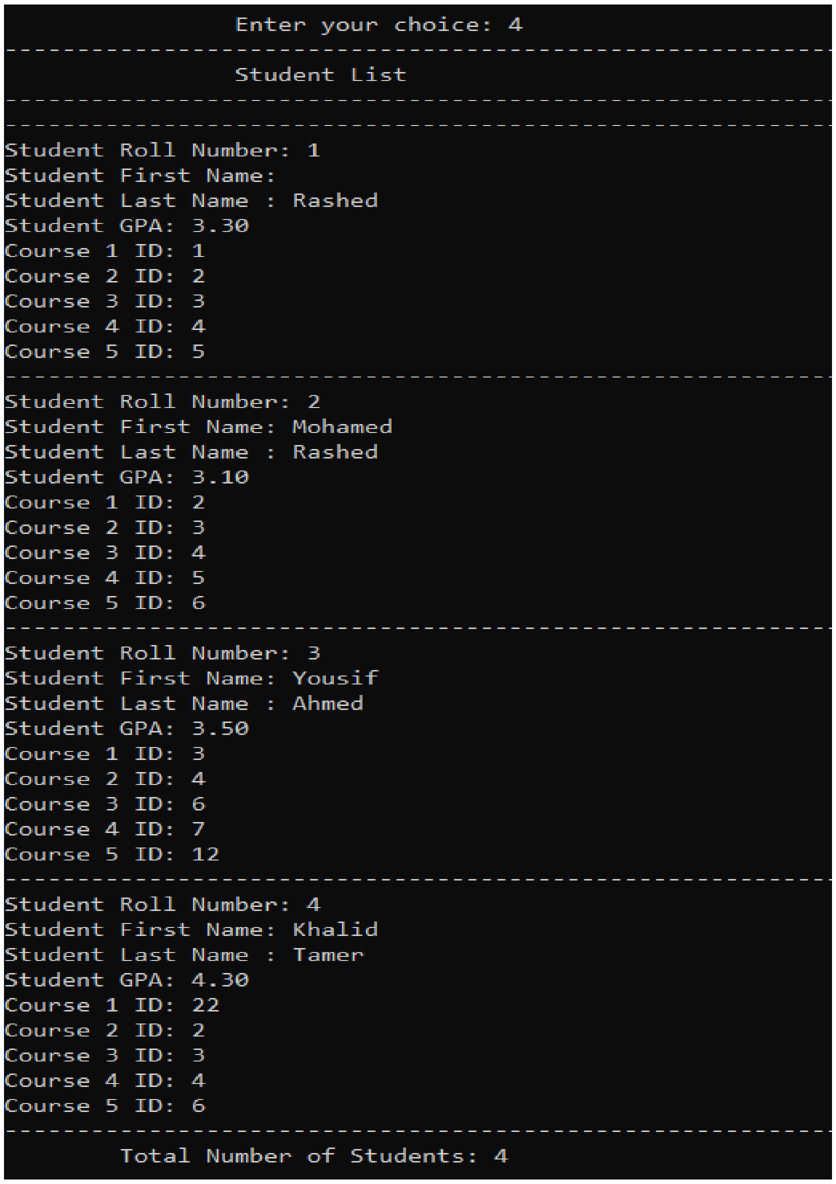


```c
#include "school_mangament_sys.h"

int main(void)
{
    FIFO_buf_t control_Buffer;
    Student_data buffer[100];
    u8 choice;
    // Initialize FIFO
    if (fifo_init(&control_Buffer, buffer, 100) == FIFO_no_error)
        printf("FIFO Length: %d\n", control_Buffer.length); // Check length after initialization

    printf("\n\n\t\t******** SCHOOL MANAGEMENT SYSTEM *****\n\n");
    do {
        MAIN_MENU();
        printf("\t\tEnter your choice: ");
        scanf(" %d", &choice);
        switch (choice) {
            case 1: Add_Student(&control_Buffer);
            break;

            case 2: add_student_from_file(&control_Buffer);
            break;

            case 3:delete_Student(&control_Buffer);
            break;

            case 4:Student_List(&control_Buffer);
            break;

            case 5:Find_student_by_roll_number(&control_Buffer);
            break;

            case 6:Find_student_by_first_name(&control_Buffer);
            break;

            case 7:find_students_regist_in_course(&control_Buffer);
            break;

            case 8:Number_of_student(&control_Buffer);
            break;

            case 9:update_student(&control_Buffer);
            break;
            default:printf("\t\tInvalid choice! Please try again.\n");break;
        }

    } while (choice != 0);

    return 0;
}
```

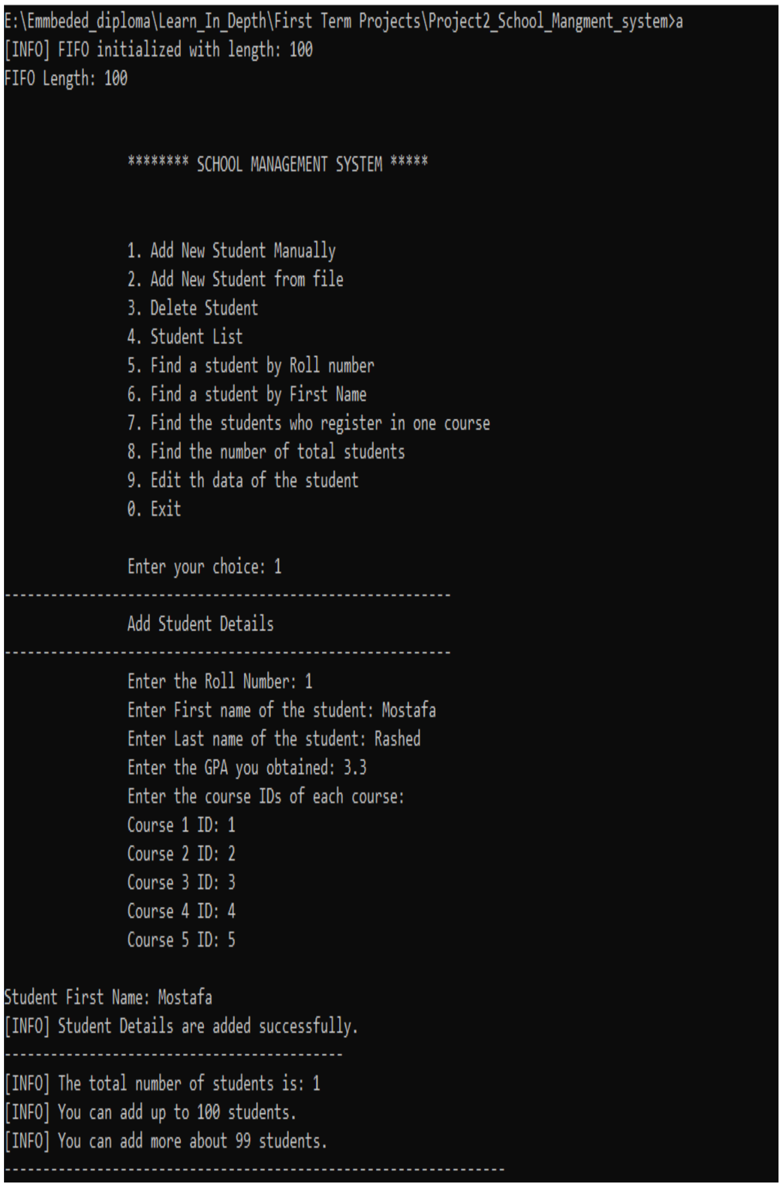
Testing code logic

1- Adding students manually

We will add 4 students with the same way.



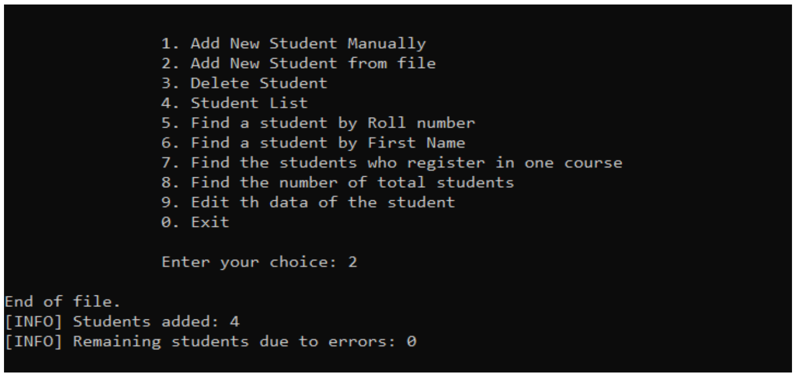
```
            Enter your choice: 4
----------------------------------------------------------
            Student List
----------------------------------------------------------
----------------------------------------------------------
Student Roll Number: 1
Student First Name:
Student Last Name : Rashed
Student GPA: 3.30
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: Mohamed
Student Last Name : Rashed
Student GPA: 3.10
Course 1 ID: 2
Course 2 ID: 3
Course 3 ID: 4
Course 4 ID: 5
Course 5 ID: 6
----------------------------------------------------------
Student Roll Number: 3
Student First Name: Yousif
Student Last Name : Ahmed
Student GPA: 3.50
Course 1 ID: 3
Course 2 ID: 4
Course 3 ID: 6
Course 4 ID: 7
Course 5 ID: 12
----------------------------------------------------------
Student Roll Number: 4
Student First Name: Khalid
Student Last Name : Tamer
Student GPA: 4.30
Course 1 ID: 22
Course 2 ID: 2
Course 3 ID: 3
Course 4 ID: 4
Course 5 ID: 6
----------------------------------------------------------
      Total Number of Students: 4
```



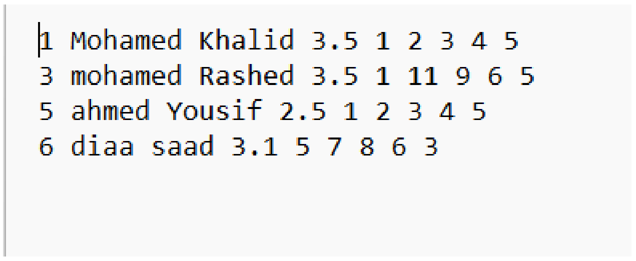
```
E:\Emmbeded_diploma\Learn_In_Depth\First Term Projects\Project2_School_Mangment_system>a
[INFO] FIFO initialized with length: 100
FIFO Length: 100


        ******** SCHOOL MANAGEMENT SYSTEM *****


        1. Add New Student Manually
        2. Add New Student from file
        3. Delete Student
        4. Student List
        5. Find a student by Roll number
        6. Find a student by First Name
        7. Find the students who register in one course
        8. Find the number of total students
        9. Edit th data of the student
        0. Exit

        Enter your choice: 1
------------------------------------------------------
        Add Student Details
------------------------------------------------------
        Enter the Roll Number: 1
        Enter First name of the student: Mostafa
        Enter Last name of the student: Rashed
        Enter the GPA you obtained: 3.3
        Enter the course IDs of each course:
        Course 1 ID: 1
        Course 2 ID: 2
        Course 3 ID: 3
        Course 4 ID: 4
        Course 5 ID: 5

Student First Name: Mostafa
[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.
------------------------------------------------------------
```

2-Adding from text file

text.txt

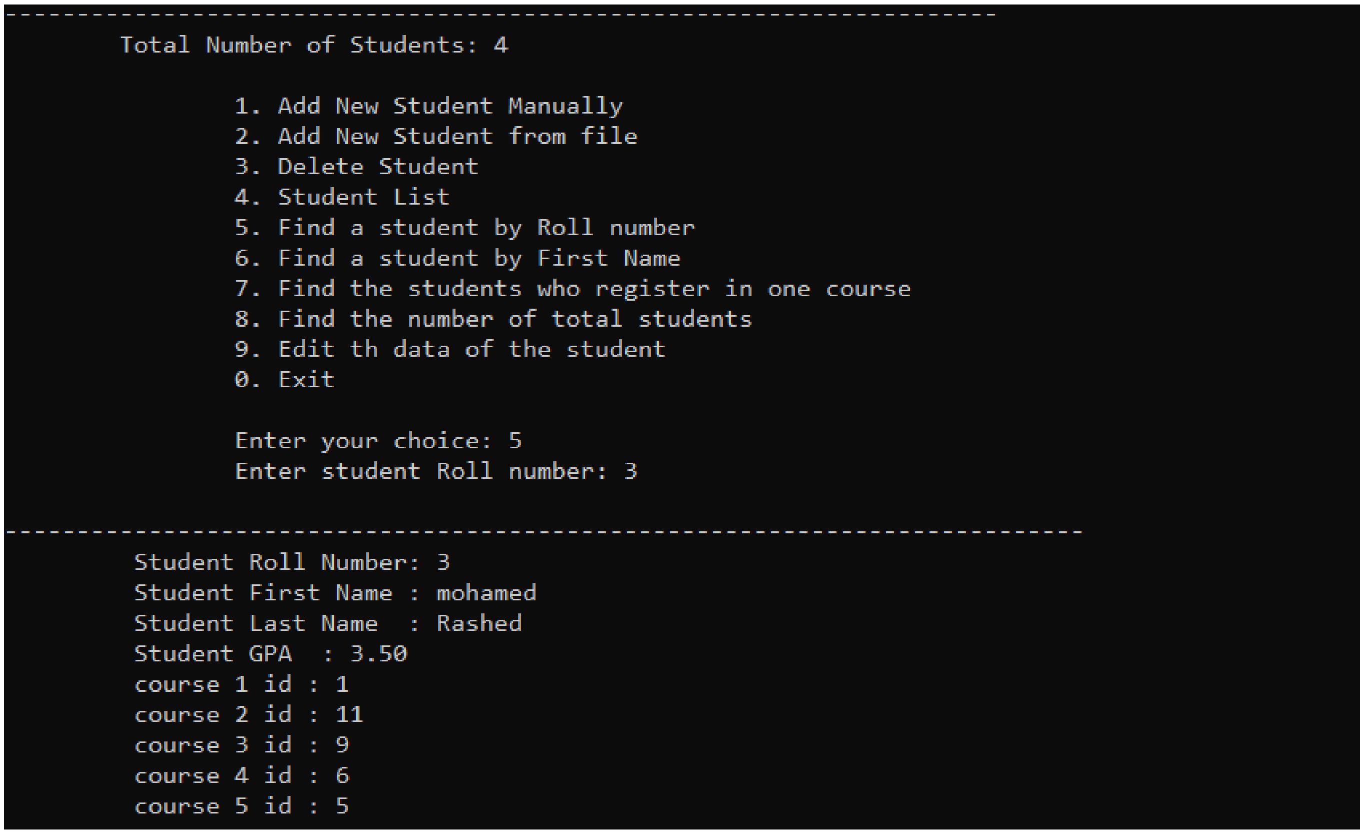


```
        1. Add New Student Manually
        2. Add New Student from file
        3. Delete Student
        4. Student List
        5. Find a student by Roll number
        6. Find a student by First Name
        7. Find the students who register in one course
        8. Find the number of total students
        9. Edit th data of the student
        0. Exit

        Enter your choice: 2

End of file.
[INFO] Students added: 4
[INFO] Remaining students due to errors: 0
```



```
1 Mohamed Khalid 3.5 1 2 3 4 5
3 mohamed Rashed 3.5 1 11 9 6 5
5 ahmed Yousif 2.5 1 2 3 4 5
6 diaa saad 3.1 5 7 8 6 3
```

3-get by roll number

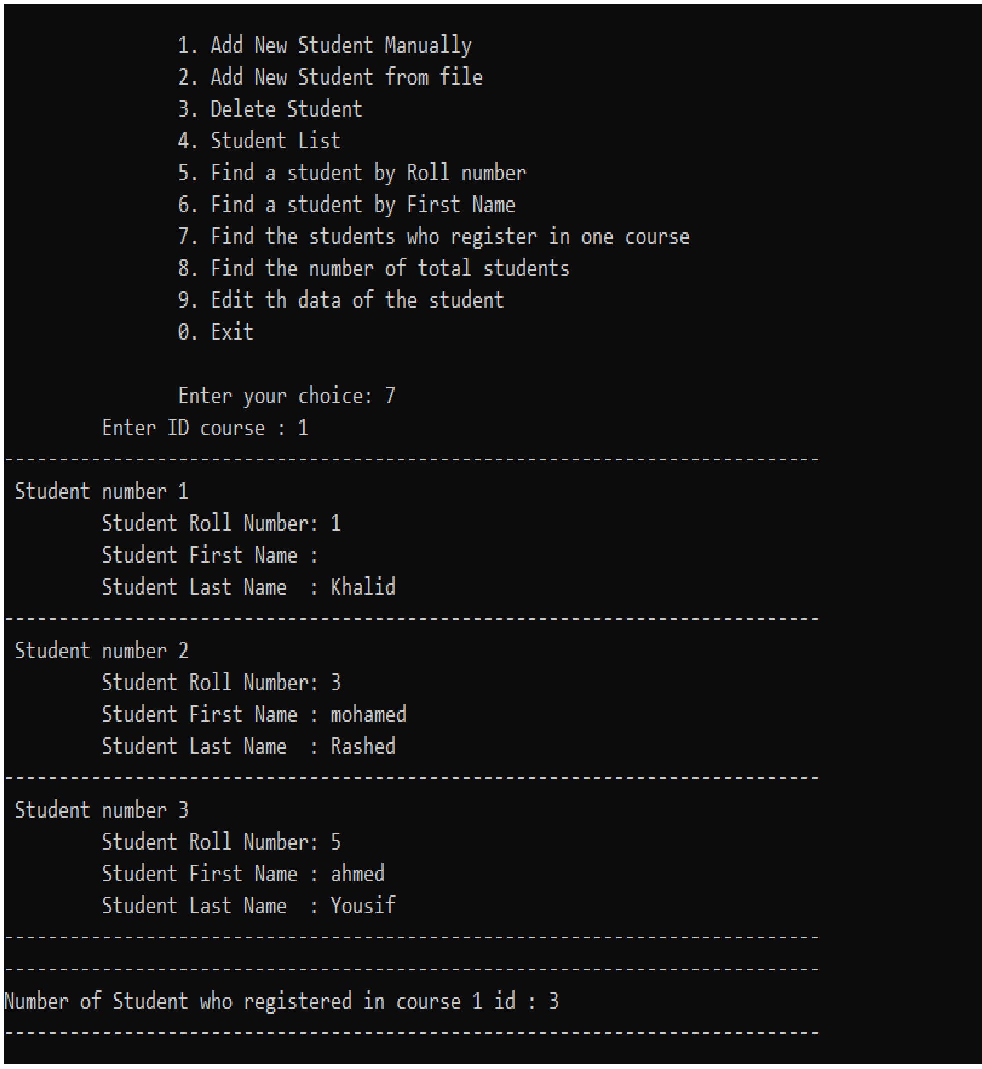


```
---------------------------------------------------------------------
       Total Number of Students: 4

              1. Add New Student Manually
              2. Add New Student from file
              3. Delete Student
              4. Student List
              5. Find a student by Roll number
              6. Find a student by First Name
              7. Find the students who register in one course
              8. Find the number of total students
              9. Edit th data of the student
              0. Exit

              Enter your choice: 5
              Enter student Roll number: 3

---------------------------------------------------------------------------
        Student Roll Number: 3
        Student First Name : mohamed
        Student Last Name  : Rashed
        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 registered in a specific course id

for ex course id: 2

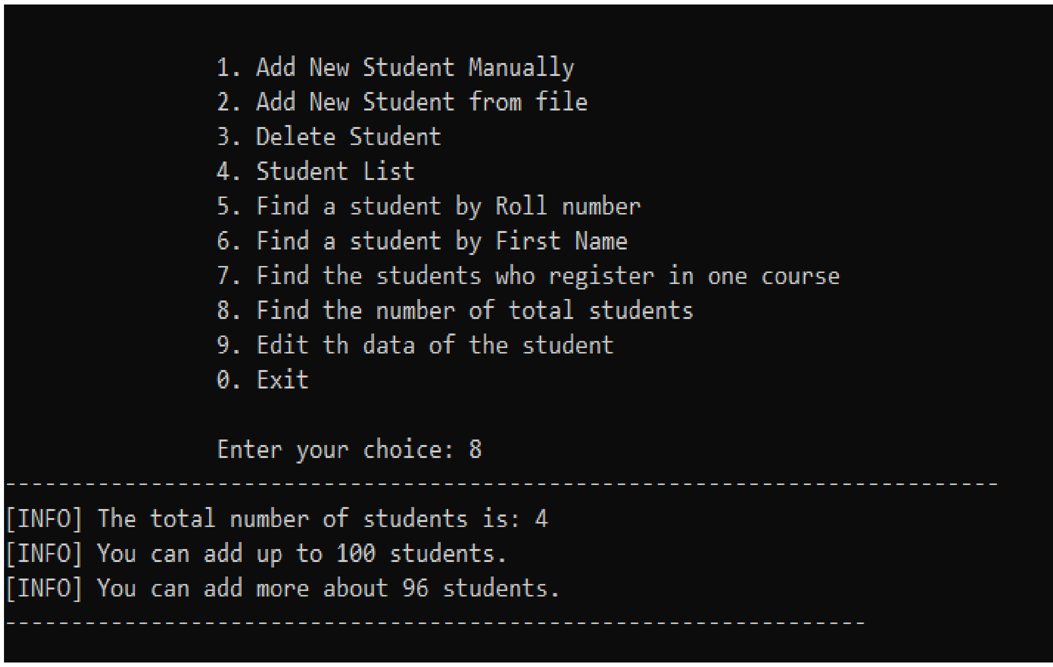
we have 3 students from 4 registered incourse id 1



```
                1. Add New Student Manually
                2. Add New Student from file
                3. Delete Student
                4. Student List
                5. Find a student by Roll number
                6. Find a student by First Name
                7. Find the students who register in one course
                8. Find the number of total students
                9. Edit th data of the student
                0. Exit

                Enter your choice: 7
         Enter ID course : 1
---------------------------------------------------------------------------
 Student number 1
         Student Roll Number: 1
         Student First Name :
         Student Last Name  : Khalid
---------------------------------------------------------------------------
 Student number 2
         Student Roll Number: 3
         Student First Name : mohamed
         Student Last Name  : Rashed
---------------------------------------------------------------------------
 Student number 3
         Student Roll Number: 5
         Student First Name : ahmed
         Student Last Name  : Yousif
---------------------------------------------------------------------------
---------------------------------------------------------------------------
Number of Student who registered in course 1 id : 3
---------------------------------------------------------------------------
```

5- Get total number of students

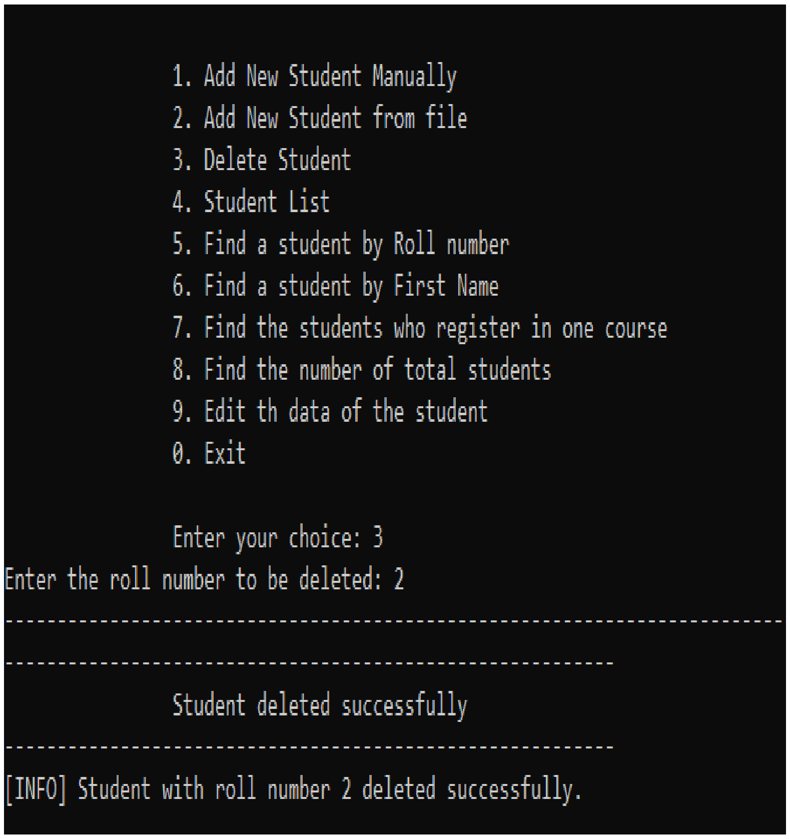
we have 4 students



```
                1. Add New Student Manually
                2. Add New Student from file
                3. Delete Student
                4. Student List
                5. Find a student by Roll number
                6. Find a student by First Name
                7. Find the students who register in one course
                8. Find the number of total students
                9. Edit th data of the student
                0. Exit

                Enter your choice: 8
---------------------------------------------------------------------------
[INFO] The total number of students is: 4
[INFO] You can add up to 100 students.
[INFO] You can add more about 96 students.
-----------------------------------------------------------------
```

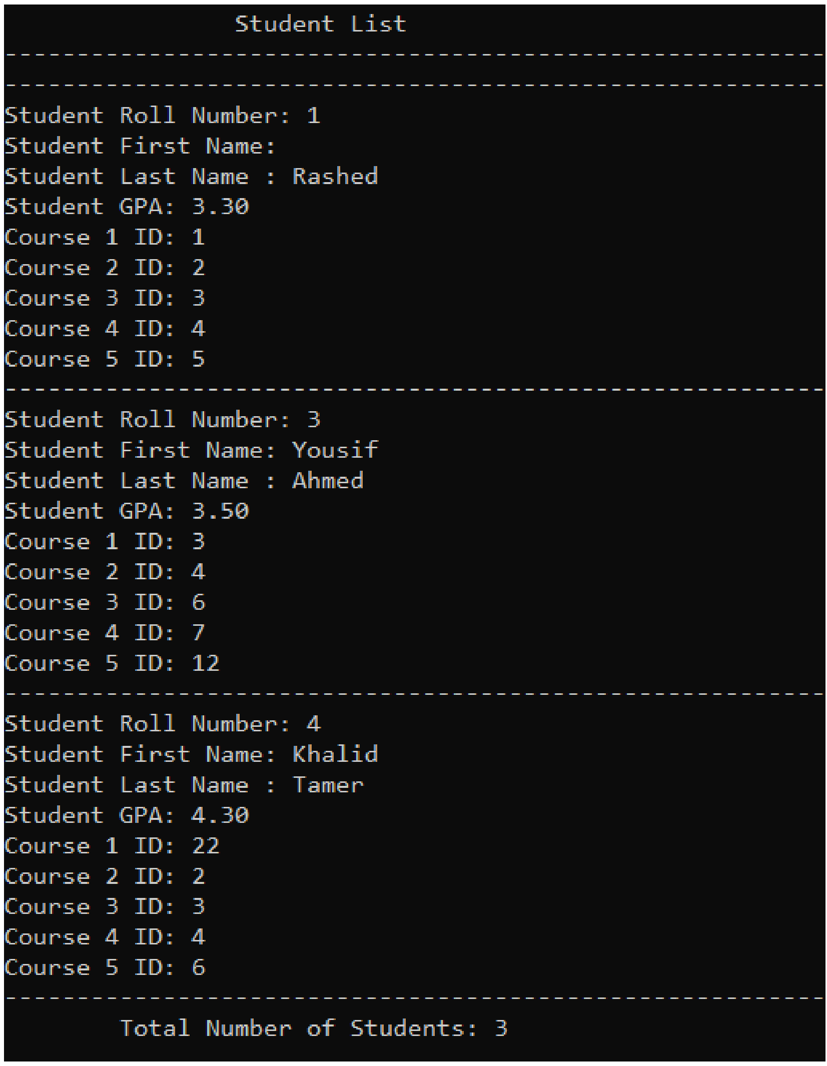
6- Delete student by roll number

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



```
            1. Add New Student Manually
            2. Add New Student from file
            3. Delete Student
            4. Student List
            5. Find a student by Roll number
            6. Find a student by First Name
            7. Find the students who register in one course
            8. Find the number of total students
            9. Edit th data of the student
            0. Exit

            Enter your choice: 3
Enter the roll number to be deleted: 2
-------------------------------------------------------------------------
-------------------------------------------------------
            Student deleted successfully
-------------------------------------------------------
[INFO] Student with roll number 2 deleted successfully.
```

Show all students after delete

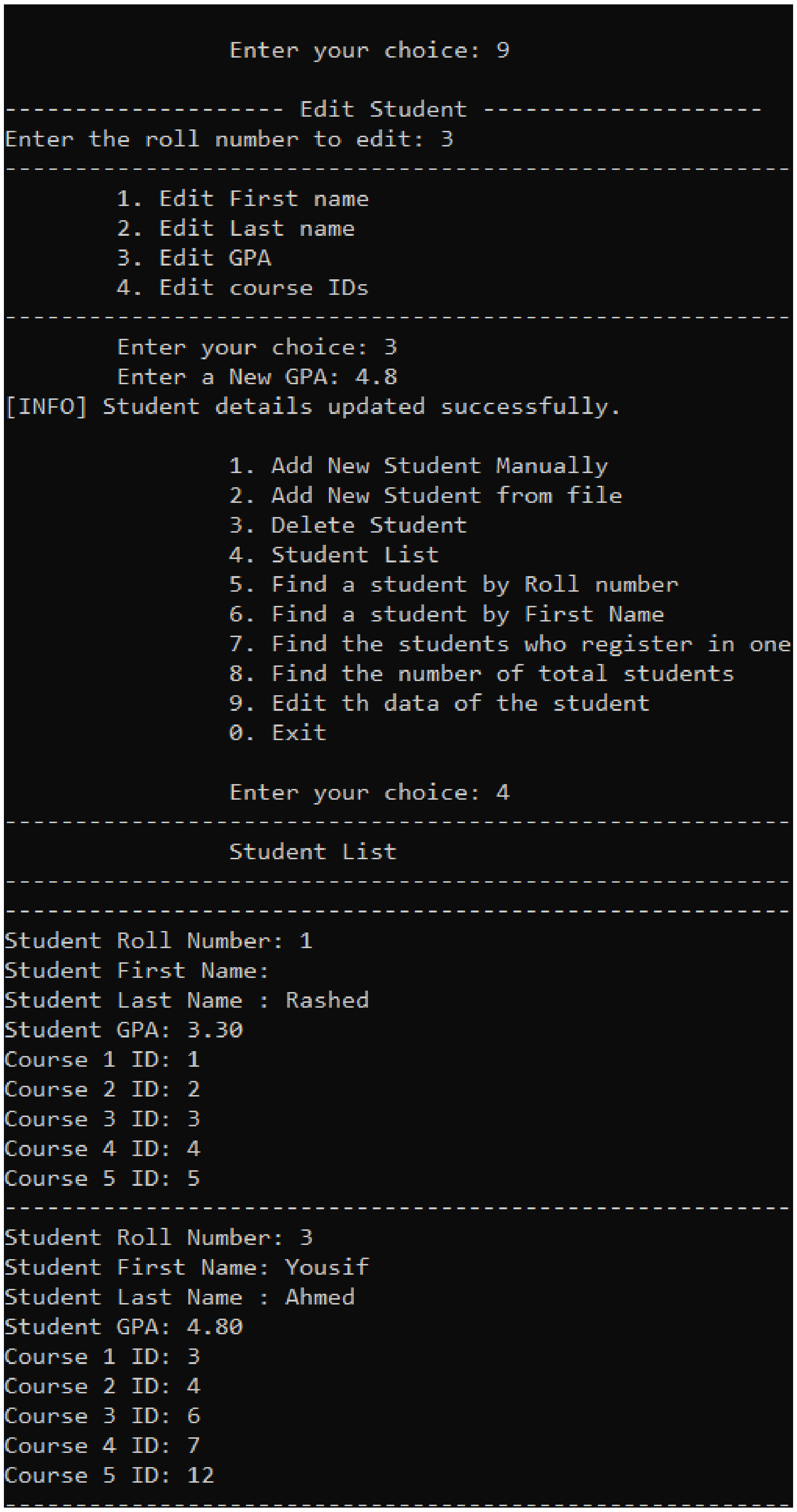


```
                Student List
---------------------------------------------------------
---------------------------------------------------------
Student Roll Number: 1
Student First Name:
Student Last Name : Rashed
Student GPA: 3.30
Course 1 ID: 1
Course 2 ID: 2
Course 3 ID: 3
Course 4 ID: 4
Course 5 ID: 5
---------------------------------------------------------
Student Roll Number: 3
Student First Name: Yousif
Student Last Name : Ahmed
Student GPA: 3.50
Course 1 ID: 3
Course 2 ID: 4
Course 3 ID: 6
Course 4 ID: 7
Course 5 ID: 12
---------------------------------------------------------
Student Roll Number: 4
Student First Name: Khalid
Student Last Name : Tamer
Student GPA: 4.30
Course 1 ID: 22
Course 2 ID: 2
Course 3 ID: 3
Course 4 ID: 4
Course 5 ID: 6
---------------------------------------------------------
        Total Number of Students: 3
```

7- Update specific data of a student

For example:

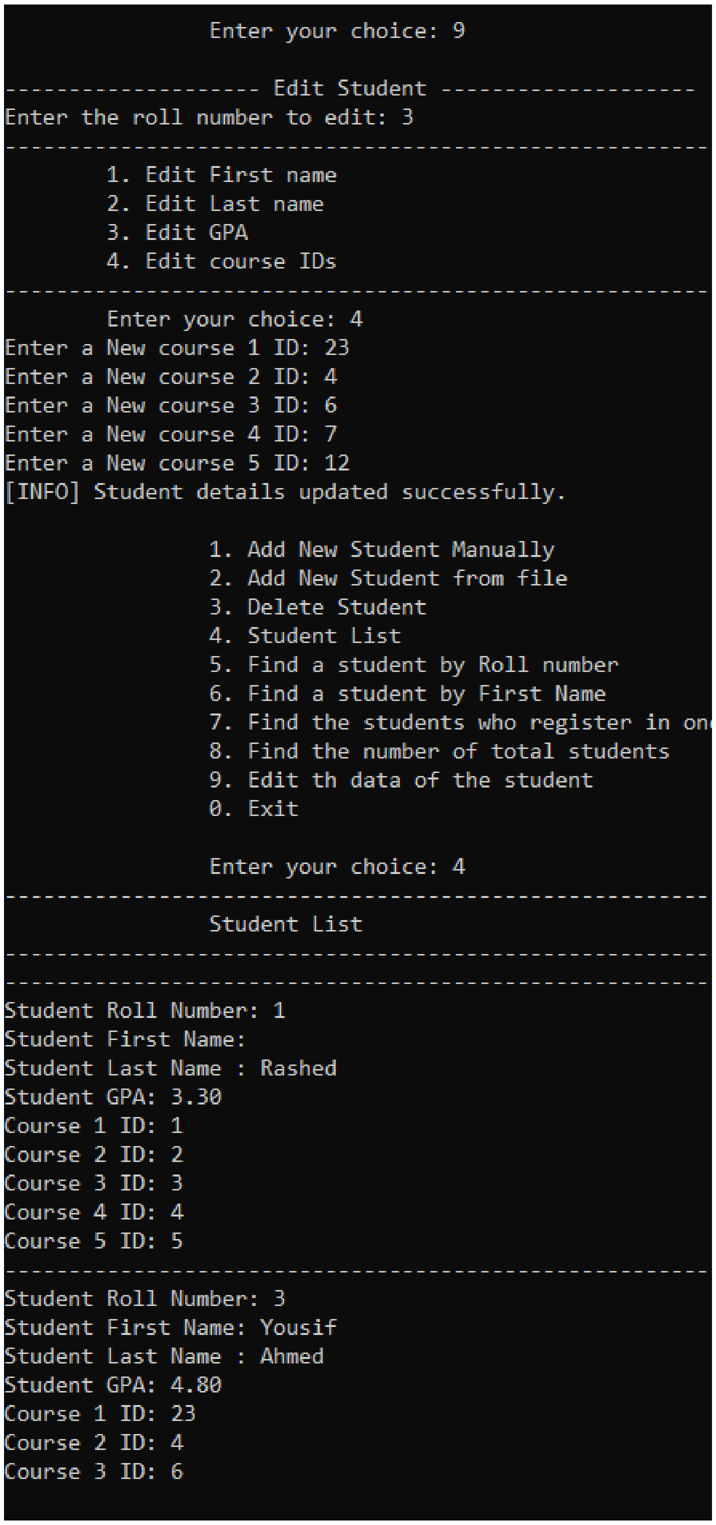
update Yousif's GPA



```
                Enter your choice: 9

-------------------- Edit Student --------------------
Enter the roll number to edit: 3
--------------------------------------------------------
        1. Edit First name
        2. Edit Last name
        3. Edit GPA
        4. Edit course IDs
--------------------------------------------------------
        Enter your choice: 3
        Enter a New GPA: 4.8
[INFO] Student details updated successfully.

                1. Add New Student Manually
                2. Add New Student from file
                3. Delete Student
                4. Student List
                5. Find a student by Roll number
                6. Find a student by First Name
                7. Find the students who register in one
                8. Find the number of total students
                9. Edit th data of the student
                0. Exit

                Enter your choice: 4
--------------------------------------------------------
                Student List
--------------------------------------------------------
--------------------------------------------------------
Student Roll Number: 1
Student First Name:
Student Last Name : Rashed
Student GPA: 3.30
Course 1 ID: 1
Course 2 ID: 2
Course 3 ID: 3
Course 4 ID: 4
Course 5 ID: 5
--------------------------------------------------------
Student Roll Number: 3
Student First Name: Yousif
Student Last Name : Ahmed
Student GPA: 4.80
Course 1 ID: 3
Course 2 ID: 4
Course 3 ID: 6
Course 4 ID: 7
Course 5 ID: 12
--------------------------------------------------------
```

update Yousif courses id

:



```
                Enter your choice: 9

-------------------- Edit Student --------------------
Enter the roll number to edit: 3
-------------------------------------------------------
        1. Edit First name
        2. Edit Last name
        3. Edit GPA
        4. Edit course IDs
-------------------------------------------------------
        Enter your choice: 4
Enter a New course 1 ID: 23
Enter a New course 2 ID: 4
Enter a New course 3 ID: 6
Enter a New course 4 ID: 7
Enter a New course 5 ID: 12
[INFO] Student details updated successfully.

                1. Add New Student Manually
                2. Add New Student from file
                3. Delete Student
                4. Student List
                5. Find a student by Roll number
                6. Find a student by First Name
                7. Find the students who register in on
                8. Find the number of total students
                9. Edit th data of the student
                0. Exit

                Enter your choice: 4
-------------------------------------------------------
                Student List
-------------------------------------------------------
-------------------------------------------------------
Student Roll Number: 1
Student First Name:
Student Last Name : Rashed
Student GPA: 3.30
Course 1 ID: 1
Course 2 ID: 2
Course 3 ID: 3
Course 4 ID: 4
Course 5 ID: 5
-------------------------------------------------------
Student Roll Number: 3
Student First Name: Yousif
Student Last Name : Ahmed
Student GPA: 4.80
Course 1 ID: 23
Course 2 ID: 4
Course 3 ID: 6
```