# Student Management System

Mastering Embedded Systems Diploma
www.learnindepth.com
First Term (Final Project 1)

Eng. Michel Adel Fouad
My Profile: https://www.learn-in-depth.com/online-diploma/meshoadel2018%40gmail.com

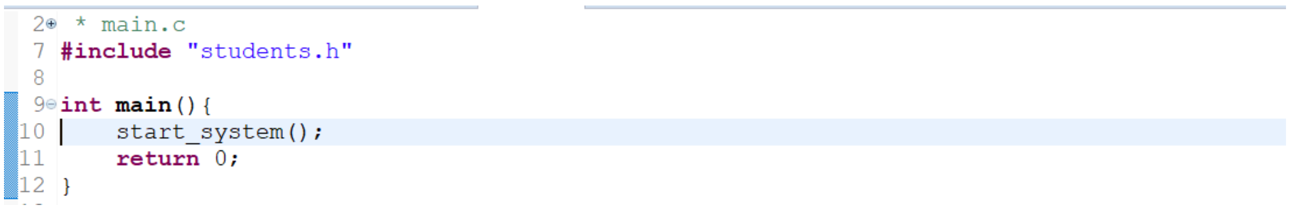
## Project Description:

It is required to make a student management system using FIFO and add the following options for the user to use:

1. Add the students 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
6. Find the total number of students
7. Delete the students' details by roll number
8. Update the student's details by roll number
9. Show all information

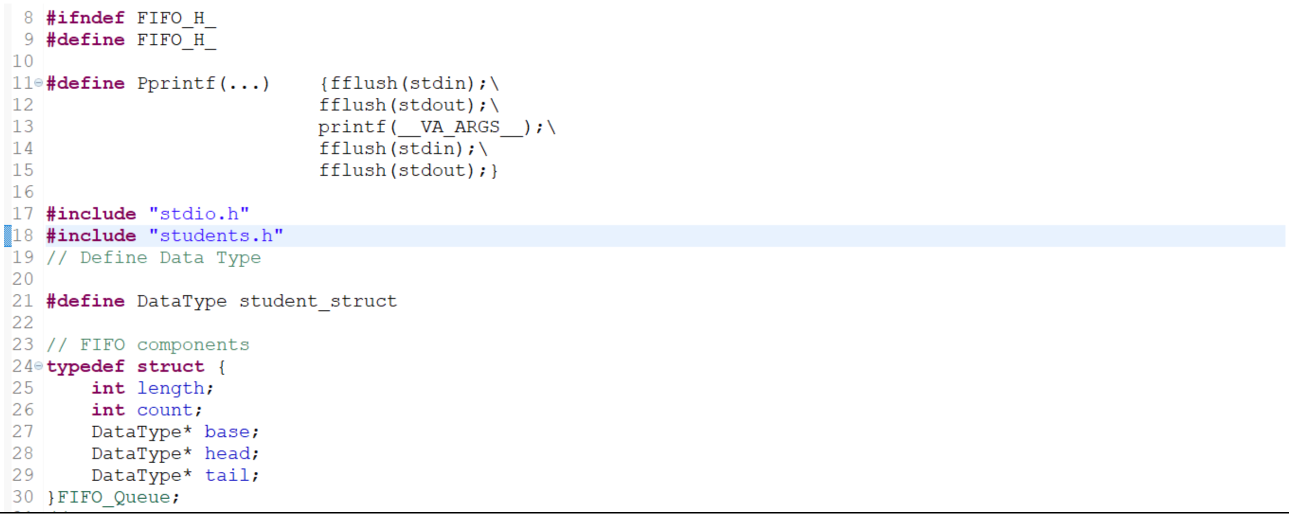
## Code:

In main.c file we just call the system to start

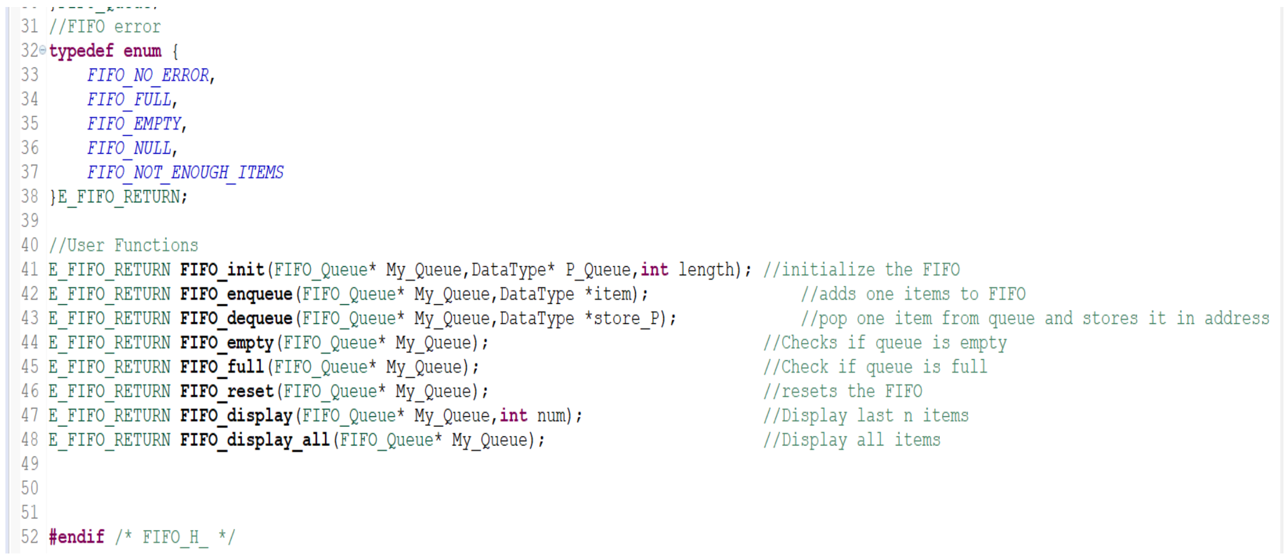


```c
 * main.c
#include "students.h"

int main(){
	start_system();
	return 0;
}
```

Then we use the FIFO driver we implemented before

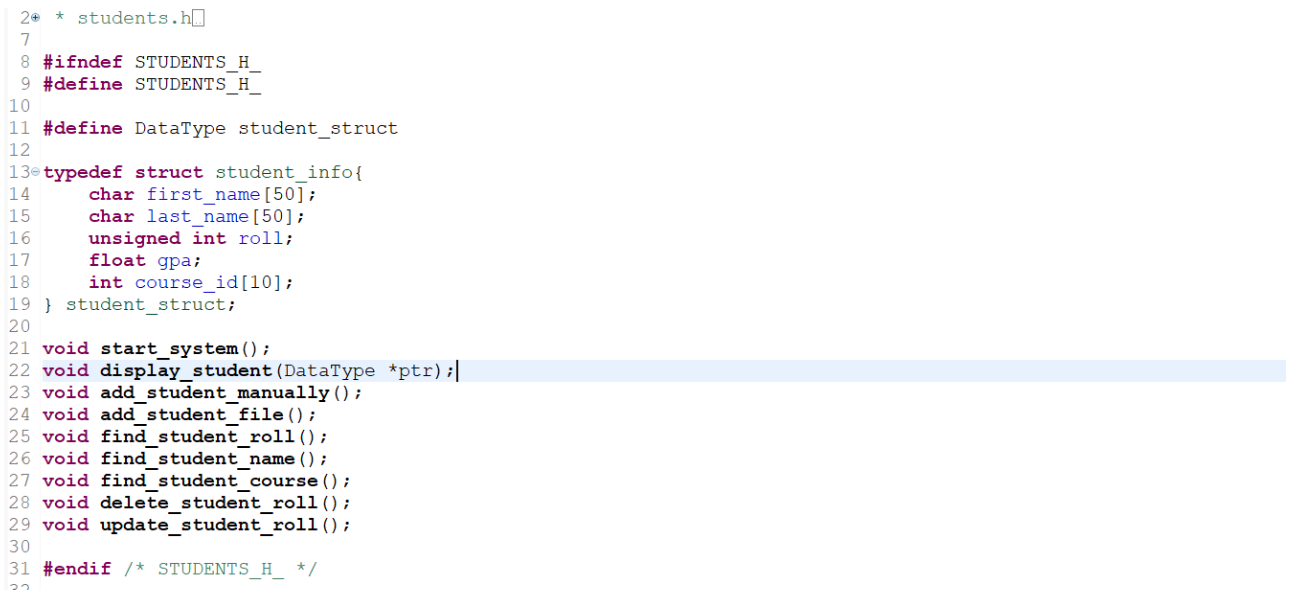


```c
#ifndef FIFO_H_
#define FIFO_H_

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

#include "stdio.h"
#include "students.h"
// Define Data Type

#define DataType student_struct

// FIFO components
typedef struct {
	int length;
	int count;
	DataType* base;
	DataType* head;
	DataType* tail;
}FIFO_Queue;
```



```c
//FIFO error
typedef enum {
    FIFO_NO_ERROR,
    FIFO_FULL,
    FIFO_EMPTY,
    FIFO_NULL,
    FIFO_NOT_ENOUGH_ITEMS
}E_FIFO_RETURN;

//User Functions
E_FIFO_RETURN FIFO_init(FIFO_Queue* My_Queue,DataType* P_Queue,int length); //initialize the FIFO
E_FIFO_RETURN FIFO_enqueue(FIFO_Queue* My_Queue,DataType *item);                //adds one items to FIFO
E_FIFO_RETURN FIFO_dequeue(FIFO_Queue* My_Queue,DataType *store_P);             //pop one item from queue and stores it in address
E_FIFO_RETURN FIFO_empty(FIFO_Queue* My_Queue);                             //Checks if queue is empty
E_FIFO_RETURN FIFO_full(FIFO_Queue* My_Queue);                              //Check if queue is full
E_FIFO_RETURN FIFO_reset(FIFO_Queue* My_Queue);                             //resets the FIFO
E_FIFO_RETURN FIFO_display(FIFO_Queue* My_Queue,int num);                   //Display last n items
E_FIFO_RETURN FIFO_display_all(FIFO_Queue* My_Queue);                       //Display all items



#endif /* FIFO_H_ */
```

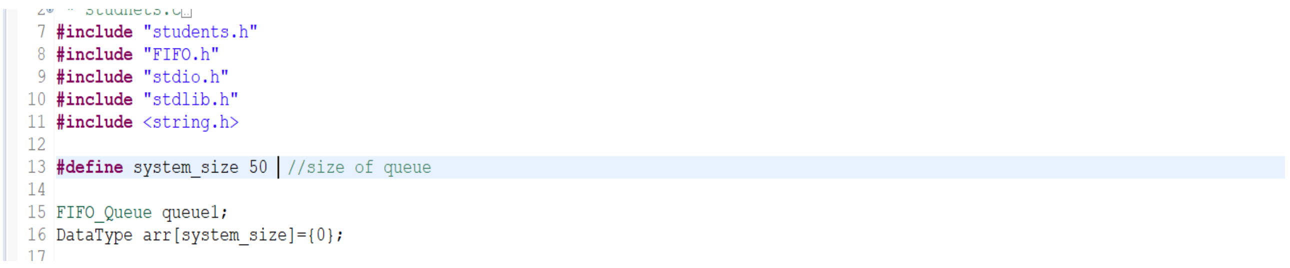
## The System driver (students.h file):



```c
 * students.h

#ifndef STUDENTS_H_
#define STUDENTS_H_

#define DataType student_struct

typedef struct student_info{
    char first_name[50];
    char last_name[50];
    unsigned int roll;
    float gpa;
    int course_id[10];
} student_struct;

void start_system();
void display_student(DataType *ptr);
void add_student_manually();
void add_student_file();
void find_student_roll();
void find_student_name();
void find_student_course();
void delete_student_roll();
void update_student_roll();

#endif /* STUDENTS_H_ */
```

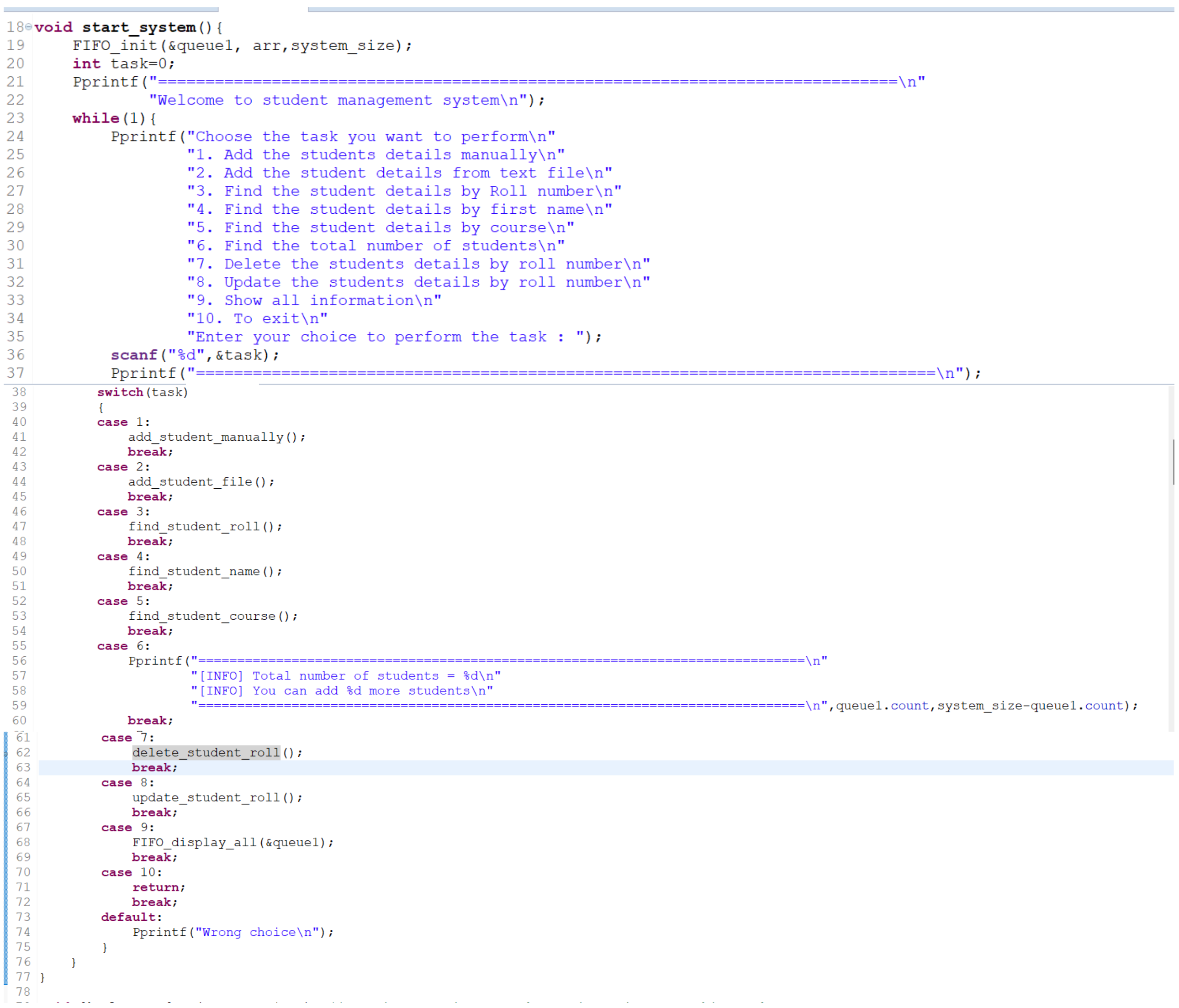
## The System driver (students.c file):

We start with includes macros and queue initialization:



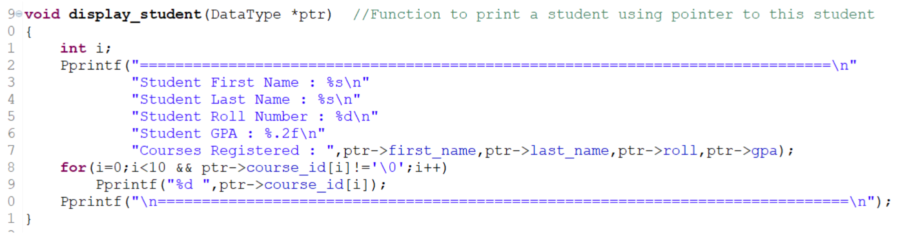
```c
#include "students.h"
#include "FIFO.h"
#include "stdio.h"
#include "stdlib.h"
#include <string.h>

#define system_size 50 //size of queue

FIFO_Queue queue1;
DataType arr[system_size]={0};
```

Then function to start the system:



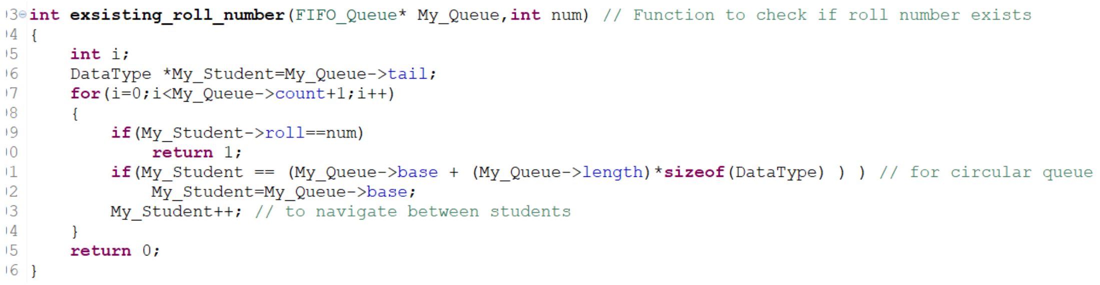
```c
void start_system(){
    FIFO_init(&queue1, arr,system_size);
    int task=0;
    Pprintf("================================================================================\n"
            "Welcome to student management system\n");
    while(1){
        Pprintf("Choose the task you want to perform\n"
                "1. Add the students details manually\n"
                "2. Add the student details from text file\n"
                "3. Find the student details by Roll number\n"
                "4. Find the student details by first name\n"
                "5. Find the student details by course\n"
                "6. Find the total number of students\n"
                "7. Delete the students details by roll number\n"
                "8. Update the students details by roll number\n"
                "9. Show all information\n"
                "10. To exit\n"
                "Enter your choice to perform the task : ");
        scanf("%d",&task);
        Pprintf("================================================================================\n");
        switch(task)
        {
        case 1:
            add_student_manually();
            break;
        case 2:
            add_student_file();
            break;
        case 3:
            find_student_roll();
            break;
        case 4:
            find_student_name();
            break;
        case 5:
            find_student_course();
            break;
        case 6:
            Pprintf("================================================================================\n"
                    "[INFO] Total number of students = %d\n"
                    "[INFO] You can add %d more students\n"
                    "================================================================================\n",queue1.count,system_size-queue1.count);
            break;
        case 7:
            delete_student_roll();
            break;
        case 8:
            update_student_roll();
            break;
        case 9:
            FIFO_display_all(&queue1);
            break;
        case 10:
            return;
            break;
        default:
            Pprintf("Wrong choice\n");
        }
    }
}
```

## Function to display a student using a pointer to the student:



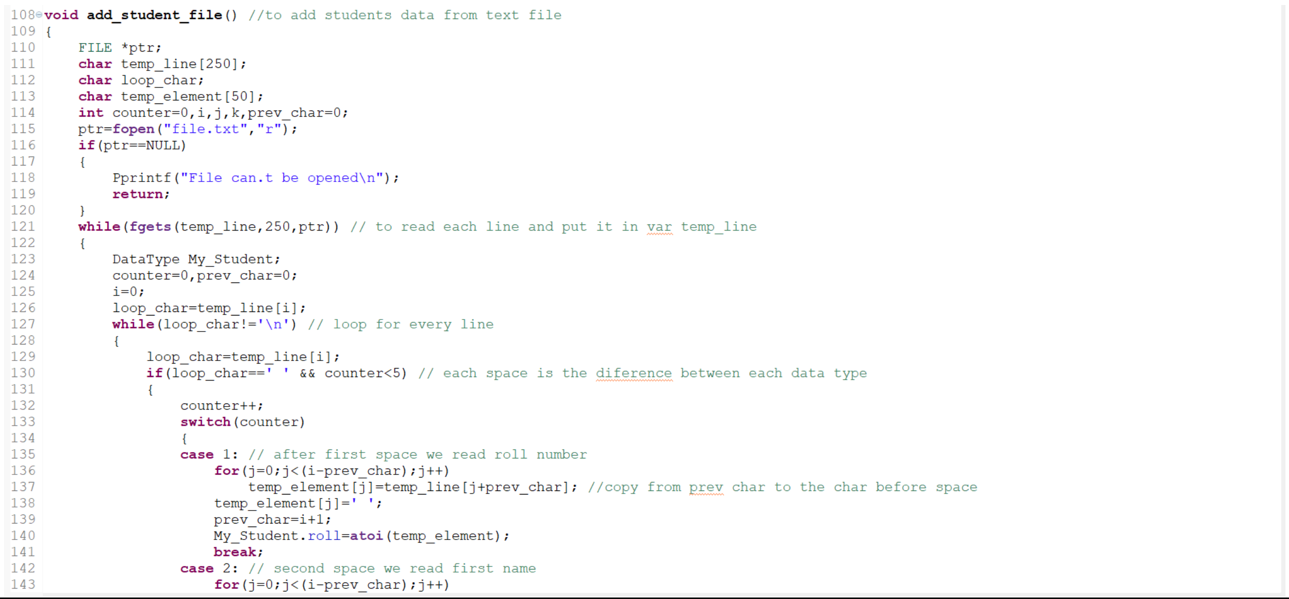
```c
void display_student(DataType *ptr)  //Function to print a student using pointer to this student
{
    int i;
    Pprintf("==============================================================================\n"
            "Student First Name : %s\n"
            "Student Last Name : %s\n"
            "Student Roll Number : %d\n"
            "Student GPA : %.2f\n"
            "Courses Registered : ",ptr->first_name,ptr->last_name,ptr->roll,ptr->gpa);
    for(i=0;i<10 && ptr->course_id[i]!='\0';i++)
        Pprintf("%d ",ptr->course_id[i]);
    Pprintf("\n==============================================================================\n");
}
```

## Function to check if a roll number exists in the system:

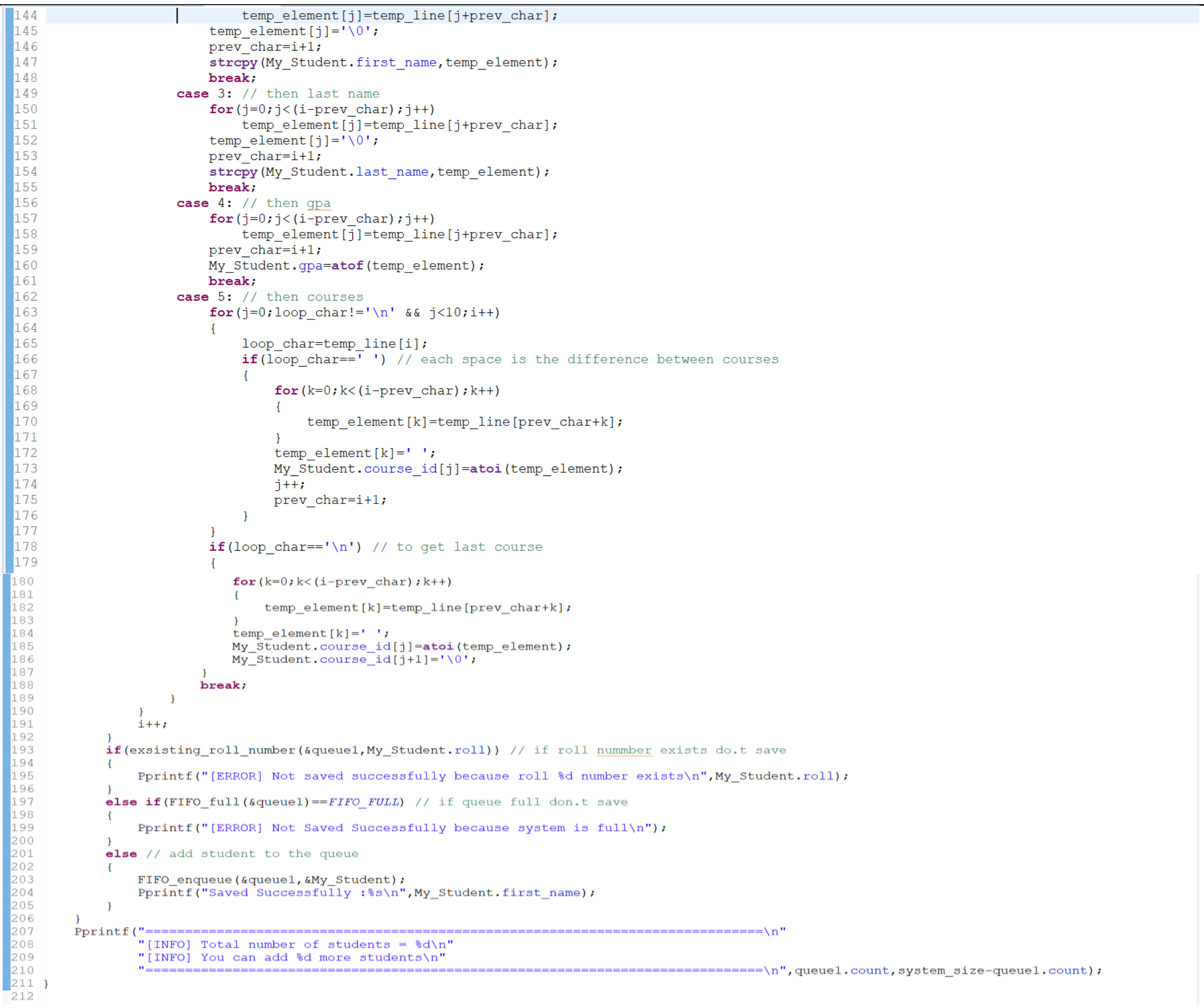


```c
int exsisting_roll_number(FIFO_Queue* My_Queue,int num) // Function to check if roll number exists
{
    int i;
    DataType *My_Student=My_Queue->tail;
    for(i=0;i<My_Queue->count+1;i++)
    {
        if(My_Student->roll==num)
            return 1;
        if(My_Student == (My_Queue->base + (My_Queue->length)*sizeof(DataType) ) ) // for circular queue
            My_Student=My_Queue->base;
        My_Student++; // to navigate between students
    }
    return 0;
}
```

## First function for the user to use is to add students from a text file:

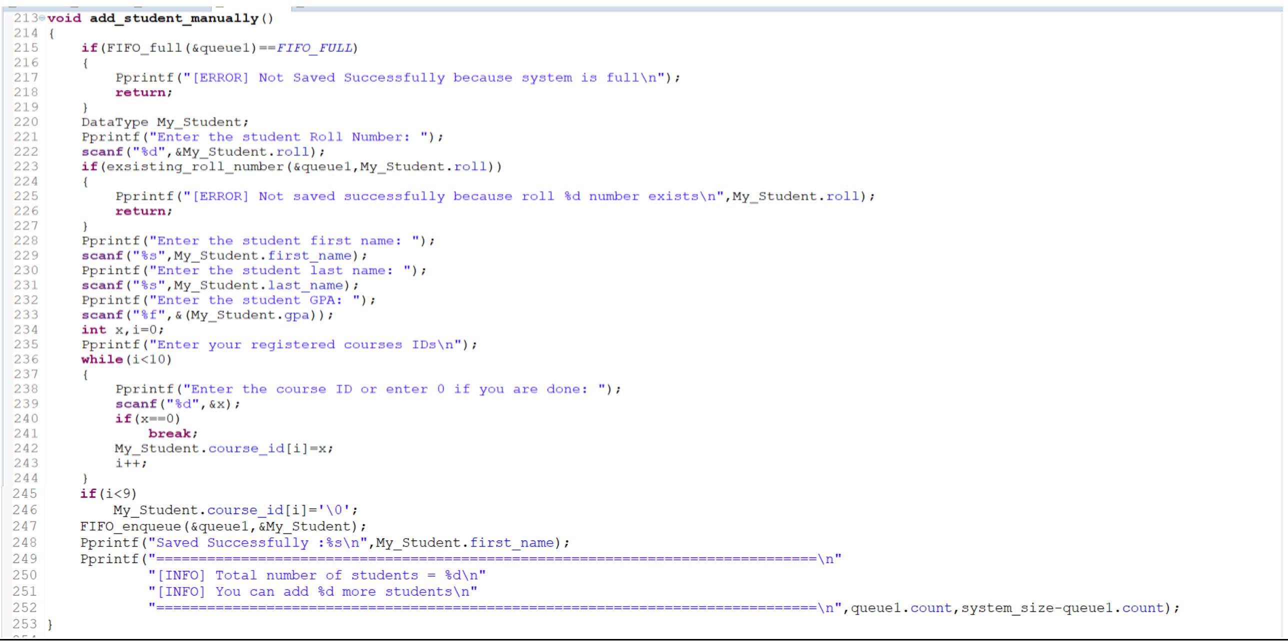


```c
void add_student_file() //to add students data from text file
{
    FILE *ptr;
    char temp_line[250];
    char loop_char;
    char temp_element[50];
    int counter=0,i,j,k,prev_char=0;
    ptr=fopen("file.txt","r");
    if(ptr==NULL)
    {
        Pprintf("File can.t be opened\n");
        return;
    }
    while(fgets(temp_line,250,ptr)) // to read each line and put it in var temp_line
    {
        DataType My_Student;
        counter=0,prev_char=0;
        i=0;
        loop_char=temp_line[i];
        while(loop_char!='\n') // loop for every line
        {
            loop_char=temp_line[i];
            if(loop_char==' ' && counter<5) // each space is the diference between each data type
            {
                counter++;
                switch(counter)
                {
                case 1: // after first space we read roll number
                    for(j=0;j<(i-prev_char);j++)
                        temp_element[j]=temp_line[j+prev_char]; //copy from prev char to the char before space
                    temp_element[j]=' ';
                    prev_char=i+1;
                    My_Student.roll=atoi(temp_element);
                    break;
                case 2: // second space we read first name
                    for(j=0;j<(i-prev_char);j++)
```



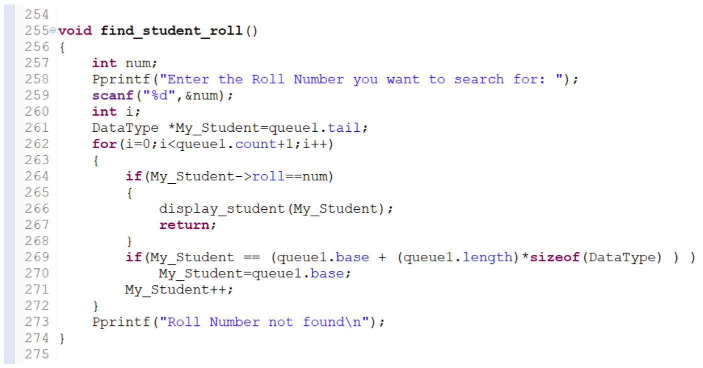
```c
                        temp_element[j]=temp_line[j+prev_char];
                    temp_element[j]='\0';
                    prev_char=i+1;
                    strcpy(My_Student.first_name,temp_element);
                    break;
                case 3: // then last name
                    for(j=0;j<(i-prev_char);j++)
                        temp_element[j]=temp_line[j+prev_char];
                    temp_element[j]='\0';
                    prev_char=i+1;
                    strcpy(My_Student.last_name,temp_element);
                    break;
                case 4: // then gpa
                    for(j=0;j<(i-prev_char);j++)
                        temp_element[j]=temp_line[j+prev_char];
                    prev_char=i+1;
                    My_Student.gpa=atof(temp_element);
                    break;
                case 5: // then courses
                    for(j=0;loop_char!='\n' && j<10;i++)
                    {
                        loop_char=temp_line[i];
                        if(loop_char==' ') // each space is the difference between courses
                        {
                            for(k=0;k<(i-prev_char);k++)
                            {
                                temp_element[k]=temp_line[prev_char+k];
                            }
                            temp_element[k]=' ';
                            My_Student.course_id[j]=atoi(temp_element);
                            j++;
                            prev_char=i+1;
                        }
                    }
                    if(loop_char=='\n') // to get last course
                    {
                        for(k=0;k<(i-prev_char);k++)
                        {
                            temp_element[k]=temp_line[prev_char+k];
                        }
                        temp_element[k]=' ';
                        My_Student.course_id[j]=atoi(temp_element);
                        My_Student.course_id[j+1]='\0';
                    }
                    break;
                }
            }
            i++;
        }
        if(exsisting_roll_number(&queue1,My_Student.roll)) // if roll nummber exists do.t save
        {
            Pprintf("[ERROR] Not saved successfully because roll %d number exists\n",My_Student.roll);
        }
        else if(FIFO_full(&queue1)==FIFO_FULL) // if queue full don.t save
        {
            Pprintf("[ERROR] Not Saved Successfully because system is full\n");
        }
        else // add student to the queue
        {
            FIFO_enqueue(&queue1,&My_Student);
            Pprintf("Saved Successfully :%s\n",My_Student.first_name);
        }
    }
    Pprintf("============================================================================\n"
            "[INFO] Total number of students = %d\n"
            "[INFO] You can add %d more students\n"
            "============================================================================\n",queue1.count,system_size-queue1.count);
}
```

## Function to add students manually:



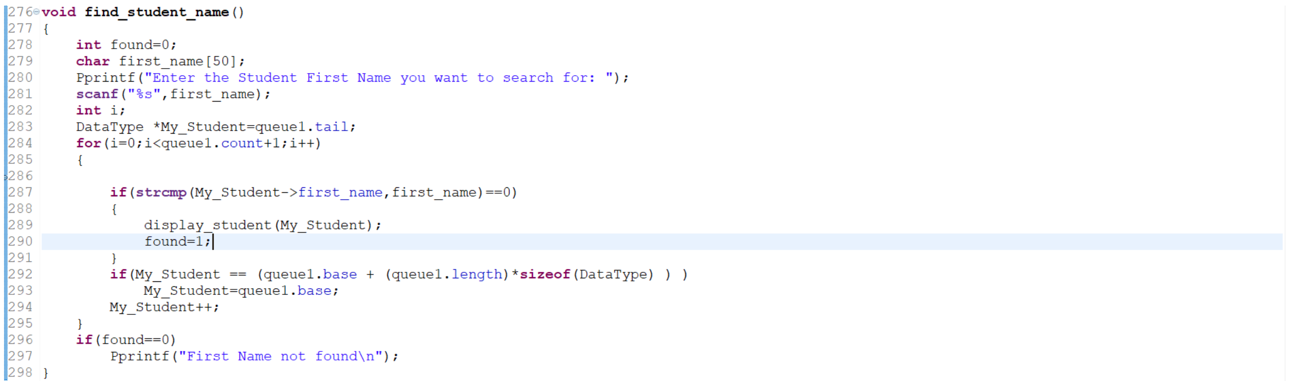
```c
void add_student_manually()
{
    if(FIFO_full(&queue1)==FIFO_FULL)
    {
        Pprintf("[ERROR] Not Saved Successfully because system is full\n");
        return;
    }
    DataType My_Student;
    Pprintf("Enter the student Roll Number: ");
    scanf("%d",&My_Student.roll);
    if(exsisting_roll_number(&queue1,My_Student.roll))
    {
        Pprintf("[ERROR] Not saved successfully because roll %d number exists\n",My_Student.roll);
        return;
    }
    Pprintf("Enter the student first name: ");
    scanf("%s",My_Student.first_name);
    Pprintf("Enter the student last name: ");
    scanf("%s",My_Student.last_name);
    Pprintf("Enter the student GPA: ");
    scanf("%f",&(My_Student.gpa));
    int x,i=0;
    Pprintf("Enter your registered courses IDs\n");
    while(i<10)
    {
        Pprintf("Enter the course ID or enter 0 if you are done: ");
        scanf("%d",&x);
        if(x==0)
            break;
        My_Student.course_id[i]=x;
        i++;
    }
    if(i<9)
        My_Student.course_id[i]='\0';
    FIFO_enqueue(&queue1,&My_Student);
    Pprintf("Saved Successfully :%s\n",My_Student.first_name);
    Pprintf("============================================================================\n"
            "[INFO] Total number of students = %d\n"
            "[INFO] You can add %d more students\n"
            "============================================================================\n",queue1.count,system_size-queue1.count);
}
```

## Function to find student by roll number:



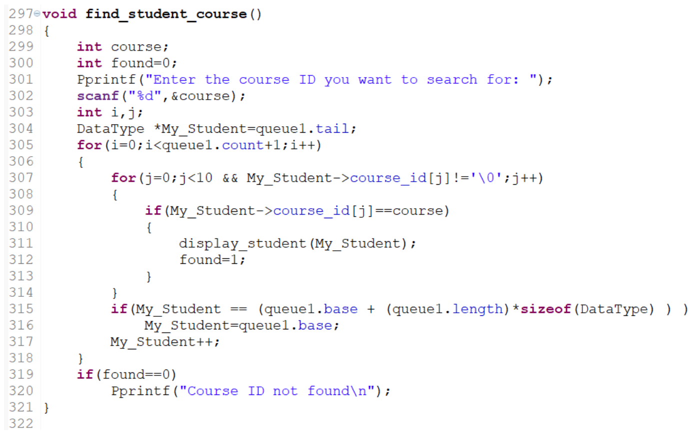
```c
void find_student_roll()
{
    int num;
    Pprintf("Enter the Roll Number you want to search for: ");
    scanf("%d",&num);
    int i;
    DataType *My_Student=queue1.tail;
    for(i=0;i<queue1.count+1;i++)
    {
        if(My_Student->roll==num)
        {
            display_student(My_Student);
            return;
        }
        if(My_Student == (queue1.base + (queue1.length)*sizeof(DataType) ) )
            My_Student=queue1.base;
        My_Student++;
    }
    Pprintf("Roll Number not found\n");
}
```

## Function to find a student by its first name:



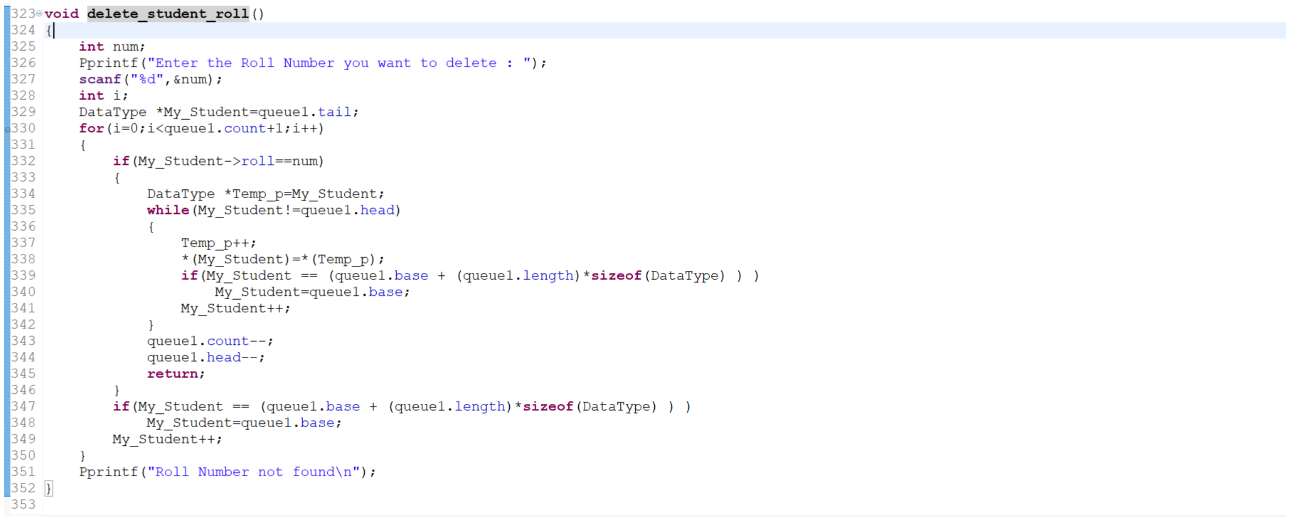
```c
void find_student_name()
{
    int found=0;
    char first_name[50];
    Pprintf("Enter the Student First Name you want to search for: ");
    scanf("%s",first_name);
    int i;
    DataType *My_Student=queue1.tail;
    for(i=0;i<queue1.count+1;i++)
    {

        if(strcmp(My_Student->first_name,first_name)==0)
        {
            display_student(My_Student);
            found=1;
        }
        if(My_Student == (queue1.base + (queue1.length)*sizeof(DataType) ) )
            My_Student=queue1.base;
        My_Student++;
    }
    if(found==0)
        Pprintf("First Name not found\n");
}
```

## Function to find students registered a certain course:



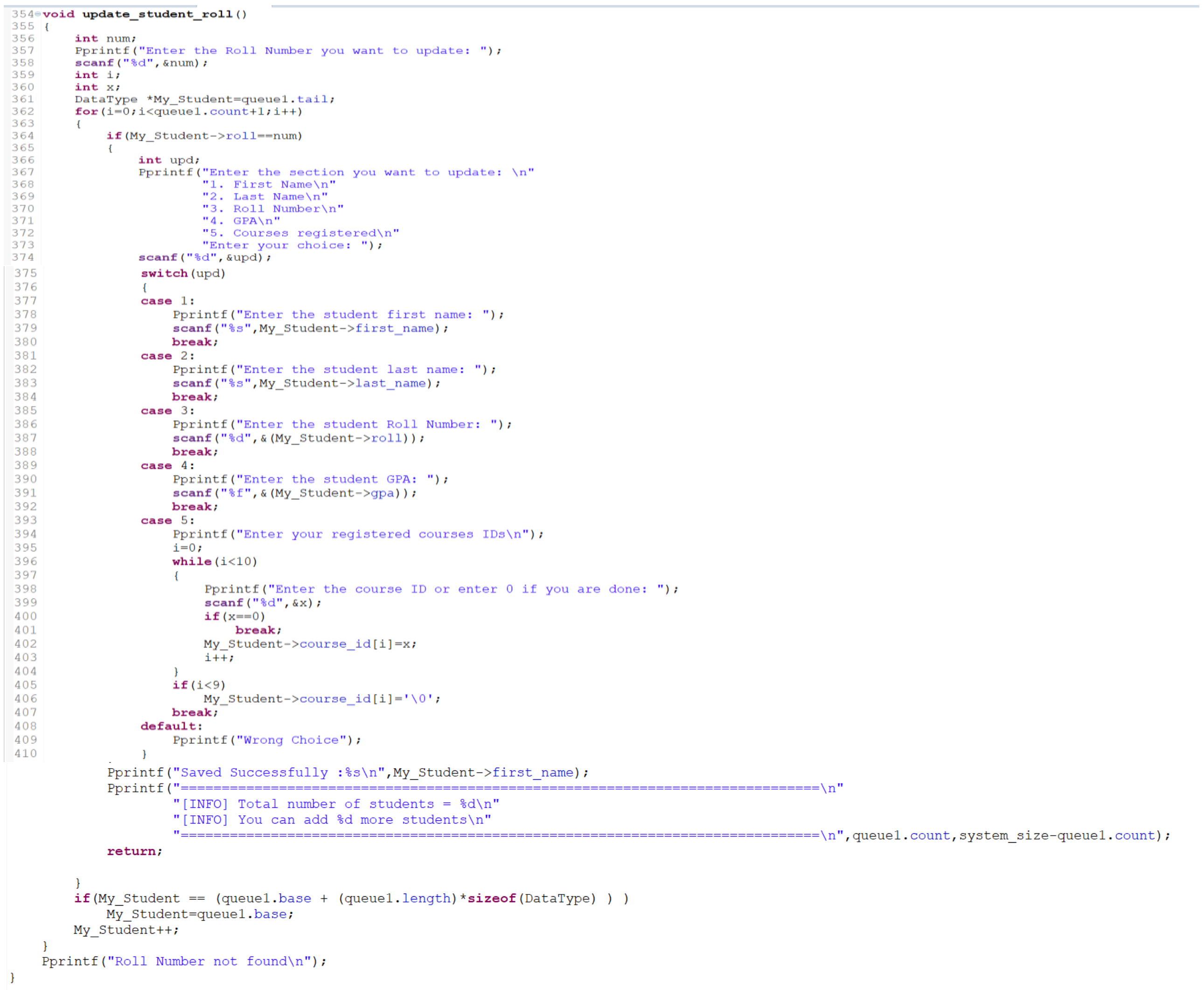
```c
void find_student_course()
{
    int course;
    int found=0;
    Pprintf("Enter the course ID you want to search for: ");
    scanf("%d",&course);
    int i,j;
    DataType *My_Student=queue1.tail;
    for(i=0;i<queue1.count+1;i++)
    {
        for(j=0;j<10 && My_Student->course_id[j]!='\0';j++)
        {
            if(My_Student->course_id[j]==course)
            {
                display_student(My_Student);
                found=1;
            }
        }
        if(My_Student == (queue1.base + (queue1.length)*sizeof(DataType) ) )
            My_Student=queue1.base;
        My_Student++;
    }
    if(found==0)
        Pprintf("Course ID not found\n");
}
```

## Function to delete student using roll number:



```c
void delete_student_roll()
{
    int num;
    Pprintf("Enter the Roll Number you want to delete : ");
    scanf("%d",&num);
    int i;
    DataType *My_Student=queue1.tail;
    for(i=0;i<queue1.count+1;i++)
    {
        if(My_Student->roll==num)
        {
            DataType *Temp_p=My_Student;
            while(My_Student!=queue1.head)
            {
                Temp_p++;
                *(My_Student)=*(Temp_p);
                if(My_Student == (queue1.base + (queue1.length)*sizeof(DataType) ) )
                    My_Student=queue1.base;
                My_Student++;
            }
            queue1.count--;
            queue1.head--;
            return;
        }
        if(My_Student == (queue1.base + (queue1.length)*sizeof(DataType) ) )
            My_Student=queue1.base;
        My_Student++;
    }
    Pprintf("Roll Number not found\n");
}
```

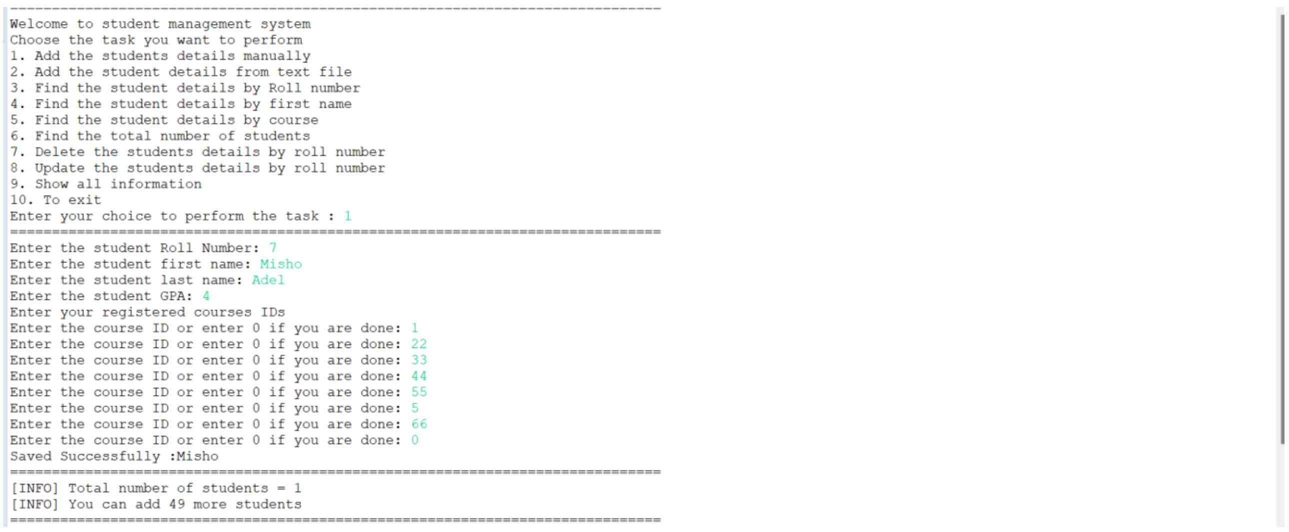
## Function to update student by roll number:



```c
void update_student_roll()
{
    int num;
    Pprintf("Enter the Roll Number you want to update: ");
    scanf("%d",&num);
    int i;
    int x;
    DataType *My_Student=queue1.tail;
    for(i=0;i<queue1.count+1;i++)
    {
        if(My_Student->roll==num)
        {
            int upd;
            Pprintf("Enter the section you want to update: \n"
                    "1. First Name\n"
                    "2. Last Name\n"
                    "3. Roll Number\n"
                    "4. GPA\n"
                    "5. Courses registered\n"
                    "Enter your choice: ");
            scanf("%d",&upd);
            switch(upd)
            {
            case 1:
                Pprintf("Enter the student first name: ");
                scanf("%s",My_Student->first_name);
                break;
            case 2:
                Pprintf("Enter the student last name: ");
                scanf("%s",My_Student->last_name);
                break;
            case 3:
                Pprintf("Enter the student Roll Number: ");
                scanf("%d",&(My_Student->roll));
                break;
            case 4:
                Pprintf("Enter the student GPA: ");
                scanf("%f",&(My_Student->gpa));
                break;
            case 5:
                Pprintf("Enter your registered courses IDs\n");
                i=0;
                while(i<10)
                {
                    Pprintf("Enter the course ID or enter 0 if you are done: ");
                    scanf("%d",&x);
                    if(x==0)
                        break;
                    My_Student->course_id[i]=x;
                    i++;
                }
                if(i<9)
                    My_Student->course_id[i]='\0';
                break;
            default:
                Pprintf("Wrong Choice");
            }
        Pprintf("Saved Successfully :%s\n",My_Student->first_name);
        Pprintf("=============================================================================\n"
                "[INFO] Total number of students = %d\n"
                "[INFO] You can add %d more students\n"
                "=============================================================================\n",queue1.count,system_size-queue1.count);
        return;

    }
    if(My_Student == (queue1.base + (queue1.length)*sizeof(DataType) ) )
        My_Student=queue1.base;
    My_Student++;
  }
  Pprintf("Roll Number not found\n");
}
```

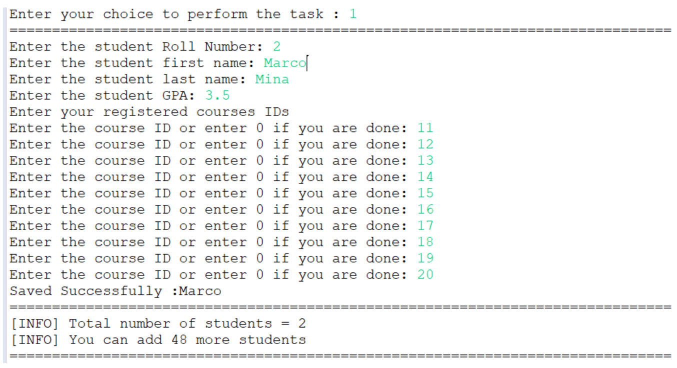
## System Running:

First, we try to add a student manually:



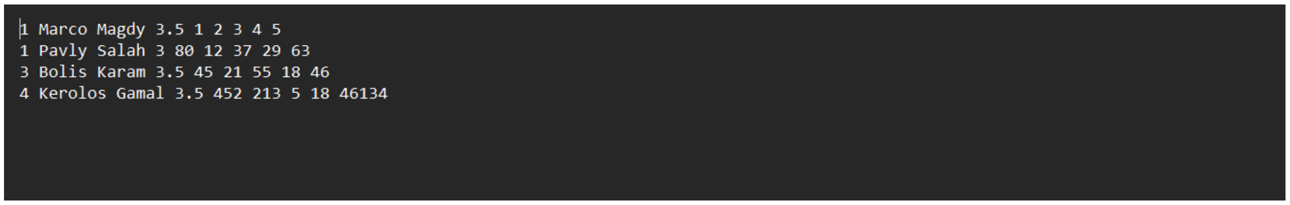
```
---------------------------------------------------------------------------
Welcome to student management system
Choose the task you want to perform
1. Add the students details manually
2. Add the student details from text file
3. Find the student details by Roll number
4. Find the student details by first name
5. Find the student details by course
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 : 1
===========================================================================
Enter the student Roll Number: 7
Enter the student first name: Misho
Enter the student last name: Adel
Enter the student GPA: 4
Enter your registered courses IDs
Enter the course ID or enter 0 if you are done: 1
Enter the course ID or enter 0 if you are done: 22
Enter the course ID or enter 0 if you are done: 33
Enter the course ID or enter 0 if you are done: 44
Enter the course ID or enter 0 if you are done: 55
Enter the course ID or enter 0 if you are done: 5
Enter the course ID or enter 0 if you are done: 66
Enter the course ID or enter 0 if you are done: 0
Saved Successfully :Misho
===========================================================================
[INFO] Total number of students = 1
[INFO] You can add 49 more students
===========================================================================
```

Then we add another student:

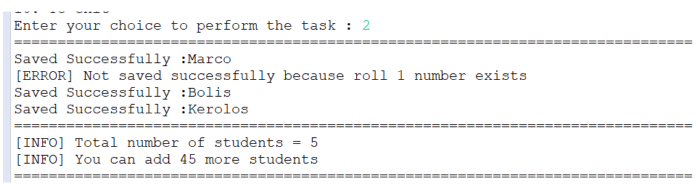


```
Enter your choice to perform the task : 1
===========================================================================
Enter the student Roll Number: 2
Enter the student first name: Marco
Enter the student last name: Mina
Enter the student GPA: 3.5
Enter your registered courses IDs
Enter the course ID or enter 0 if you are done: 11
Enter the course ID or enter 0 if you are done: 12
Enter the course ID or enter 0 if you are done: 13
Enter the course ID or enter 0 if you are done: 14
Enter the course ID or enter 0 if you are done: 15
Enter the course ID or enter 0 if you are done: 16
Enter the course ID or enter 0 if you are done: 17
Enter the course ID or enter 0 if you are done: 18
Enter the course ID or enter 0 if you are done: 19
Enter the course ID or enter 0 if you are done: 20
Saved Successfully :Marco
===========================================================================
[INFO] Total number of students = 2
[INFO] You can add 48 more students
===========================================================================
```

Now we try to add from the following text file:



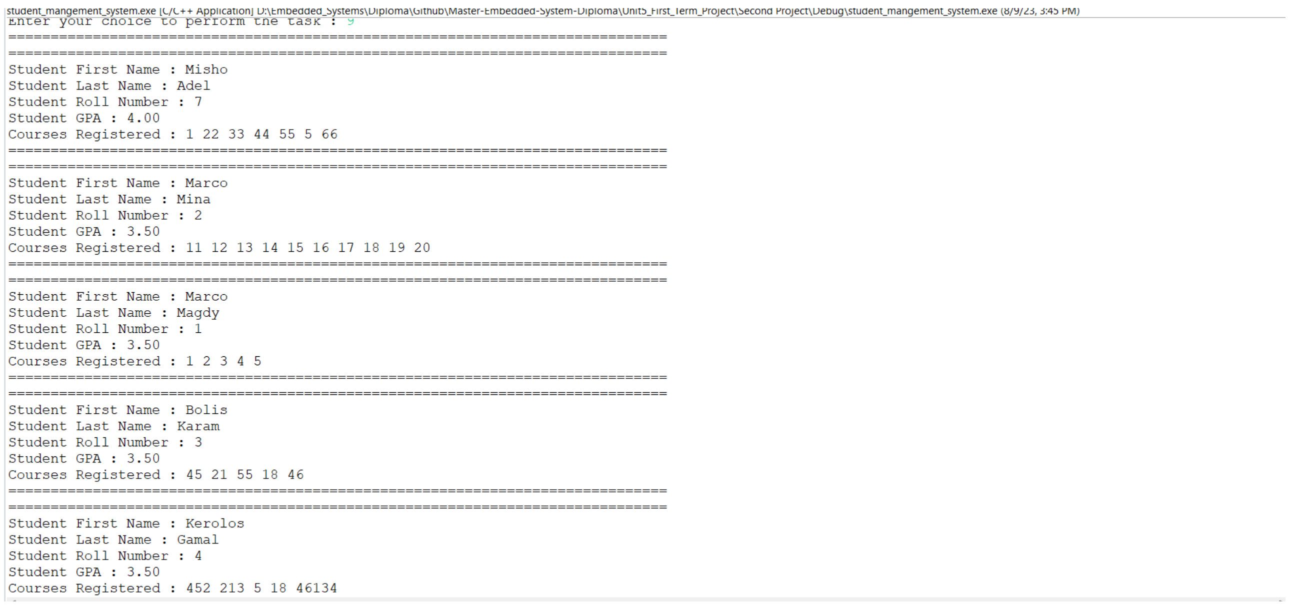
```
1 Marco Magdy 3.5 1 2 3 4 5
1 Pavly Salah 3 80 12 37 29 63
3 Bolis Karam 3.5 45 21 55 18 46
4 Kerolos Gamal 3.5 452 213 5 18 46134
```



```
Enter your choice to perform the task : 2
===========================================================================
Saved Successfully :Marco
[ERROR] Not saved successfully because roll 1 number exists
Saved Successfully :Bolis
Saved Successfully :Kerolos
===========================================================================
[INFO] Total number of students = 5
[INFO] You can add 45 more students
===========================================================================
```

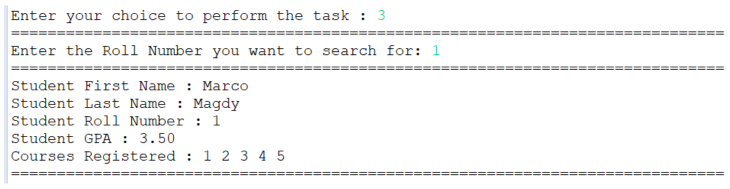
We can notice that it counts the students and checks for existing roll number correctly.

Now we try to display all students:

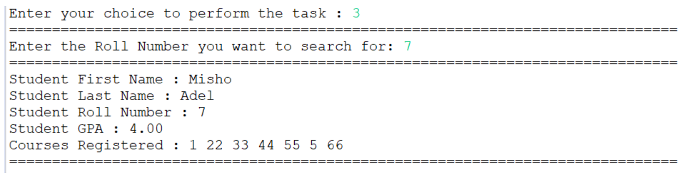


```
student_mangement_system.exe [C/C++ Application] D:\Embedded_Systems\Diploma\Github\Master-Embedded-System-Diploma\Unit5_First_Term_Project\Second Project\Debug\student_mangement_system.exe (8/9/23, 3:45 PM)
Enter your choice to perform the task : 9
============================================================================
============================================================================
Student First Name : Misho
Student Last Name : Adel
Student Roll Number : 7
Student GPA : 4.00
Courses Registered : 1 22 33 44 55 5 66
============================================================================
============================================================================
Student First Name : Marco
Student Last Name : Mina
Student Roll Number : 2
Student GPA : 3.50
Courses Registered : 11 12 13 14 15 16 17 18 19 20
============================================================================
============================================================================
Student First Name : Marco
Student Last Name : Magdy
Student Roll Number : 1
Student GPA : 3.50
Courses Registered : 1 2 3 4 5
============================================================================
============================================================================
Student First Name : Bolis
Student Last Name : Karam
Student Roll Number : 3
Student GPA : 3.50
Courses Registered : 45 21 55 18 46
============================================================================
============================================================================
Student First Name : Kerolos
Student Last Name : Gamal
Student Roll Number : 4
Student GPA : 3.50
Courses Registered : 452 213 5 18 46134
```

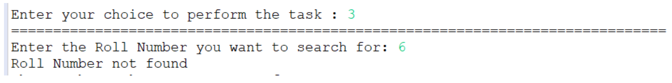
Now we try the search by roll number:



```
Enter your choice to perform the task : 3
============================================================================
Enter the Roll Number you want to search for: 1
============================================================================
Student First Name : Marco
Student Last Name : Magdy
Student Roll Number : 1
Student GPA : 3.50
Courses Registered : 1 2 3 4 5
============================================================================
```

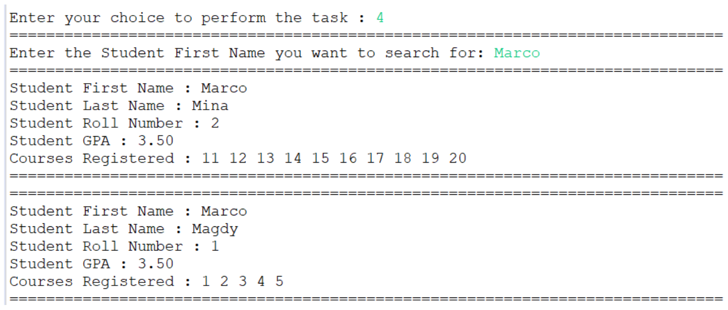


```
Enter your choice to perform the task : 3
============================================================================
Enter the Roll Number you want to search for: 7
============================================================================
Student First Name : Misho
Student Last Name : Adel
Student Roll Number : 7
Student GPA : 4.00
Courses Registered : 1 22 33 44 55 5 66
============================================================================
```

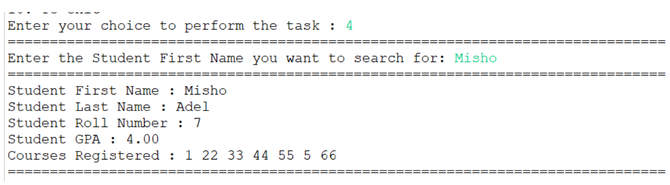


```
Enter your choice to perform the task : 3
============================================================================
Enter the Roll Number you want to search for: 6
Roll Number not found
```

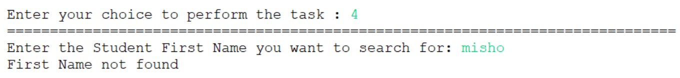
## Now we try the search by first name:



```
Enter your choice to perform the task : 4
==============================================================================
Enter the Student First Name you want to search for: Marco
==============================================================================
Student First Name : Marco
Student Last Name : Mina
Student Roll Number : 2
Student GPA : 3.50
Courses Registered : 11 12 13 14 15 16 17 18 19 20
==============================================================================
==============================================================================
Student First Name : Marco
Student Last Name : Magdy
Student Roll Number : 1
Student GPA : 3.50
Courses Registered : 1 2 3 4 5
==============================================================================
```

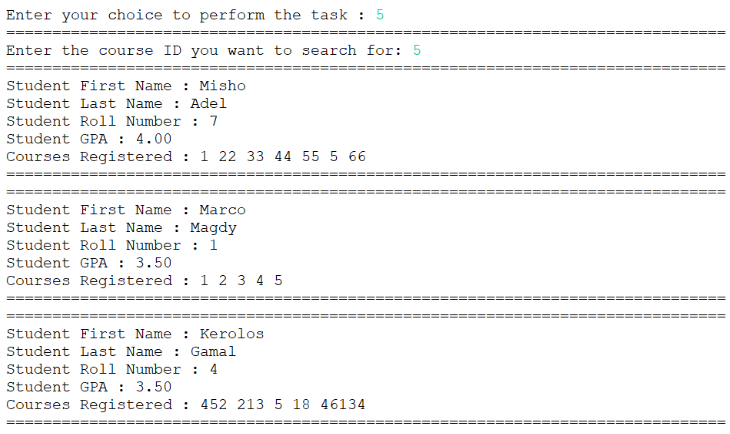


```
Enter your choice to perform the task : 4
==============================================================================
Enter the Student First Name you want to search for: Misho
==============================================================================
Student First Name : Misho
Student Last Name : Adel
Student Roll Number : 7
Student GPA : 4.00
Courses Registered : 1 22 33 44 55 5 66
==============================================================================
```

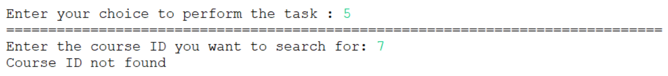


```
Enter your choice to perform the task : 4
==============================================================================
Enter the Student First Name you want to search for: misho
First Name not found
```

## Now we try the search for student with course ID:

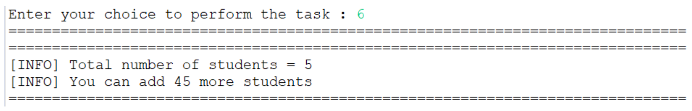


```
Enter your choice to perform the task : 5
==============================================================================
Enter the course ID you want to search for: 5
==============================================================================
Student First Name : Misho
Student Last Name : Adel
Student Roll Number : 7
Student GPA : 4.00
Courses Registered : 1 22 33 44 55 5 66
==============================================================================
==============================================================================
Student First Name : Marco
Student Last Name : Magdy
Student Roll Number : 1
Student GPA : 3.50
Courses Registered : 1 2 3 4 5
==============================================================================
==============================================================================
Student First Name : Kerolos
Student Last Name : Gamal
Student Roll Number : 4
Student GPA : 3.50
Courses Registered : 452 213 5 18 46134
==============================================================================
```



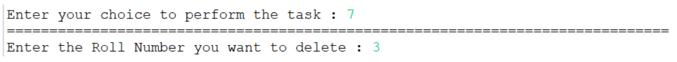
```
Enter your choice to perform the task : 5
==============================================================================
Enter the course ID you want to search for: 7
Course ID not found
```

## Now we try the info function:

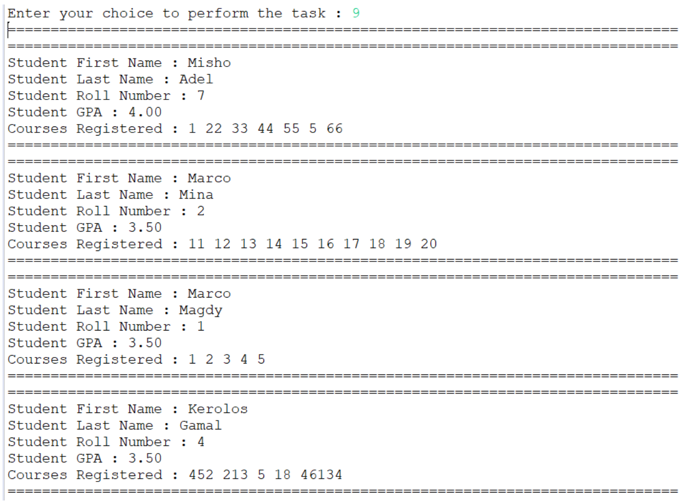


```
Enter your choice to perform the task : 6
==============================================================================
==============================================================================
[INFO] Total number of students = 5
[INFO] You can add 45 more students
==============================================================================
```

## Now we try the delete function:

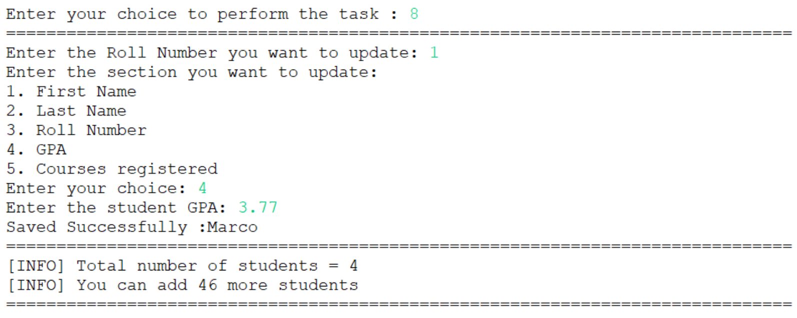


```
Enter your choice to perform the task : 7
==============================================================================
Enter the Roll Number you want to delete : 3
```

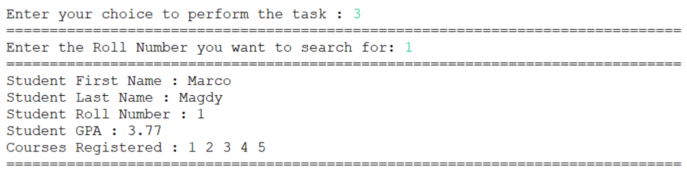


```
Enter your choice to perform the task : 9
==============================================================================
==============================================================================
Student First Name : Misho
Student Last Name : Adel
Student Roll Number : 7
Student GPA : 4.00
Courses Registered : 1 22 33 44 55 5 66
==============================================================================
==============================================================================
Student First Name : Marco
Student Last Name : Mina
Student Roll Number : 2
Student GPA : 3.50
Courses Registered : 11 12 13 14 15 16 17 18 19 20
==============================================================================
==============================================================================
Student First Name : Marco
Student Last Name : Magdy
Student Roll Number : 1
Student GPA : 3.50
Courses Registered : 1 2 3 4 5
==============================================================================
==============================================================================
Student First Name : Kerolos
Student Last Name : Gamal
Student Roll Number : 4
Student GPA : 3.50
Courses Registered : 452 213 5 18 46134
==============================================================================
```

## Now we try the update function:



```
Enter your choice to perform the task : 8
==============================================================================
Enter the Roll Number you want to update: 1
Enter the section you want to update:
1. First Name
2. Last Name
3. Roll Number
4. GPA
5. Courses registered
Enter your choice: 4
Enter the student GPA: 3.77
Saved Successfully :Marco
==============================================================================
[INFO] Total number of students = 4
[INFO] You can add 46 more students
==============================================================================
```



```
Enter your choice to perform the task : 3
==============================================================================
Enter the Roll Number you want to search for: 1
==============================================================================
Student First Name : Marco
Student Last Name : Magdy
Student Roll Number : 1
Student GPA : 3.77
Courses Registered : 1 2 3 4 5
==============================================================================
```

This code is tested for several test cases and proved to be working just fine in all of them and test cases given in this report is just little demonstration of the system.