Mastering embedded system online diploma

www.learn-in-depth.com

First term (Final project 2)

Eng. Mo'men Amr Mohammed Amr

My profile: https://www.learn-in-depth.com/online-diploma/momen55amr@gmail.com

Problem Statement

Write a program to build 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 of students.
7. Delete a student.
8. Update a student.
9. Show all the data.
10. Exit the program.

```
void add_student_manually ();
```

if option 1 is selected this function gets called to store a student record into the database manually. we tested the function in different scenarios: first, if we entered a new roll number then, if it already used and show the data registered.

```
Welcome to the student management system
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:1
-------------------------
Add the student details
-------------------------
Enter the roll number:1
Enter the first name of student:momen
Enter the last name of student:amr
Enter the GPA you obtained:3.0
Enter the course ID of each course
Course 1 ID:1
Course 2 ID:2
Course 3 ID:3
Course 4 ID:4
Course 5 ID:5
-------------------------
[INFO]  Student details was added successfully
-------------------------
-------------------------
[INFO] The total number of students is 1
[INFO] You can add up to 50 students
[INFO] You can add 49 more students
-------------------------
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:1
-------------------------
Add the student details
-------------------------
Enter the roll number:1
-------------------------
[ERROR]  Roll number 1 is already taken
-------------------------
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:1
-------------------------
Add the student details
-------------------------
Enter the roll number:2
Enter the first name of student:ahmed
Enter the last name of student:ali
Enter the GPA you obtained:3.2
Enter the course ID of each course
Course 1 ID:1
Course 2 ID:2
Course 3 ID:3
Course 4 ID:4
Course 5 ID:5
-------------------------
[INFO]  Student details was added successfully
-------------------------
-------------------------
[INFO] The total number of students is 2
[INFO] You can add up to 50 students
[INFO] You can add 48 more students
-------------------------
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:9

-------------------------
Student first name: momen
Student last name: amr
Student roll number: 1
Student GPA number: 3.00
The course ID: 1
The course ID: 2
The course ID: 3
The course ID: 4
The course ID: 5
-------------------------
Student first name: ahmed
Student last name: ali
Student roll number: 2
Student GPA number: 3.20
The course ID: 1
The course ID: 2
The course ID: 3
The course ID: 4
The course ID: 5
-------------------------
```

```
void add_student_file ();
```

if option 2 is selected this function gets called to store all accepted student records into the database from a file. we tested the function in different scenarios: first, if we entered a wrong text file then, if the text contains duplicate roll numbers and at last show the data registered.

```
Welcome to the student management system
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:2
[ERROR]  Students.text can't be opened
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:2
[INFO]  Roll number 1 was saved successfully
[INFO]  Roll number 2 was saved successfully
[ERROR]  Roll number 2 is already taken
[INFO]  Roll number 4 was saved successfully
[INFO]  Roll number 5 was saved successfully
[INFO]  Student details was added successfully
-------------------------
[INFO] The total number of students is 4
[INFO] You can add up to 50 students
[INFO] You can add 46 more students
-------------------------
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:2
[ERROR]  Roll number 1 is already taken
[ERROR]  Roll number 2 is already taken
[INFO]  Roll number 3 was saved successfully
[ERROR]  Roll number 4 is already taken
```

```
[ERROR]  Roll number 5 is already taken
[INFO]  Student details was added successfully
-------------------------
[INFO] The total number of students is 5
[INFO] You can add up to 50 students
[INFO] You can add 45 more students
-------------------------
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:9

-------------------------
Student first name: momen
Student last name: amr
Student roll number: 1
Student GPA number: 3.20
The course ID: 1
The course ID: 2
The course ID: 3
The course ID: 4
The course ID: 5
-------------------------
Student first name: ahmed
Student last name: salem
Student roll number: 2
Student GPA number: 2.80
The course ID: 4
The course ID: 5
The course ID: 6
The course ID: 7
The course ID: 8
-------------------------
Student first name: salem
Student last name: ahmed
Student roll number: 4
Student GPA number: 3.40
The course ID: 1
The course ID: 2
The course ID: 3
The course ID: 4
The course ID: 5
-------------------------
Student first name: ahmed
Student last name: said
Student roll number: 5
Student GPA number: 3.00
The course ID: 5
The course ID: 8
The course ID: 9
The course ID: 7
The course ID: 4
-------------------------
Student first name: mohammad
Student last name: gaber
Student roll number: 3
Student GPA number: 3.00
The course ID: 3
The course ID: 5
The course ID: 4
The course ID: 7
The course ID: 6
-------------------------
```

```
void find_rl ();
```

if option 3 is selected this function gets called to find a student record by its unique roll number. we tested the function in different scenarios: first, if the database is empty then, if the roll number was not found then, if a roll number was found.

```
Welcome to the student management system
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:3
Enter the roll number of the student:1
-------------------------
[ERROR]  Students database is empty
-------------------------
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:2
[INFO]  Roll number 1 was saved successfully
[INFO]  Roll number 2 was saved successfully
[INFO]  Roll number 3 was saved successfully
[INFO]  Roll number 4 was saved successfully
[INFO]  Roll number 5 was saved successfully
[INFO]  Student details was added successfully
-------------------------
[INFO] The total number of students is 5
[INFO] You can add up to 50 students
[INFO] You can add 45 more students
-------------------------
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:3
Enter the roll number of the student:9
```

```
-------------------------
[ERROR]  Roll number 9 was not found
-------------------------
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:3
Enter the roll number of the student:2
-------------------------
The student details are
Student first name: ahmed
Student last name: salem
Student roll number: 2
Student GPA number: 2.80
The course ID: 4
The course ID: 5
The course ID: 6
The course ID: 7
The course ID: 8
-------------------------
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:3
Enter the roll number of the student:5
-------------------------
The student details are
Student first name: ahmed
Student last name: said
Student roll number: 5
Student GPA number: 3.00
The course ID: 5
The course ID: 8
The course ID: 9
The course ID: 7
The course ID: 4
-------------------------
```

```
void find_fn ();
```

if option 4 is selected this function gets called to find all student records matching by first name. we tested the function in different scenarios: first, if the database is empty then, if the name was not found then, if a name was found or matches multiple entries.

```
Welcome to the student management system
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:4
Enter the first name of the student:ahmed
-------------------------
[ERROR]  Students database is empty
-------------------------
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:2
[INFO]  Roll number 1 was saved successfully
[INFO]  Roll number 2 was saved successfully
[INFO]  Roll number 3 was saved successfully
[INFO]  Roll number 4 was saved successfully
[INFO]  Roll number 5 was saved successfully
[INFO]  Student details was added successfully
-------------------------
[INFO] The total number of students is 5
[INFO] You can add up to 50 students
[INFO] You can add 45 more students
-------------------------
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:4
Enter the first name of the student:ali
-------------------------
```

```
[ERROR]  First name ali was not found
-------------------------
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:4
Enter the first name of the student:momen
-------------------------
The student details are
Student first name: momen
Student last name: amr
Student roll number: 1
Student GPA number: 3.20
The course ID: 1
The course ID: 2
The course ID: 3
The course ID: 4
The course ID: 5
-------------------------
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:4
Enter the first name of the student:ahmed
-------------------------
The student details are
Student first name: ahmed
Student last name: salem
Student roll number: 2
Student GPA number: 2.80
The course ID: 4
The course ID: 5
The course ID: 6
The course ID: 7
The course ID: 8
-------------------------
The student details are
Student first name: ahmed
Student last name: said
Student roll number: 5
Student GPA number: 3.00
The course ID: 5
The course ID: 8
The course ID: 9
The course ID: 7
The course ID: 4
```

```
void find_c ();
```

if option 5 is selected this function gets called to find all student records matching by course ID, we tested the function in different scenarios: first, if the database is empty then, if the course was not found then, if a course was found or matches multiple.

```
Welcome to the student management system
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:5
Enter the course ID:1
-------------------------
[ERROR]  Students database is empty
-------------------------
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:2
[INFO]  Roll number 1 was saved successfully
[INFO]  Roll number 2 was saved successfully
[INFO]  Roll number 3 was saved successfully
[INFO]  Roll number 4 was saved successfully
[INFO]  Roll number 5 was saved successfully
[INFO]  Student details was added successfully
-------------------------
[INFO] The total number of students is 5
[INFO] You can add up to 50 students
[INFO] You can add 45 more students
-------------------------
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:5
Enter the course ID:22
-------------------------
```

```
[ERROR]  Course ID 22 was not found
-------------------------
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:5
Enter the course ID:9
-------------------------
The student details are
Student first name: ahmed
Student last name: said
Student roll number: 5
Student GPA number: 3.00
-------------------------
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:5
Enter the course ID:5
-------------------------
The student details are
Student first name: momen
Student last name: amr
Student roll number: 1
Student GPA number: 3.20
-------------------------
The student details are
Student first name: ahmed
Student last name: salem
Student roll number: 2
Student GPA number: 2.80
-------------------------
The student details are
Student first name: mohammad
Student last name: gaber
Student roll number: 3
Student GPA number: 3.00
-------------------------
The student details are
Student first name: salem
Student last name: ahmed
Student roll number: 4
Student GPA number: 3.40
-------------------------
```

```
void tot_s ();
```

if option 6 is selected this function gets called to find student count, we tested the function in different scenarios: first, if the database is empty then, if it contains entries.

```
Welcome to the student management system
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:6
-------------------------
[INFO] The total number of students is 0
[INFO] You can add up to 50 students
[INFO] You can add 50 more students

Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:2
[INFO]  Roll number 1 was saved successfully
[INFO]  Roll number 2 was saved successfully
[INFO]  Roll number 3 was saved successfully
[INFO]  Student details was added successfully
-------------------------
[INFO] The total number of students is 5
[INFO] You can add up to 50 students
[INFO] You can add 45 more students
-------------------------
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:6
-------------------------
[INFO] The total number of students is 5
[INFO] You can add up to 50 students
[INFO] You can add 45 more students
-------------------------
```

```
void del_s ();
```

if option 7 is selected this function gets called to delete a student record by roll number, we tested the function in different scenarios: first, if the database is empty then, if the roll number was not found then, if a roll number was found and at last show the data registered.

```
Welcome to the student management system
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:7
Enter the roll number which you want to delete:1
-------------------------
[ERROR]  Students database is empty
-------------------------
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:2
[INFO]  Roll number 1 was saved successfully
[INFO]  Roll number 2 was saved successfully
[INFO]  Roll number 3 was saved successfully
[INFO]  Roll number 4 was saved successfully
[INFO]  Roll number 5 was saved successfully
[INFO]  Student details was added successfully
-------------------------
[INFO] The total number of students is 5
[INFO] You can add up to 50 students
[INFO] You can add 45 more students
-------------------------
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:7
Enter the roll number which you want to delete:9
-------------------------
```

```
[ERROR]  Roll number 9 was not found
-------------------------
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:7
Enter the roll number which you want to delete:2
-------------------------
[INFO]  The roll number was removed successfully
-------------------------
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:9

-------------------------
Student first name: momen
Student last name: amr
Student roll number: 1
Student GPA number: 3.20
The course ID: 1
The course ID: 2
The course ID: 3
The course ID: 4
The course ID: 5
-------------------------
Student first name: mohammad
Student last name: gaber
Student roll number: 3
Student GPA number: 3.00
The course ID: 3
The course ID: 5
The course ID: 4
The course ID: 7
The course ID: 6
-------------------------
Student first name: salem
Student last name: ahmed
Student roll number: 4
Student GPA number: 3.40
The course ID: 1
The course ID: 2
The course ID: 3
The course ID: 4
The course ID: 5
-------------------------
Student first name: ahmed
Student last name: said
Student roll number: 5
Student GPA number: 3.00
The course ID: 5
The course ID: 8
The course ID: 9
The course ID: 7
The course ID: 4
-------------------------
```

```c
void up_s ();
```

if option 8 is selected this function gets called to update a student record by roll number, we tested the function in different scenarios: first, if the database is empty then, if the roll number was not found then, if a roll number was found and at last show the data registered.

```
Welcome to the student management system
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:2
[INFO]  Roll number 1 was saved successfully
[INFO]  Roll number 2 was saved successfully
[INFO]  Roll number 3 was saved successfully
[INFO]  Roll number 4 was saved successfully
[INFO]  Roll number 5 was saved successfully
[INFO]  Student details was added successfully
-------------------------
[INFO] The total number of students is 5
[INFO] You can add up to 50 students
[INFO] You can add 45 more students
-------------------------
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:8
Enter the roll number of the student:11
-------------------------
[ERROR]  Roll number 11 was not found
-------------------------
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:8
Enter the roll number of the student:2
1. first name
```

```
2. last name
3. roll number
4. GPA
5. courses
1
Enter the new first name: saber
-------------------------
[INFO]  Updated successfully
-------------------------
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:9

-------------------------
Student first name: momen
Student last name: amr
Student roll number: 1
Student GPA number: 3.20
The course ID: 1
The course ID: 2
The course ID: 3
The course ID: 4
The course ID: 5
-------------------------
Student first name: saber
Student last name: salem
Student roll number: 2
Student GPA number: 2.80
The course ID: 4
The course ID: 5
The course ID: 6
The course ID: 7
The course ID: 8
-------------------------
Student first name: mohammad
Student last name: gaber
Student roll number: 3
Student GPA number: 3.00
The course ID: 3
The course ID: 5
The course ID: 4
The course ID: 7
The course ID: 6
-------------------------
Student first name: salem
Student last name: ahmed
Student roll number: 4
Student GPA number: 3.40
The course ID: 1
The course ID: 2
The course ID: 3
The course ID: 4
The course ID: 5
-------------------------
Student first name: ahmed
Student last name: said
Student roll number: 5
Student GPA number: 3.00
The course ID: 5
The course ID: 8
The course ID: 9
The course ID: 7
The course ID: 4
-------------------------
```

```
void show_s ();
```

if option 9 is selected this function gets called to show all the data stored in the database, we tested the function in different scenarios: first, if the database is empty then, if it contains entries.

```
Welcome to the student management system
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:9
-------------------------
[ERROR]  Students database is empty
-------------------------
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:2
[INFO]  Roll number 1 was saved successfully
[INFO]  Roll number 2 was saved successfully
[INFO]  Roll number 3 was saved successfully
[INFO]  Roll number 4 was saved successfully
[INFO]  Roll number 5 was saved successfully
[INFO]  Student details was added successfully
-------------------------
[INFO] The total number of students is 5
[INFO] You can add up to 50 students
[INFO] You can add 45 more students
-------------------------
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:9

-------------------------
Student first name: momen
Student last name: amr
Student roll number: 1
Student GPA number: 3.20
The course ID: 1
The course ID: 2
The course ID: 3
The course ID: 4
The course ID: 5
-------------------------
Student first name: ahmed
Student last name: salem
Student roll number: 2
Student GPA number: 2.80
The course ID: 4
The course ID: 5
The course ID: 6
The course ID: 7
The course ID: 8
-------------------------
Student first name: mohammad
Student last name: gaber
Student roll number: 3
Student GPA number: 3.00
The course ID: 3
The course ID: 5
The course ID: 4
The course ID: 7
The course ID: 6
-------------------------
Student first name: salem
Student last name: ahmed
Student roll number: 4
Student GPA number: 3.40
The course ID: 1
The course ID: 2
The course ID: 3
The course ID: 4
The course ID: 5
-------------------------
Student first name: ahmed
Student last name: said
Student roll number: 5
Student GPA number: 3.00
The course ID: 5
The course ID: 8
The course ID: 9
The course ID: 7
The course ID: 4
-------------------------
```

Exit and wrong choices handling;

In case the student enters a non-valid option, it shows a warning and start over, and if exit is selected the program terminates the console.

```
Welcome to the student management system
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:22
please enter a valid option !!
Choose the task that you want to perform
 1. Add the student details manually
 2. Add the student details from text file
 3. Find the student details by roll number
 4. Find the student details by first name
 5. Find the student details by course ID
 6. Find the total number of students
 7. Delete the student details by roll number
 8. Update the student details by roll number
 9. Show all information
10. Exit
Enter your choice to perform the task:10
program is terminated !!
```

Project c&h code files

Main.c file

```c
/*
 * main.c
 *
 *  Created on: Jun 19, 2022
 *      Author: momen
 */


#include "stdio.h"
#include "Database.h"
#include "FIFO.h"

FIFO_Buffer buffer;

void main()
{
	//initialize the FIFO buffer with the global array
	FIFO_init(&buffer, st, 50);

	int x;
	printf("Welcome to the student management system\n");

	//this loop acts as the program to execute the operations required
	while(1)
	{
		printf("Choose the task that you want to perform\n");
		printf(" 1. Add the student details manually\n");
		printf(" 2. Add the student details from text file\n");
		printf(" 3. Find the student details by roll number\n");
		printf(" 4. Find the student details by first name\n");
		printf(" 5. Find the student details by course ID\n");
		printf(" 6. Find the total number of students\n");
		printf(" 7. Delete the student details by roll number\n");
		printf(" 8. Update the student details by roll number\n");
		printf(" 9. Show all information\n");
		printf("10. Exit\n");
		printf("Enter your choice to perform the task:");
		fflush(stdin); fflush(stdout);
		scanf("%d",&x);

		//i know the switch statement is better in this situation but i used it much in
		//the functions so i used if here

		if(x==1)
		{
			add_student_manually();
		}

		else if(x==2)
		{
			add_student_file();
		}

		else if(x==3)
		{
			find_rl();
		}

		else if(x==4)
```

```
            {
                    find_fn();
            }

            else if(x==5)
            {
                    find_c();
            }

            else if(x==6)
            {
                    tot_s();
            }

            else if(x==7)
            {
                    del_s();
            }


            else if (x==8)
            {
                    up_s();
            }

            else if (x==9)
            {
                    show_s();
            }

            else if (x==10)
            {
                    printf("program is terminated !!");;
                    break;
            }

            else
            {
                    printf("please enter a valid option !!\n");
            }

    }
}
```

Database.h file

```c
/*
 * Database.h
 *
 *  Created on: Jun 26, 2022
 *      Author: momen
 */

#ifndef DATABASE_H_
#define DATABASE_H_

struct sinfo
{
	char  fname[50];
	char  lname[50];
	int   roll;
	float GPA;
	int   cid[10];
}st[55];

void add_student_manually();
void add_student_file();
void find_rl();
void find_fn();
void find_c();
void tot_s();
void del_s();
void up_s();
void show_s();

#endif /* DATABASE_H_ */
```

Database.c file

```c
/*
 * Database.c
 *
 *  Created on: Jun 26, 2022
 *      Author: momen
 */


#include"Database.h"
#include"FIFO.h"
#include"stdio.h"

extern FIFO_Buffer buffer;

void tot_s()
{
	//checking the LIFO buffer parameters for number of elements and empty space using count
	//and length
	printf("-------------------------\n");
	printf("[INFO] The total number of students is %d\n",buffer.count);
	printf("[INFO] You can add up to %d students\n",buffer.length);
	printf("[INFO] You can add %d more students\n",buffer.length-buffer.count);
	printf("-------------------------\n");
}

void add_student_manually()
{
	struct sinfo temp;
	int i;

	// using a straight forward approach getting the data from student bit by bit and store
	//it in the buffer
	printf("-------------------------\nAdd the student details\n-------------------------
\n");
	printf("Enter the roll number:");
	fflush(stdin); fflush(stdout);
	scanf("%d",&temp.roll);

	//this is a guard if the roll number already exist so we don't need to complete the
	//operation to get the remaining data from student, we simply send a feedback
	if (FIFO_check_duplicate(&buffer, temp.roll)==FIFO_DUPLICATE_ID)
	{
		printf("-------------------------\n");
		printf("[ERROR]  Roll number %d is already taken\n",temp.roll);
		printf("-------------------------\n");
	}

	//if the roll number is unique then we proceed to get the remaining data from student
	else
	{
		printf("Enter the first name of student:");
		fflush(stdin); fflush(stdout);
		gets(temp.fname);
		printf("Enter the last name of student:");
		fflush(stdin); fflush(stdout);
		gets(temp.lname);
		printf("Enter the GPA you obtained:");
		fflush(stdin); fflush(stdout);
		scanf("%f",&temp.GPA);

		printf("Enter the course ID of each course\n");
		for(i=0;i<5;i++)
		{
			printf("Course %d ID:",i+1);
			fflush(stdin); fflush(stdout);
			scanf("%d",&temp.cid[i]);
		}

		//calling the function to store the data into the buffer inside a switch body so
		//we can use the buffer states as feedback to output for the student if the
		//operation succeeded or not and the type of error if occurs

		switch (FIFO_enqueue(&buffer,&temp))
		{
		case FIFO_FULL:
			printf("-------------------------\n");
			printf("[ERROR]  Students database is full\n");
			printf("-------------------------\n");
			break;
		case FIFO_OK:
			printf("-------------------------\n");
			printf("[INFO]  Student details was added successfully\n");
			printf("-------------------------\n");
			break;
		default:
			printf("-------------------------\n");
			printf("[INFO]  Buffer is not initialized properly\n");
			printf("-------------------------\n");
		}
```

```
            //as required to show the total number of students after each successful student
            //entry

            tot_s();
      }

}

void add_student_file()
{
      int count;
      struct sinfo temp;
      FILE* fp = fopen("students.txt","r");

      //it's a simple way to use fopen, fscan ,and fclose to read from file the data stored in
      //a certain way

      if(fp)
      {
            while(fscanf(fp,"%d %s %s %f %d %d %d %d
%d",&temp.roll,temp.fname,temp.lname,&temp.GPA,
                              &temp.cid[0]
,&temp.cid[1],&temp.cid[2],&temp.cid[3],&temp.cid[4])!=EOF)
            {
                  //calling the function to store the data into the buffer inside a switch
                  //body so we can use the buffer states as feedback to output for the
                  //student if the operation succeeded or not and the type of error if occurs

                  switch (FIFO_enqueue(&buffer,&temp))
                  {
                  case FIFO_FULL:
                        printf("[ERROR]  Students database is full\n");
                        break;
                  case FIFO_OK:
                        printf("[INFO]  Roll number %d was saved successfully\n",temp.roll);
                        count++;
                        break;
                  case FIFO_DUPLICATE_ID:
                        printf("[ERROR]  Roll number %d is already taken\n",temp.roll);
                        break;
                  default:
                        printf("[INFO]  Buffer is not initialized properly\n");
                  }
            }

            //using count as an easy way to check if any entry was stored successfully from
            //the file guarding from wrong massage if all the entry from the file got rejected

            if(count) printf("[INFO]  Student details was added successfully\n");

            //as required to show the total number of students after each successful student
            //entry

            tot_s();
            fclose(fp);
      }

      else
      {
            printf("-------------------------\n");
            printf("[ERROR]  Students.text can't be opened\n");
            printf("-------------------------\n");
      }
}
```

```c
void show_s()
{
	//calling the function to print all the data stored in the buffer inside a switch body
	//so we can use the buffer states as feedback to output for the student if the operation
	//succeeded or not and the type of error if occurs

	switch (FIFO_print(&buffer))
	{
	case FIFO_EMPTY:
		printf("-------------------------\n");
		printf("[ERROR]  Students database is empty\n");
		printf("-------------------------\n");
		break;
	case FIFO_OK:
		printf("-------------------------\n");
		break;
	default:
		printf("-------------------------\n");
		printf("[INFO]  Buffer is not initialized properly\n");
		printf("-------------------------\n");
	}

}

void find_rl()
{
	int roll;

	//getting the roll number that will be checked against the database entries from the
	//student

	printf("Enter the roll number of the student:");
	fflush(stdin); fflush(stdout);
	scanf("%d",&roll);

	//calling the function to check the data stored in the buffer inside a switch body so we
	//can use the buffer states as feedback to output for the student if the operation
	//succeeded or not and the type of error if occurs

	switch(FIFO_find(&buffer,roll,NULL,0))
	{
	case FIFO_NO_ID:
		printf("-------------------------\n");
		printf("[ERROR]  Roll number %d was not found\n",roll);
		printf("-------------------------\n");
		break;
	case FIFO_EMPTY:
		printf("-------------------------\n");
		printf("[ERROR]  Students database is empty\n");
		printf("-------------------------\n");
		break;
	case FIFO_OK:
		printf("-------------------------\n");
		break;
	default:
		printf("-------------------------\n");
		printf("[INFO]  Buffer is not initialized properly\n");
		printf("-------------------------\n");
	}
}

void find_fn()
{
	char name[50];

	//getting the first name that will be checked against the database entries from the
	//student

	printf("Enter the first name of the student:");
	fflush(stdin); fflush(stdout);
	gets(name);

	//calling the function to check the data stored in the buffer inside a switch body so we
	//can use the buffer states as feedback to output for the student if the operation
	//succeeded or not and the type of error if occurs

	switch(FIFO_find(&buffer,0,name,0))
	{
	case FIFO_NO_ID:
		printf("-------------------------\n");
		printf("[ERROR]  First name %s was not found\n",name);
		printf("-------------------------\n");
		break;
	case FIFO_EMPTY:
		printf("-------------------------\n");
		printf("[ERROR]  Students database is empty\n");
		printf("-------------------------\n");
		break;
	case FIFO_OK:
		printf("-------------------------\n");
		break;
	default:
		printf("-------------------------\n");
		printf("[INFO]  Buffer is not initialized properly\n");
		printf("-------------------------\n");
	}
}

void find_c()
{
	int cid;

	//getting the course ID that will be checked against the database entries from the
	//student

	printf("Enter the course ID:");
	fflush(stdin); fflush(stdout);
	scanf("%d",&cid);

	//calling the function to check the data stored in the buffer inside a switch body so we
	//can use the buffer states as feedback to output for the student if the operation
	//succeeded or not and the type of error if occurs

	switch(FIFO_find(&buffer,0,NULL,cid))
	{
	case FIFO_NO_ID:
		printf("-------------------------\n");
		printf("[ERROR]  Course ID %d was not found\n",cid);
		printf("-------------------------\n");
		break;
	case FIFO_EMPTY:
		printf("-------------------------\n");
		printf("[ERROR]  Students database is empty\n");
		printf("-------------------------\n");
		break;
	case FIFO_OK:
		printf("-------------------------\n");
```

```c
			break;
		default:
			printf("-------------------------\n");
			printf("[INFO]  Buffer is not initialized properly\n");
			printf("-------------------------\n");
		}
}

void up_s()
{
		int roll;
		int choice;

		//getting the roll number that will be checked against the database entries from the
		//student

		printf("Enter the roll number of the student:");
		fflush(stdin); fflush(stdout);
		scanf("%d",&roll);

		//this is a guard if the roll number doesn't exist so we don't need to complete the
		//operation to get the remaining data from student, we simply send a feedback
		if (FIFO_check_duplicate(&buffer,roll)==FIFO_OK)
		{
			printf("-------------------------\n");
			printf("[ERROR]  Roll number %d was not found\n",roll);
			printf("-------------------------\n");
		}

		//if the roll number is found then we proceed to get the remaining data from student
		else
		{
			//a simple loop guarding from entering wrong options. although the method of
			//execution for this function is not perfect, the project video demanded it to be
			//done like showed so i adhered to what was asked

			while(1)
			{
				printf("1. first name\n2. last name\n3. roll number\n4. GPA\n5.
courses\n");
				fflush(stdin); fflush(stdout);
				scanf("%d",&choice);
				if(choice==1 || choice==2 || choice==3 || choice==4 || choice==5) break;
				printf("please enter a valid option !!\n");
			}

			//calling the function to update one student data stored in the buffer inside a
			//switch body so we can use the buffer states as feedback to output for the
			//student if the operation succeeded or not and the type of error if occurs

			switch(FIFO_update(&buffer,roll,choice))
			{
			case FIFO_EMPTY:
				printf("-------------------------\n");
				printf("[ERROR]  Students database is empty\n");
				printf("-------------------------\n");
				break;
			case FIFO_OK:
				printf("-------------------------\n");
				printf("[INFO]  Updated successfully\n");
				printf("-------------------------\n");
				break;
			default:
				printf("-------------------------\n");
```

```
					printf("[INFO]  Buffer is not initialized properly\n");
					printf("-------------------------\n");
			}
		}
}

void del_s()
{
		int roll;

		//getting the roll number that will be checked against the database entries from the
		//student

		printf("Enter the roll number which you want to delete:");
		fflush(stdin); fflush(stdout);
		scanf("%d",&roll);

		//calling the function to delete one student data stored in the buffer inside a switch
		//body so we can use the buffer states as feedback to output for the student if the
		//operation succeeded or not and the type of error if occurs

		switch(FIFO_delete(&buffer,roll))
		{
		case FIFO_NO_ID:
			printf("-------------------------\n");
			printf("[ERROR]  Roll number %d was not found\n",roll);
			printf("-------------------------\n");
			break;
		case FIFO_EMPTY:
			printf("-------------------------\n");
			printf("[ERROR]  Students database is empty\n");
			printf("-------------------------\n");
			break;
		case FIFO_OK:
			printf("-------------------------\n");
			printf("[INFO]  The roll number was removed successfully\n");
			printf("-------------------------\n");
			break;
		default:
			printf("-------------------------\n");
			printf("[INFO]  Buffer is not initialized properly\n");
			printf("-------------------------\n");
		}
}
```

FIFO.h file

```c
/*
 * FIFO.h
 *
 *  Created on: Jun 19, 2022
 *      Author: momen
 */

#ifndef FIFO_H_
#define FIFO_H_

#include"Database.h"

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

typedef enum
{
	FIFO_NULL, FIFO_OK, FIFO_FULL, FIFO_EMPTY, FIFO_DUPLICATE_ID, FIFO_NO_ID
}buffer_status;

buffer_status FIFO_find(FIFO_Buffer* buffer, int roll, char*name, int cid);
buffer_status FIFO_update(FIFO_Buffer* buffer, int roll, int choice);
buffer_status FIFO_init(FIFO_Buffer* buffer, struct sinfo* array, int length);
buffer_status FIFO_check_duplicate(FIFO_Buffer* buffer, int roll);
buffer_status FIFO_stat(FIFO_Buffer* buffer);
buffer_status FIFO_enqueue(FIFO_Buffer* buffer, struct sinfo* temp);
buffer_status FIFO_print(FIFO_Buffer* buffer);
buffer_status FIFO_delete(FIFO_Buffer* buffer, int roll);


#endif /* FIFO_H_ */
```

FIFO.c file

```c
/*
 * FIFO.c
 *
 *  Created on: Jun 26, 2022
 *      Author: momen
 */


#include "FIFO.h"
#include "stdio.h"
#include "string.h"


buffer_status FIFO_init(FIFO_Buffer* buffer, struct sinfo* array, int length)
{
	//this function should be called only once in the setup or the start of the program to
	//initialize a LIFO buffer based on an existing array of the same type.
	//it checks if the array is valid and put the parameters into the FIFO buffer

	if (array)
	{
		buffer->length = length;
		buffer->count = 0;
		buffer->base = array;
		buffer->head = array;
		buffer->tail = array;
		return FIFO_OK;
	}
	else return FIFO_NULL;
}

buffer_status FIFO_stat(FIFO_Buffer* buffer)
{
	//this function checks the buffer state to determine if that operations are done
	//successfully or not

	if(!buffer || !buffer->base || !buffer->head || !buffer->tail ) return FIFO_NULL;
	if(buffer->count == buffer->length) return FIFO_FULL;
	if(buffer->count == 0) return FIFO_EMPTY;
	else return FIFO_OK;
}

buffer_status FIFO_check_duplicate(FIFO_Buffer* buffer, int roll)
{
	//this function loops through all the database entries to check if the ID entered exist
	//already or not although the implementation is not perfect since it loops on all the
	//record, it will be called once or twice only

	int i;
	struct sinfo* p = buffer->tail;

	//this loop will work perfectly with way we use the FIFO since we don't dequeue from it
	//then the tail and head won't overlap with the end of the buffer but if we use the FIFO
	//as intended to get elements from it then must use a different loop first from tail to
	//end and then from base to head to surf through all the elements
	for(i=0;i<buffer->count;i++)
	{
		//if any entry has the same ID as the entered data, no need to continue the loop
		//we abort the operation altogether
		if(p->roll==roll) return FIFO_DUPLICATE_ID;
		p++;
	}

	//if the function loops through the whole buffer without finding any duplicated IDs then
	//we can proceed
	return FIFO_OK;
}

buffer_status FIFO_update(FIFO_Buffer* buffer, int roll, int choice)
{
	//this function searches the database for an ID which when found can be updated only in
	//one field which is not perfect but it was required like this

	//checking the buffer states before operation to preserve time if it was empty then
	//there is no entry to update so we abort the operation altogether

	if(FIFO_stat(buffer) == FIFO_NULL) return FIFO_NULL;
	if(FIFO_stat(buffer) != FIFO_EMPTY)
	{
		int i;
		int x;
		struct sinfo* p = buffer->tail;
	//this loop will work perfectly with way we use the FIFO since we don't dequeue from it
	//then the tail and head won't overlap with the end of the buffer but if we use the FIFO
	//as intended to get elements from it then must use a different loop first from tail to
	//end and then from base to head to surf through all the elements
		for(i=0;i<buffer->count;i++)
		{
			if(roll==p->roll)
			{
				//if the ID required if found then we proceed to edit only one
				//parameter of the data
				switch(choice)
				{
				case 1:
					printf("Enter the new first name: ");
					fflush(stdin); fflush(stdout);
					gets(p->fname);
					break;
				case 2:
					printf("Enter the new last name: ");
					fflush(stdin); fflush(stdout);
					gets(p->lname);
					break;
				case 3:
					printf("Enter the new roll number: ");
					fflush(stdin); fflush(stdout);
					scanf("%d",&p->roll);
					break;
				case 4:
					printf("Enter the new GPA: ");
					fflush(stdin); fflush(stdout);
					scanf("%f",&p->GPA);
					break;
				case 5:
					printf("Enter the new courses IDs: ");
					for(x=0;x<5;x++)
						printf("Course %d ID:",i+1);
					fflush(stdin); fflush(stdout);
					scanf("%d",&p->cid[x]);
					break;
				}
				return FIFO_OK;
			}
			p++;
		}
		//if the function loops through the whole buffer without finding any matching ID
		//then it's not in the database
		return FIFO_NO_ID;
	}
	return FIFO_EMPTY;
}

buffer_status FIFO_find(FIFO_Buffer* buffer, int roll, char* name, int cid)
{
	//this function can be used to find a record in the database using three different
	//methods either by a roll number or by a first name or by a course id
	//checking the buffer states before operation to preserve time if it was empty then
	//there is no entry to find so we abort the operation altogether

	if(FIFO_stat(buffer) == FIFO_NULL) return FIFO_NULL;
	if(FIFO_stat(buffer) != FIFO_EMPTY)
	{
		int i;
		int x;
		int y=0;
```

```
        int count=0;
        struct sinfo* p = buffer->tail;

    //this loop will work perfectly with way we use the FIFO since we don't dequeue from it
    //then the tail and head won't overlap with the end of the buffer but if we use the FIFO
    //as intended to get elements from it then must use a different loop first from tail to
    //end and then from base to head to surf through all the elements
        for(i=0;i<buffer->count;i++)
        {
            //this function is very delicate that is checks for the method to follow so
            //it proceeds only in one direction disregarding the others so if the
            //search is by roll number it will check for the matching between the IDs
            //if the search is by first name it will compare the entered name with the
            //records if the search is by course ID it will compare the entered ID with
            //all the stored courses in the records if an ID match if found or many
            //name matches are found or many course //matches are found it proceeds

            if (roll && p->roll==roll) y=1;
            else if (!roll && !cid && !stricmp(name,p->fname)) y=1;
            else if (cid)
            {
                for(x=0;x<5;x++)
                {
                    if (cid == p->cid[x]) y = 1;
                }
            }

            if(y)
            {
                //to show the found data in the records we can divide the data into
                //two parts, one is mutual between the three methods, and another is
                //unique to each method (if the search is by course ID then we don't
                //show all the courses for the student attending that course, or if
                //the search is by first name we can have multiple results so we
                //continue the loop, but if the search is but roll number we break
                //the loop immediately as it's unique so there is no matchings)

                printf("-------------------------\n");
                printf("The student details are\n");
                printf("Student first name: %s\n",p->fname);
                printf("Student last name: %s\n",p->lname);
                printf("Student roll number: %d\n",p->roll);
                printf("Student GPA number: %0.2f\n",p->GPA);

                if(!cid)
                {
                    for(x=0;x<5;x++)
                    {
                        printf("The course ID: %d\n",p->cid[x]);
                    }
                    if(roll)return FIFO_OK;
                }
                //to enter another loop from the start we must zero the flag but
                //save the count for multiple matches
                y=0;
                count ++;
            }
            p++;
        }
        //we can have two outcomes from the loop if the search is done by roll number then
        //no ID were matching since using search by roll number method will end the
        //function if matches were found but if the search is done by first name or course
        //ID then we check for count if zero then no matches were found so if count is not
```

```
            //zero that means matches were found either by first name or course ID method so
            //we proceed
            if(!count) return FIFO_NO_ID;
            return FIFO_OK;
      }
      return FIFO_EMPTY;
}

buffer_status FIFO_enqueue(FIFO_Buffer* buffer, struct sinfo* temp)
{
      //this function adds an entry in the database
      //checking the buffer states before operation to preserve time if it was full then there
      //is no space to store into so we abort the operation altogether

      if(FIFO_stat(buffer) == FIFO_NULL) return FIFO_NULL;
      if(FIFO_stat(buffer) != FIFO_FULL)
      {
            //checks first if the ID exist already in the database if not then proceed to
            //enter the student data

            if(FIFO_check_duplicate(buffer,temp->roll)== FIFO_OK)
            {
                  *buffer->head = *temp;
                  buffer->count++;

                  //to check if the FIFO buffer rolled over the array space if head reached
                  //the end and there is empty space at the start to store at, but we don't
                  //use the dequeue function of the buffer so it doesn't matter we will not
                  //ever roll over the buffer and seriously i don't know why we used a FIFO
                  //buffer for database if we don't use if to get the data stored in order in
                  //the first place !!!!
                  if ((unsigned int)buffer->head == ((unsigned int)buffer->base + (buffer-
>length-1)))
                        buffer->head = buffer->base;
                  else buffer->head++;

                  return FIFO_OK;
            }
            else return FIFO_DUPLICATE_ID;
      }
      else return FIFO_FULL;
}


buffer_status FIFO_print(FIFO_Buffer* buffer)
{
      //this function shows all the entries stored in the database
      //checking the buffer states before operation to preserve time if it was empty then
      //there is no entries to show so we abort the operation altogether

      if(FIFO_stat(buffer) == FIFO_NULL) return FIFO_NULL;
      if(FIFO_stat(buffer) != FIFO_EMPTY)
      {
            printf("\n");
            int i;
            struct sinfo* p = buffer->tail;

      //this loop will work perfectly with way we use the FIFO since we don't dequeue from it
      //then the tail and head won't overlap with the end of the buffer but if we use the FIFO
      //as intended to get elements from it then must use a different loop first from tail to
      //end and then from base to head to surf through all the elements
            for(i=0;i<buffer->count;i++)
            {
                  int x;
```

```
                printf("--------------------------\n");
                printf("Student first name: %s\n",p->fname);
                printf("Student last name: %s\n",p->lname);
                printf("Student roll number: %d\n",p->roll);
                printf("Student GPA number: %0.2f\n",p->GPA);
                for(x=0;x<5;x++)
                {
                        printf("The course ID: %d\n",p->cid[x]);
                }
                p++;
            }
            return FIFO_OK;
        }
        else return FIFO_EMPTY;
}


buffer_status FIFO_delete(FIFO_Buffer* buffer, int roll)
{
        //this function deletes the entry by roll number if found in the database by a simple
        //technique checking the buffer states before operation to preserve time if it was empty
        //then there is no entries to find so we abort the operation altogether

        if(FIFO_stat(buffer) == FIFO_NULL)return FIFO_NULL;
        if(FIFO_stat(buffer) != FIFO_EMPTY)
        {
            int i;
            struct sinfo* p = buffer->tail;
            struct sinfo* temp;

        //this loop will work perfectly with way we use the FIFO since we don't dequeue from it
        //then the tail and head won't overlap with the end of the buffer but if we use the FIFO
        //as intended to get elements from it then must use a different loop first from tail to
        //end and then from base to head to surf through all the elements
            for(i=0;i<buffer->count;i++)
            {
                if(p->roll==roll)
                {
                    //if a roll number match is found we simply move all the next
                    //entries a step early overwriting the entry be deleted and also
                    //decrease the count and the buffer head by one
                    for(temp=p;p<buffer->head;temp++)
                    {
                        *temp=*(++p);
                    }

                    buffer->head--;
                    buffer->count--;
                    return FIFO_OK;
                }
                p++;
            }
            //if the function loops through the whole buffer without finding any matching ID
            //then it's not in the database
            return FIFO_NO_ID;
        }
        return FIFO_EMPTY;
}
```