PROJECT 2

Student Management System

Presented By

Mohammed Hassan

Presented To

Eng.Keroles Shenouda

Contact Information

GitHub: Click Here

My Drive: Click Here

My progress Page: Click Here

mohasanbder@gmail.com

Table of Contents

- ***Problem Statement.....3***
- ***Approach.....3***
- ***Student_System.h 4***
  - ***1- Student Structure 5***
  - ***2- Database Structure.....5***
  - ***3- Function Prototypes 6***
  - ***4- Buffer Status.....6***
- ***Main.c 7***
- ***Student_System.c.....8***

1- Student_System_init.....8

2- Add Student Manually 9

- ***Check Student ID 10***
- ***FIFO Enqueue 11***
- ***Check FIFO Status 12***

3- Add Student from File.....13

4- Find Student by Roll Number.....14

5- Find Student by First Name 15

6- Find Student by Course.....16

7- Delete Student by ID Number.....17

8- Update Student by id 18

9- Show All Student Information.....19

10- Print total Student Number 20

11- Print Student information 20

❖ *Program Execution Snapshots 21*

- *Add Student Manually 21*
- *Add Student from File 22*
- *Find Student Details by ID Number 23*
- *Find Student details by First Name 24*
- *Find Student Details by Course ID 25*
- *Print Total Number of Students 26*
- *Delete Student by Roll Number 27*
- *Update Student Details by Roll Number 28*

1- If Student ID Number Not found: 28

2- Update Student id: 29

3- update Student id failed 30

4- update Student First Name 31

5- update Student Last Name 32

6- Update Student GPA 33

7- update Student Course ID Number 34

8- Show All Details information of Students 35

9- Exit the Program 36

❖ Problem Statement

Write a Program to build a simple software for Student Information Management System which can perform the following operations:

1. **Store the First name of the student.**
2. **Store the Last name of the student.**
3. **Store Unique ID number for each student.**
4. **Store GPA for each student.**
5. **Store Courses Registered by the student.**

❖ Approach

There are 10 options to form an individual functions for every operations use it to handle Student Information.

1. **Add the Student Details Manually.**
2. **Add the Student Details Form File.**
3. **Find the Student Details by given id Number of Student.**
4. **Find the Student Details by Given Student First Name.**
5. **Find the Student Details by Given Student Last Name.**
6. **Count total Number of Students.**
7. **Delete Specific Student by Given his ID Number.**
8. **Update Student Information by Given his ID Number.**
    a. **Change Student ID Number.**
    b. **Change Student First Name.**
    c. **Change Student Last Name**
    d. **Change Student Courses.**
9. **Print All Student Information.**
10. **Exit The Program.**

❖ Student_System.h

```c
/*
 ================================================================================
 Name        : Student_System.h
 Author      : Mohammed Hassan
 Created on  : Oct 30, 2023
 Description : First Term Project 2
 ================================================================================
 */

#ifndef STUDENT_SYS_H_
#define STUDENT_SYS_H_

#include <stdio.h>
#include <stdint.h>
#include <string.h>
// ===========>> printf <<===========
#define print(...)      \
                            fflush(stdin);fflush(stdout); \
                            printf(__VA_ARGS__);          \
                            fflush(stdin);fflush(stdout);


// ===========>> Macros <<===========
#define Name_Length     20
#define Number_Courses  5
#define Max_Number_Students  50

// ===========>> Student information <<===========
struct Studen_info_t
{
    char first_name[Name_Length];
    char last_name[Name_Length];
    int roll ;
    float GPA ;
    int Course_id[Number_Courses];
};
typedef struct Studen_info_t SStudent_Node ;

// ===========>> Type Definition <<===========
typedef struct
{
    SStudent_Node* base ;
    SStudent_Node* head ;
    SStudent_Node* tail ;
    uint32_t length ;
    uint32_t count ;
} FIOF_Buf_t;

// ===========>> Buf Status <<===========
typedef enum
{
    FIFO_no_error,
    FIFO_full,
    FIFO_empty,
    FIFO_Null
} FIFO_Buf_Status;

// ==============>> APIs <<==============

FIFO_Buf_Status FIFO_init(FIOF_Buf_t* Student_Queue, SStudent_Node* Buf, uint32_t length );     // Student Queue Initialization
FIFO_Buf_Status FIFO_enqueue(FIOF_Buf_t* Student_Queue, SStudent_Node Student);
FIFO_Buf_Status FIFO_is_Full(FIOF_Buf_t* Student_Queue);                                        // Check FIFO Full

void Add_Student_manually(FIOF_Buf_t* Student_Queue);

void Add_Student_From_File(FIOF_Buf_t* Student_Queue);

void Find_Student_By_id(FIOF_Buf_t* Student_Queue);

void Find_Student_By_FirstName(FIOF_Buf_t *Student_Queue);

void Find_Student_By_Course_id(FIOF_Buf_t *Student_Queue);

void Delete_Student_By_id(FIOF_Buf_t *Student_Queue);

void Update_Student_By_id(FIOF_Buf_t *Student_Queue);

void Print_Student_Count(FIOF_Buf_t *Student_Queue);                                            // print total Number of Students
void Show_Student_Information(FIOF_Buf_t *Student_Queue);                                       // View all Students information
void print_Student_info(SStudent_Node *student);                                                // print information of Student

struct Studen_info_t *Check_Student_id(FIOF_Buf_t *Student_Queue, uint32_t id);                 // Check if student is exist or not

#endif /* STUDENT_SYS_H_ */
```

1-Student Structure

```c
// ===========>> Student information <<===========
struct Studen_info_t
{
    char first_name[Name_Length];
    char last_name[Name_Length];
    int roll ;
    float GPA ;
    int Course_id[Number_Courses];
};
typedef struct Studen_info_t SStudent_Node ;
```

2-Database Structure

```c
// ===========>> Type Definition <<===========
typedef struct
{
    SStudent_Node* base ;
    SStudent_Node* head ;
    SStudent_Node* tail ;
    uint32_t length ;
    uint32_t count ;
} FIOF_Buf_t;
```

3-Function Prototypes

```c
FIFO_Buf_Status FIFO_init(FIOF_Buf_t* Student_Queue, SStudent_Node* Buf, uint32_t length );     // Student Queue Initialization
FIFO_Buf_Status FIFO_enqueue(FIOF_Buf_t* Student_Queue, SStudent_Node Student);
FIFO_Buf_Status FIFO_is_Full(FIOF_Buf_t* Student_Queue);                                        // Check FIFO Full

void Add_Student_manually(FIOF_Buf_t* Student_Queue);

void Add_Student_From_File(FIOF_Buf_t* Student_Queue);

void Find_Student_By_id(FIOF_Buf_t* Student_Queue);

void Find_Student_By_FirstName(FIOF_Buf_t *Student_Queue);

void Find_Student_By_Course_id(FIOF_Buf_t *Student_Queue);

void Delete_Student_By_id(FIOF_Buf_t *Student_Queue);

void Update_Student_By_id(FIOF_Buf_t *Student_Queue);

void Print_Student_Count(FIOF_Buf_t *Student_Queue);                                            // print total Number of Students
void Show_Student_Information(FIOF_Buf_t *Student_Queue);                                       // View all Students information
void print_Student_info(SStudent_Node *student);                                                // print information of Student

struct Studen_info_t *Check_Student_id(FIOF_Buf_t *Student_Queue, uint32_t id);                 // Check if student is exist or not

#endif /* STUDENT_SYS_H_ */
```

4-Buffer Status

```c
// ============>> Buf Status <<============
typedef enum
{
    FIFO_no_error,
    FIFO_full,
    FIFO_empty,
    FIFO_Null
} FIFO_Buf_Status;
```

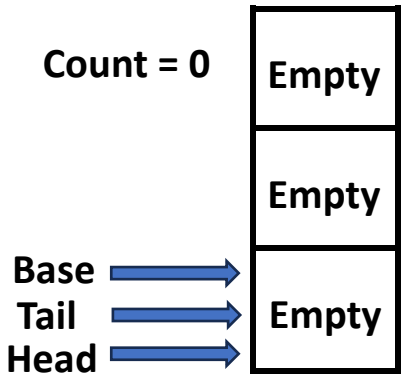
❖ Main.c

```c
/*
 ============================================================================
 Name        : main.c
 Author      : Mohammed Hassan
 Created on  : Oct 30, 2023
 Description : First Term Project 2
 ============================================================================
 */
#include "Student_System.h"
int main()
{
    int Select_Option;
    FIOF_Buf_t Student_Queue;
    struct Studen_info_t Student_Buffer[Max_Number_Students];

    FIFO_init(&Student_Queue,Student_Buffer,Max_Number_Students);
    while (1)
    {
        print("\n\t\t\t\t*****************************************\n");
        print("\t\t\t\t* Choose the Task that you want to Perform *");
        print("\n\t\t\t\t*****************************************\n\n");
        print(" 1- Add the Student Details Manually\n\n");
        print(" 2- Add the Student Details From Text File\n\n");
        print(" 3- Find the Student Details by Roll Number\n\n");
        print(" 4- Find the Student Details by First Name\n\n");
        print(" 5- Find the Student Details by Course id\n\n");
        print(" 6- Find the Total Number of Students\n\n");
        print(" 7- Delete the Students Details by Roll Number\n\n");
        print(" 8- Update the students Details by Roll Number\n\n");
        print(" 9- Show All Information\n\n");
        print(" 10- Exit\n\n");
        print(" Enter your Choice to Perform the task: ");

        scanf("%d", &Select_Option);
        print("\n=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*\n\n");
        switch (Select_Option)
        {
        case 1:
            Add_Student_manually(&Student_Queue);
            break;
        case 2:
            Add_Student_From_File(&Student_Queue);
            break;
        case 3:
            Find_Student_By_id(&Student_Queue);
            break;
        case 4:
            Find_Student_By_FirstName(&Student_Queue);
            break;
        case 5:
            Find_Student_By_Course_id(&Student_Queue);
            break;
        case 6:
            Print_Student_Count(&Student_Queue);
            break;
        case 7:
            Delete_Student_By_id(&Student_Queue);
            break;
        case 8:
            Update_Student_By_id(&Student_Queue);
            break;
        case 9:
            Show_Student_Information(&Student_Queue);
            break;
        case 10:
            print("\t\t\t\t ***************** Thank you *****************\n");
            return 0;
            break;
        default:
            print("\n Wrong Option Try Again\n");
            break;
        }
    }
    return 0;
}
```

❖ Student_System.c

1- Student_System_init

- The function will Start by checking the input parameters validity.
- Start Initialize FIFO.




```c
// ==============>> Initialize FIFO Buffer <<==============
FIFO_Buf_Status FIFO_init(FIOF_Buf_t *Student_Queue, SStudent_Node *Buf, uint32_t length)
{
    if (Buf == NULL)
    {
        return FIFO_Null;
    }
    // Initialize Students Data Base
    Student_Queue->base = Buf;
    Student_Queue->head = Buf;
    Student_Queue->tail = Buf;
    Student_Queue->length = length;
    Student_Queue->count = 0;
    return FIFO_no_error;
}
```

2- Add Student Manually

Take id number form user.

Check if this id is existed or not.

If not, start taking data from user.

After getting all data add it to the

```
// ==============>> Add Student Manually <<==============
void Add_Student_manually(FIOF_Buf_t *Student_Queue)
{
    SStudent_Node New_Student;
    int i;

    // *Check if Student Data Base is valid
    if (!Student_Queue->base || !Student_Queue->head || !Student_Queue->tail)
    {
        print("\n*****************************************************\n");
        print("* [ERROR] Student Data Base Initialization ==>> Failed\n");
        print("*****************************************************\n");
        return;
    }

    // *Check if Student Data Base is Full
    if (Student_Queue->count == Student_Queue->length)
    {
        print("*******************************************\n");
        print("* [ERROR] Student Data Base is ==>> Full\n");
        print("*******************************************\n");
        return;
    }

    // * ADD STUDENT INFORMATION
    print("\t\t\t=================================================\n");
    print("\t\t\t==>>            Enter Student Data            <<==\n");
    print("\t\t\t=================================================\n\n");

    print(" Enter Student id : ");
    scanf("%d", &New_Student.roll);

    if (Check_Student_id(Student_Queue, New_Student.roll))
    {
        print("\n*****************************************************\n");
        print("* [ERROR] Roll Number %d is Already Taken Before\n", New_Student.roll);
        print("* [ERROR] Adding Student Manually Failed\n");
        print("*****************************************************\n");
        return ;
    }
    else
    {
        print(" Enter Student First Name : ");
        scanf("%s", New_Student.first_name);

        print(" Enter Student Last Name : ");
        scanf("%s", New_Student.last_name);

        print(" Enter Student GPA : ");
        scanf("%f", &New_Student.GPA);

        for (i = 0; i < Number_Courses; i++)
        {
            print(" Enter Course id Number %d : ", i + 1);
            scanf("%d", &New_Student.Course_id[i]);
        }
        if (FIFO_enqueue(Student_Queue, New_Student) == FIFO_no_error)
        {
            print("\n*******************************************\n");
            print("* [INFO] Student Details is Added Successfully");
        }
        else
        {
            print("*******************************************\n");
            print("* [ERROR] Adding Student Manually Failed\n");
            print("*******************************************\n");
            return;
        }
    }
    Print_Student_Count(Student_Queue);
}
```

➢ Check Student ID

- ➢ Scan if Student id existed or not.
- ➢ If not, return NULL.

```c
// ==============>> Scan if Student id Existed or not <<==============
struct Studen_info_t *Check_Student_id(FIOF_Buf_t *Student_Queue, uint32_t id)
{
    int i;
    SStudent_Node *New_Student = Student_Queue->tail;

    for (i = 0; i < Student_Queue->count; i++)
    {
        if (New_Student->roll == id)
        {
            return New_Student;
        }
        else
        {
            New_Student++;
        }
    }
    New_Student = NULL;
    return New_Student;
}
```

➢ FIFO Enqueue

- ➢ Check buffer validity.
- ➢ Check if FIFO is Full.
- ➢ Add all data to the Buffer.
- ➢ Increment counter.
- ➢ Chek if the head at the last element in the array.

```c
// ==============>> FIFO Equeue <<==============
FIFO_Buf_Status FIFO_enqueue(FIOF_Buf_t *Student_Queue, SStudent_Node Student)
{
    // Check FIFO Valid
    if (!Student_Queue->base || !Student_Queue->head || !Student_Queue->tail)
    {
        return FIFO_Null;
    }

    // FIFO is Full
    if (FIFO_is_Full(Student_Queue) == FIFO_full)
    {
        return FIFO_full;
    }

    *(Student_Queue->head) = Student;
    Student_Queue->count++;
    // Check if The head at the last element in the array.

    if (Student_Queue->head == (Student_Queue->base + (Student_Queue->length * sizeof(SStudent_Node))))
    {
        Student_Queue->head = Student_Queue->base;
    }
    else
    {
        Student_Queue->head++;
    }

    return FIFO_no_error;
}
```

➢ Check FIFO Status

```c
// ==============>> Check FIFO Status <<==============
FIFO_Buf_Status FIFO_is_Full(FIOF_Buf_t *Student_Queue)
{
    // Check FIFO Valid
    if (!Student_Queue->base || !Student_Queue->head || !Student_Queue->tail)
    {
        print("*********************************************\n");
        print("* [ERROR] FIFO Enqueue Failed ==>> FIFO is Null\n");
        print("*********************************************\n");
        return FIFO_Null;
    }

    if (Student_Queue->count == Student_Queue->length)
    {
        print("*********************************************\n");
        print("* [ERROR] FIFO Enqueue Failed ==>> FIFO is Full\n");
        print("*********************************************\n");
        return FIFO_full;
    }

    return FIFO_no_error;
}
```

3- Add Student from File

- open Stydent_Data.txt with option read only.
- Check if the file is existed and include data or not.
- Reading file information until end of file using while loop.
- Start reading data from file and check if Student id exists before or not.

```c
// ==============>> Add Student From File <<==============
void Add_Student_From_File(FIOF_Buf_t *Student_Queue)
{
    FILE *Student_file;
    SStudent_Node New_Student;
    int i;

    //* Open Text file
    Student_file = fopen("Student_Data.txt", "r");

    //* check if this file is exist or not
    if (Student_file == NULL)
    {
        print("***********************************************\n");
        print("* [ERROR] this File not fount\n");
        print("* [ERROR] Add Student From File Failed\n");
        print("***********************************************\n");
        return;
    }
    //* reading file information until end of file
    while (!feof(Student_file))
    {
        //* Reading Student id from file
        fscanf(Student_file, "%d", &New_Student.roll);
        if (Check_Student_id(Student_Queue, New_Student.roll))
        {
            print("***********************************************\n");
            print("* [ERROR] Roll Number %d is Already Taken Before\n", New_Student.roll);
            print("***********************************************\n\n");
            fscanf(Student_file, "%*[^\n]");
            continue ;
        }

        fscanf(Student_file, "%s", New_Student.first_name);
        fscanf(Student_file, "%s", New_Student.last_name);
        fscanf(Student_file, "%f", &New_Student.GPA);

        for(i = 0 ; i < Number_Courses ; i++)
        {
            fscanf(Student_file, "%d", &New_Student.Course_id[i]);
        }

        if(FIFO_enqueue(Student_Queue, New_Student) == FIFO_no_error)
        {
            print("* [INFO] Roll Number %d is Saved Successfully\n", New_Student.roll);
        }
        else
        {
            print("***********************************************\n");
            print("* [ERROR] Adding Student From File Failed\n");
            print("***********************************************\n");
            return ;
        }
    }
    print("* [Info] Student Details Are Saved Successfully ");
    Print_Student_Count(Student_Queue);

    //* Close text file
    fclose(Student_file);
}
```

4- Find Student by Roll Number

- Check if there is Students in the queue or not.
- Take an ID from the user to find it.
- Check if this id exists before.
- If Student was found print Student's information.

```c
// ==============>> Find Student by id <<==============
void Find_Student_By_id(FIOF_Buf_t* Student_Queue)
{
    int Student_id;
    SStudent_Node* Student ;
    // *Check if Student Data Base is Empty
    if (Student_Queue->count == 0)
    {
        print("*********************************************\n");
        print("* [ERROR] Student Data Base is Empty\n");
        print("*********************************************\n");
        return;
    }
    print("\n*********************************************\n");
    print("\nEnter Student id number you want to find :")
    scanf("%d", &Student_id);


    Student = Check_Student_id(Student_Queue, Student_id);
    if(Student == NULL)
    {
        print("*********************************************\n");
        print("* [ERROR] Student id number %d not found \n", Student_id);
        print("*********************************************\n");
        return ;
    }
    else
    {
        print("Student id : %d is Found Successfully\n", Student_id);
        print_Student_info(Student);
    }
}
```

5- Find Student by First Name

```c
// ==============>> Find Student by First Name  <<==============
void Find_Student_By_FirstName(FIOF_Buf_t *Student_Queue)
{
    SStudent_Node* Student = Student_Queue->tail ;
    char First_Name[Name_Length] ;
    int i, flag = 0 ;

    // *Check if Student Data Base is Empty
    if (Student_Queue->count == 0)
    {
        print("*********************************************\n");
        print("* [ERROR] Student Data Base is Empty\n");
        print("*********************************************\n");
        return;
    }
    print("Enter Student First Name :")
    scanf("%s",First_Name);

    for(i = 0 ; i < Student_Queue->count ; i++)
    {
        if(strcmp(First_Name, Student->first_name) == 0)
        {
            print("\n*********************************************\n");
            print("Student : %s is found Successfully\n",First_Name);
            print_Student_info(Student);
            flag = 1 ;
            return ;
        }
        // Check if we reach the last item in the queue
        if((Student + 1) == (Student_Queue->base + Student_Queue->length))
        {
            Student = Student_Queue->base;
        }
        else
        {
            Student++;
        }
    }
    if(!flag)
    {
        print("*********************************************\n");
        print("* [ERROR] Student %s Not found \n", First_Name );
        print("*********************************************\n");
        return ;
    }

}
```

6- Find Student by Course

```c
// ==============>> Find Student by course id <<==============
void Find_Student_By_Course_id(FIOF_Buf_t *Student_Queue)
{
    SStudent_Node* Student = Student_Queue->tail ;
    int i, j,counter = 0, Course_id;

    if (Student_Queue->count == 0)
    {
        print("***********************************************\n");
        print("* [ERROR] Student Data Base is Empty\n");
        print("***********************************************\n");
        return;
    }

    print("\nEnter course id: ");
    scanf("%d", &Course_id);

    for(i = 0 ; i < Student_Queue->count ; i++)
    {
        for(j = 0 ; j < Number_Courses ; j++)
        {
            if(Student->Course_id[j] == Course_id)
            {
                print_Student_info(Student);
                counter++ ;
            }
        }
        Student++;
    }
    if(counter == 0)
    {
        print("***********************************************\n");
        print("* [ERROR] Course ID %d Not found on the System\n", Course_id);
        print("***********************************************\n");
        return;
    }
    else
    {
        print("*************************************************************\n");
        print("[INFO] Number of Students enrolled in this course is : %d\n", counter);
        print("*************************************************************\n");
    }
}
```

7- Delete Student by ID Number

```c
// ===============>> Delete Student By id <<==============
void Delete_Student_By_id(FIOF_Buf_t *Student_Queue)
{
    int Student_id, i, flag = 0 ;
    SStudent_Node* Student = Student_Queue->base ;

    if (Student_Queue->count == 0)
    {
        print("*********************************************\n");
        print("* [ERROR] Student Data Base is Empty\n");
        print("*********************************************\n");
        return;
    }

    print("Enter Student id to Delete it from the system: ");
    scanf("%d", &Student_id);

    for(i = 0 ; i < Student_Queue->count ; i++)
    {
        if(Student->roll == Student_id)
        {
            *Student = *(Student_Queue->tail);
            Student_Queue->count-- ;

            if( (Student_Queue->tail+1) == (Student_Queue->base + Student_Queue->length) )
            {
                Student_Queue->tail = Student_Queue->base ;
            }
            else
            {
                Student_Queue->tail++;
            }
            flag = 1 ;
            break ;
        }

        else
        {
            flag = 0 ;
        }
        Student++ ;
    }

    if(flag == 1)
    {
        print("************************************************\n");
        print("* [INFO] Student ID Number %d Removed Successfully\n", Student_id);
        print("************************************************\n");
    }
    else
    {
        print("*********************************************\n");
        print("* [ERROR] Student ID Number %d Not found\n",Student_id );
        print("*********************************************\n");
    }
}
```

8- Update Student by id

```c
// ==============>> Update Student Data <<==============
void Update_Student_By_id(FIOF_Buf_t *Student_Queue)
{
    SStudent_Node* Student = Student_Queue->tail;
    SStudent_Node *Student_Data_mod ;
    int option_number, Student_id, New_Student_id, i, flag = 0;

    if (Student_Queue->count == 0)
    {
        print("************************************************\n");
        print("* [ERROR] Student Data Base is Empty\n");
        print("************************************************\n");
        return;
    }
    print(" Enter Student id to update data: ");
    scanf("%d", &Student_id);

    Student_Data_mod = Check_Student_id(Student_Queue, Student_id);
    if(Student_Data_mod != NULL)
    {
        print("\n\t\t***************\n");
        print("\t\t* Student Data *\n");
        print("\t\t***************\n");
        print_Student_info(Student_Data_mod);
        print("\n************************************************\n");
    }
    else
    {
        print("************************************************\n");
        print("* [ERROR] Student ID number %d Not found\n", Student_id);
        print("************************************************\n");
        return ;
    }

    print(" What do you want to change\n");
    print(" 1: Student ID Number\n");
    print(" 2: Student First Name\n");
    print(" 3: Student Last Name\n");
    print(" 4: Student GPA\n");
    print(" 5: Student Course ID\n\n");
    print("Enter the Number: ");
    scanf("%d", &option_number);

    switch(option_number)
    {
    case 1:
        print(" \nEnter the New Student ID: ");
        scanf("%d", &New_Student_id);

        for(i = 0 ; i < Student_Queue->count ; i++)
        {
            while(Student->roll == New_Student_id)
            {
                print("************************************************\n");
                print("* [ERROR] Roll Number %d is Exist\n", New_Student_id);
                print("************************************************\n");
                flag = 1 ;
                break ;
            }
            if((Student + 1) == (Student_Queue->base + Student_Queue->length))
            {
                Student = Student_Queue->base;
            }
            else
            {
                Student++;
            }
        }
        if(flag == 0)
        {
            Student_Data_mod->roll = New_Student_id ;

            print("\n* [INFO] Student ID Number %d Updated Successfully\n\n",New_Student_id);

            print("************************************************\n");
            print("************ Student Data After update ************\n");
            print_Student_info(Student_Data_mod);
        }
        break;
    case 2:
        print(" Enter The New Student First Name: ");
        scanf("%s",Student_Data_mod->first_name);

        print("* [INFO] Student %s Updated Successfully\n\n",Student_Data_mod->first_name);

        print("************************************************\n");
        print("********** Student Data After update **********\n");
        print_Student_info(Student_Data_mod);
        break ;
    case 3:
        print(" Enter The New Student Last Name: ");
        scanf("%s",Student_Data_mod->last_name);

        print("* [INFO] Student Last Name %s Updated Successfully\n\n",Student_Data_mod->last_name);

        print("************************************************\n");
        print("********** Student Data After update **********\n");
        print_Student_info(Student_Data_mod);
        break ;
    case 4:
        print(" Enter The New Student GPA: ");
        scanf("%f",&Student_Data_mod->GPA);

        print("* [INFO] Student GPA %.2f Updated Successfully\n\n",Student_Data_mod->GPA);

        print("************************************************\n");
        print("********** Student Data After update **********\n");
        print_Student_info(Student_Data_mod);
        break;
    case 5:
        print(" Enter Course Number From 1 to %d to update: ", Number_Courses);
        scanf("%d", &option_number);
        print(" Enter New Course ID: ");
        scanf("%d", &Student_Data_mod->Course_id[option_number - 1]);

        print("* [INFO] Student Course ID Number %d Updated Successfully\n\n",Student_Data_mod->Course_id[option_number - 1]);

        print("************************************************\n");
        print("********** Student Data After update **********\n");
        print_Student_info(Student_Data_mod);
        break;
    default:
        print("----------------------\n");
        print("* [ERROR]Wrong Option\n");
        print("======================\n");
        break;
    }
}
```

9- Show All Student Information

```c
// ==============>> Show All Student Information <<==============
void Show_Student_Information(FIOF_Buf_t *Student_Queue)
{
    int i;
    SStudent_Node *Student_info;

    // *Check if Student Data Base is Empty
    if (Student_Queue->count == 0)
    {
        print("*********************************************\n");
        print("* [ERROR] Student Data Base is Empty\n");
        print("*********************************************\n");
        return;
    }

    // *Check if Student Data Base is valid
    if (!Student_Queue->base || !Student_Queue->head || !Student_Queue->tail)
    {
        print("*********************************************\n");
        print("* [ERROR] Student Data Base Initialization ==>> Failed\n");
        print("*********************************************\n");
        return;
    }

    Student_info = Student_Queue->tail;

    for (i = 0; i < Student_Queue->count; i++)
    {
        print_Student_info(Student_info);
        Student_info++;
    }
    Print_Student_Count(Student_Queue);
}
```

10- Print total Student Number

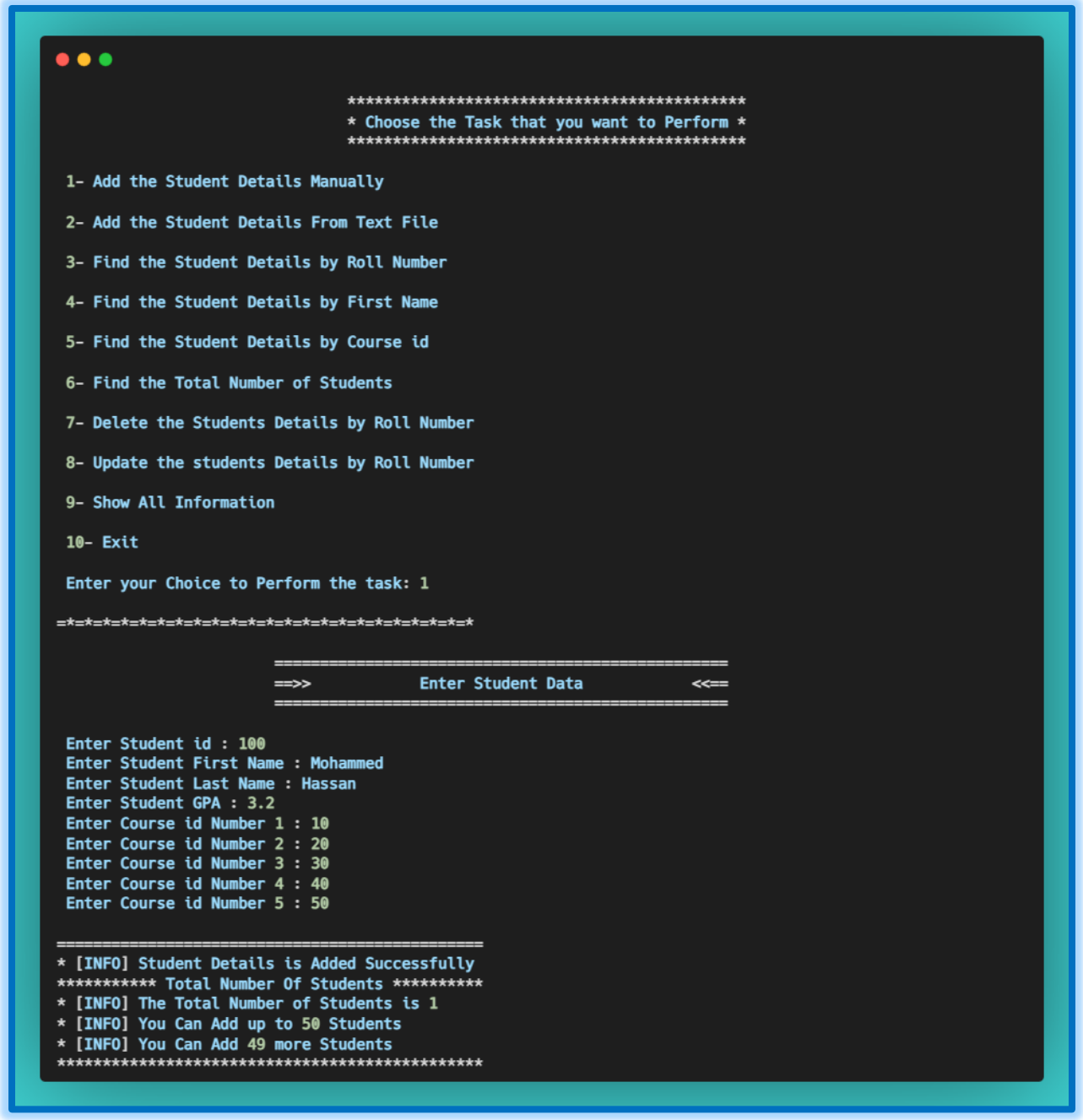
```c
// ==============>> Print Student Count Info <<==============
void Print_Student_Count(FIOF_Buf_t *Student_Queue)
{
    print("\n********** Total Number Of Students **********\n")
    print("* [INFO] The Total Number of Students is %d\n", Student_Queue->count);
    print("* [INFO] You Can Add up to %d Students\n", Max_Number_Students);
    print("* [INFO] You Can Add %d more Students\n", Max_Number_Students - Student_Queue->count);
    print("*********************************************\n");
}
```

11- Print Student information

```c
// ==============>> print Student info<<==============
void print_Student_info(SStudent_Node *student)
{
    int i ;
    print("*************************************************\n");
    print("Student ID: %d \n", student->roll);
    print("Student First Name: %s \n", student->first_name);
    print("Student Last Name: %s \n", student->last_name);
    print("Student GPA Number: %.2f \n", student->GPA);

    for (i = 0; i < Number_Courses; i++)
    {
        print("Course id Number %d are: %d \n", i + 1, student->Course_id[i]);
    }
}
```

❖ Program Execution Snapshots

➢ Add Student Manually

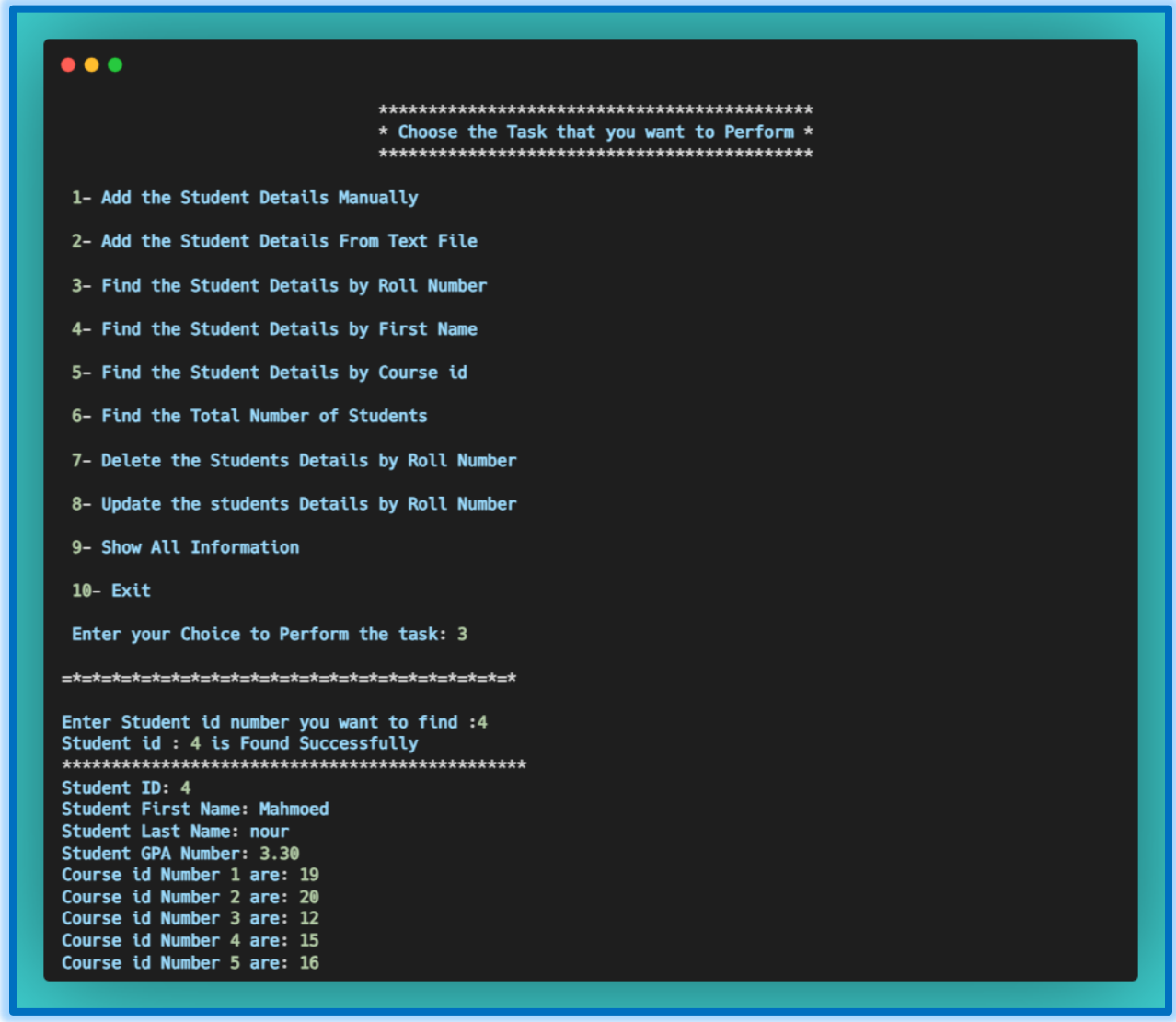


➢ Add Student from File

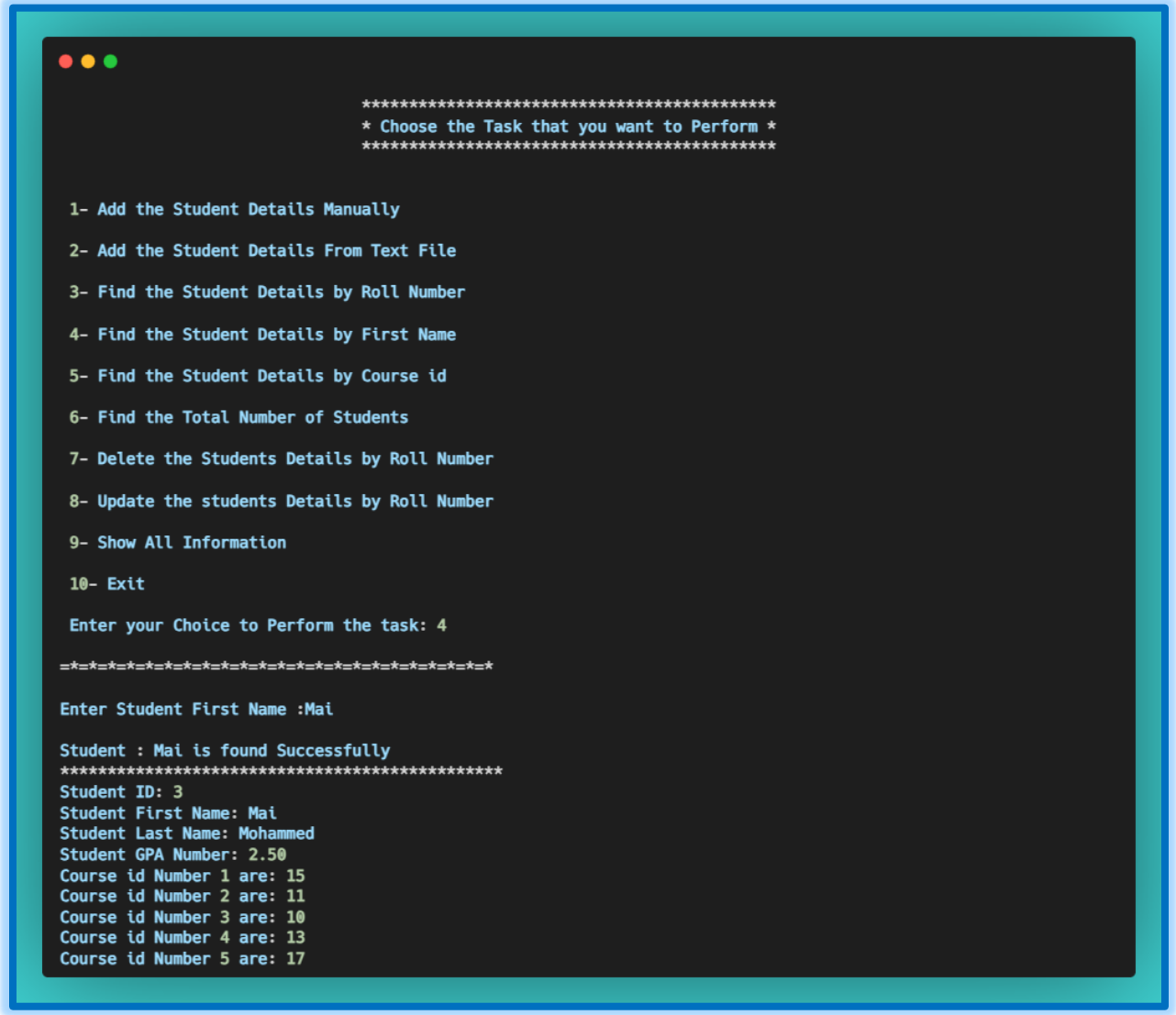
```
1   Mohammed Hassan    3.2    10 15 16 19 20
2   Khaled   Alaa      2.7    12 16 20 14 13
3   Mohammed Goda      3.7    15 11 10 13 17
1   Yahya    Amr       2.8    16 20 14 13 12
4   Mahmoed  nour      3.9    19 20 12 15 16
5   Salma    Ahmed     3.5    12 15 16 18 19
5   Mostafa  Ali       3.4    11 12 10 20 15
```

```
                                  ********************************************
                                  * 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 Students Details by Roll Number

 8- Update the students Details by Roll Number

 9- Show All Information

 10- Exit

 Enter your Choice to Perform the task: 2

=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

* [INFO] Roll Number 1 is Saved Successfully
* [INFO] Roll Number 2 is Saved Successfully
* [INFO] Roll Number 3 is Saved Successfully
===============================================
* [ERROR] Roll Number 1 is Already Taken Before
===============================================

* [INFO] Roll Number 4 is Saved Successfully
* [INFO] Roll Number 5 is Saved Successfully
===============================================
* [ERROR] Roll Number 5 is Already Taken Before
===============================================

* [Info] Student Details Are Saved Successfully
*********** Total Number Of Students **********
* [INFO] The Total Number of Students is 5
* [INFO] You Can Add up to 50 Students
* [INFO] You Can Add 45 more Students
***********************************************
```

➢ Find Student Details by ID Number



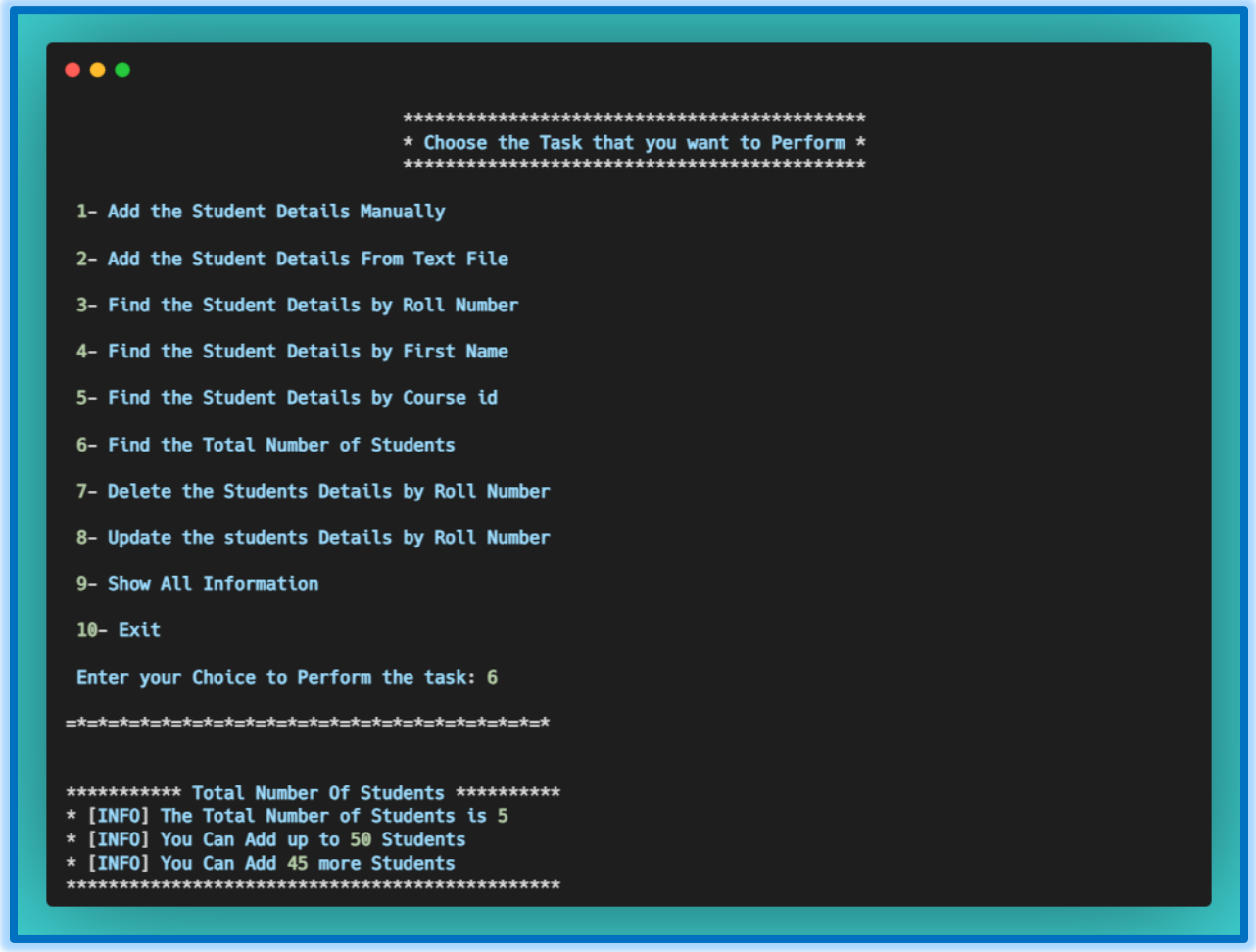
➢ Find Student details by First Name



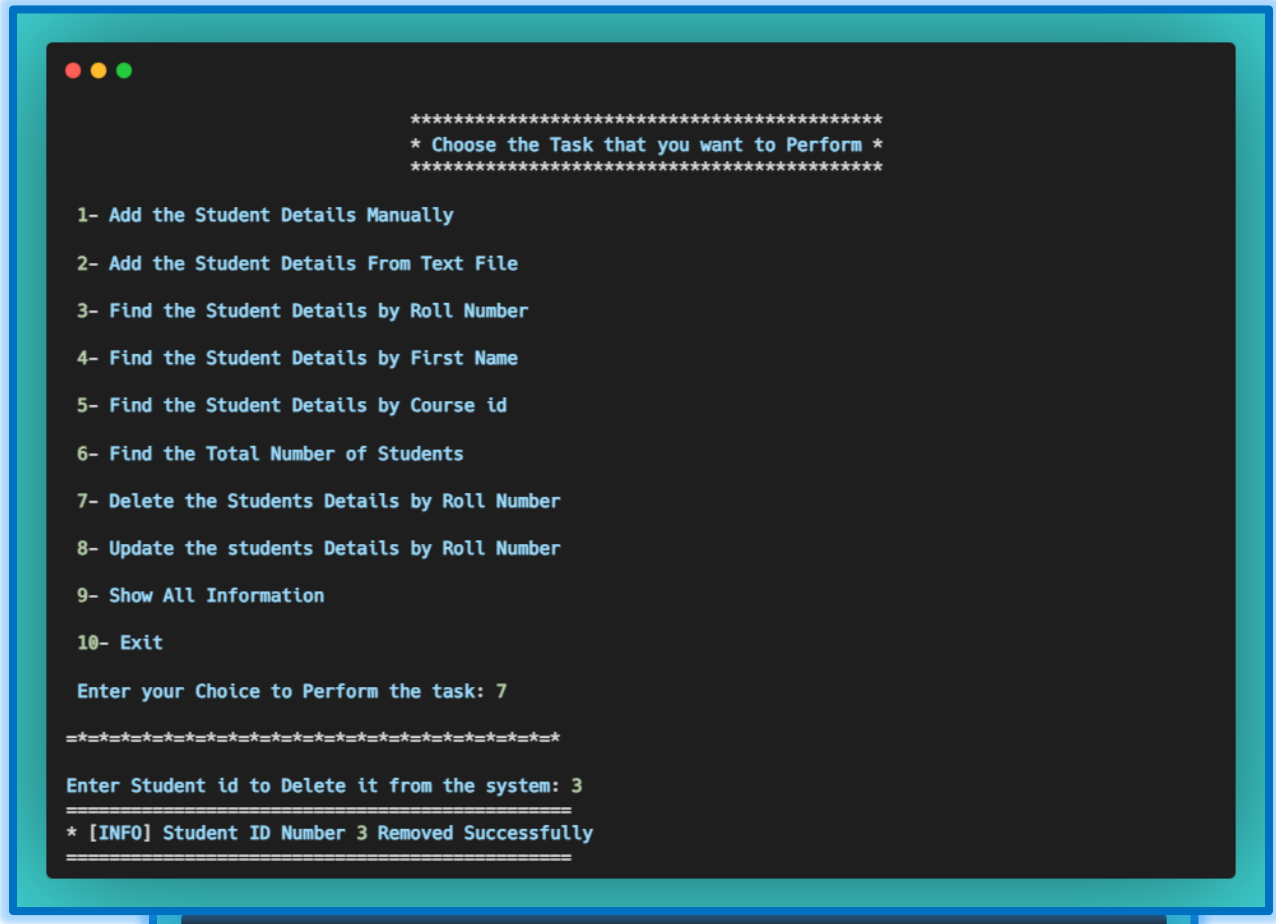
➢ Find Student Details by Course ID

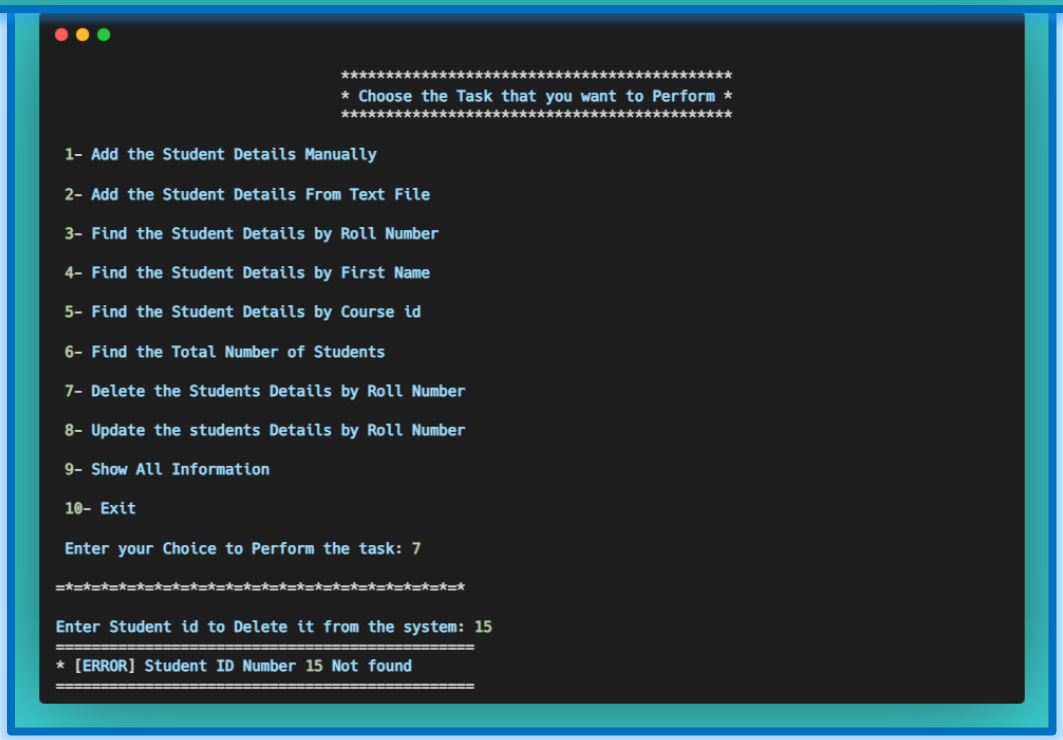
```
                                  *********************************************
                                  * Choose the Task that you want to Perform *
                                  *********************************************

 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 Students Details by Roll Number

 8- Update the students Details by Roll Number

 9- Show All Information

 10- Exit

 Enter your Choice to Perform the task: 5

=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

Enter course id: 12
***********************************************
Student ID: 2
Student First Name: Khaled
Student Last Name: Alaa
Student GPA Number: 2.70
Course id Number 1 are: 12
Course id Number 2 are: 16
Course id Number 3 are: 20
Course id Number 4 are: 14
Course id Number 5 are: 13
***********************************************
Student ID: 4
Student First Name: Mahmoed
Student Last Name: nour
Student GPA Number: 3.30
Course id Number 1 are: 19
Course id Number 2 are: 20
Course id Number 3 are: 12
Course id Number 4 are: 15
Course id Number 5 are: 16
***********************************************
Student ID: 5
Student First Name: Salma
Student Last Name: Ahmed
Student GPA Number: 3.50
Course id Number 1 are: 12
Course id Number 2 are: 15
Course id Number 3 are: 16
Course id Number 4 are: 18
Course id Number 5 are: 19
============================================================
[INFO] Number of Students enrolled in this course is : 3
============================================================
```

➢ Print Total Number of Students



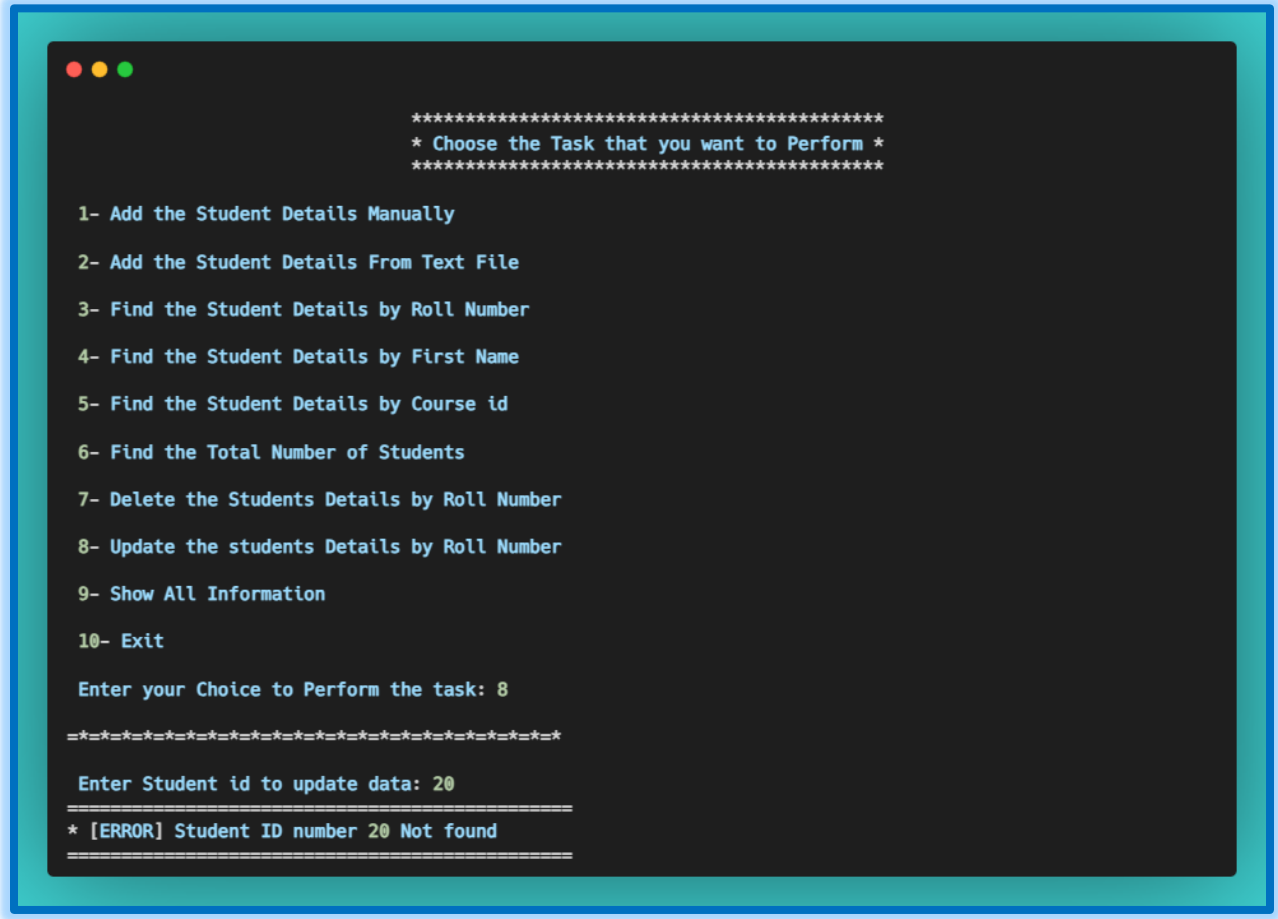
➢ Delete Student by Roll Number





➢ Update Student Details by Roll Number

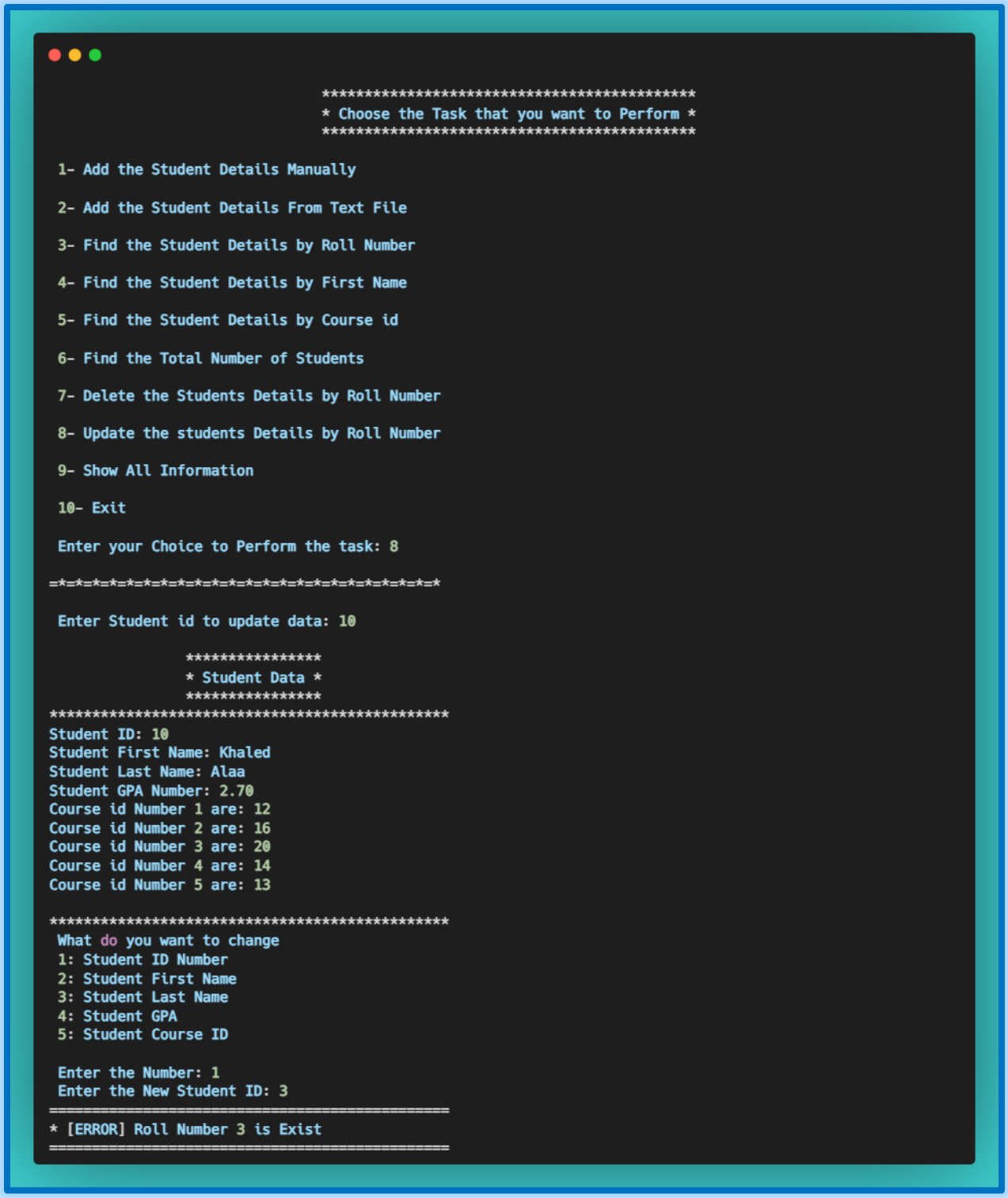
1- If Student ID Number Not found:



2- Update Student id:

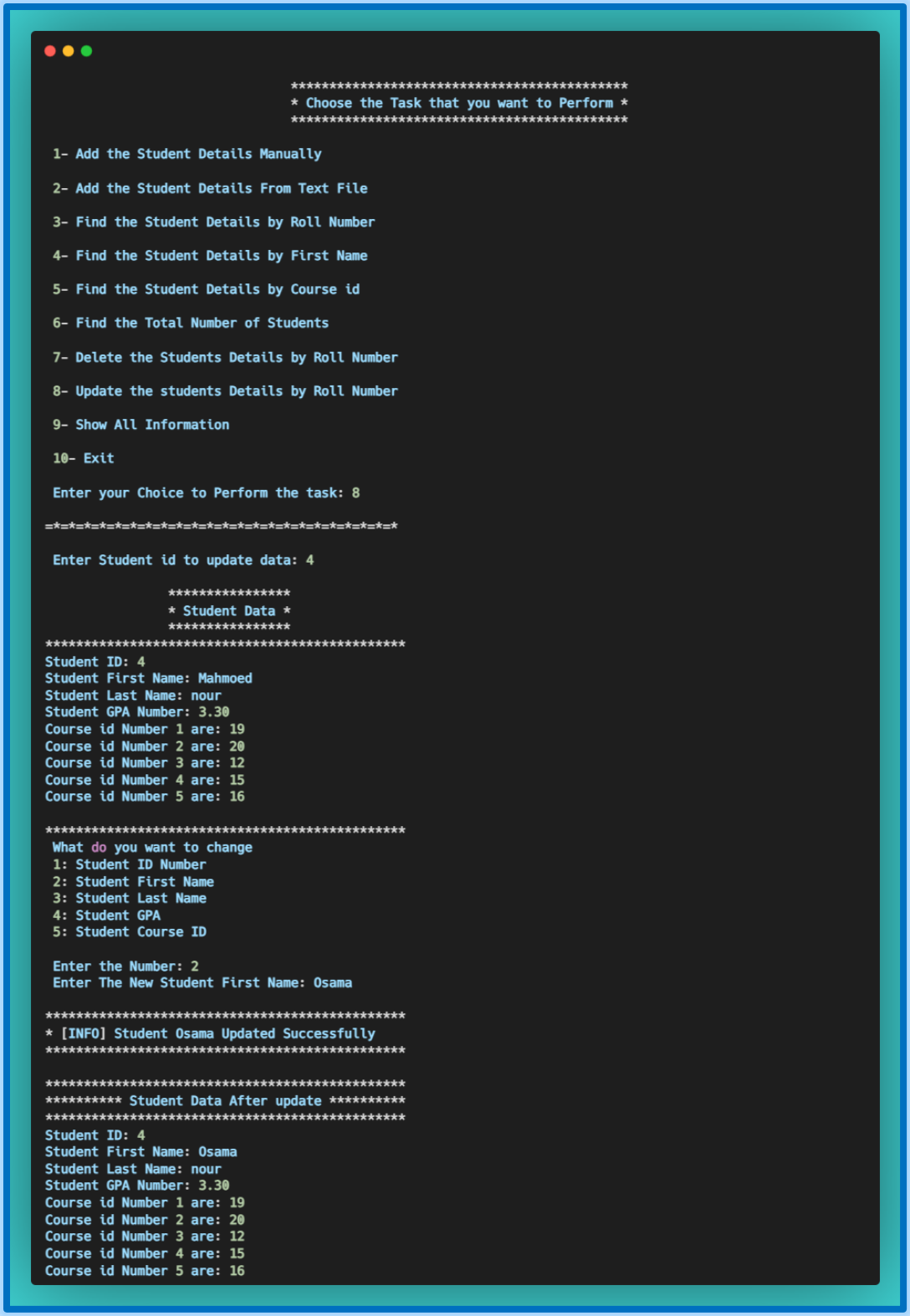
```
                                 ********************************************
                                 * Choose the Task that you want to Perform *
                                 ********************************************

 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 Students Details by Roll Number

 8- Update the students Details by Roll Number

 9- Show All Information

 10- Exit

 Enter your Choice to Perform the task: 8

=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

 Enter Student id to update data: 2

                ****************
                * Student Data *
                ****************
***********************************************
Student ID: 2
Student First Name: Khaled
Student Last Name: Alaa
Student GPA Number: 2.70
Course id Number 1 are: 12
Course id Number 2 are: 16
Course id Number 3 are: 20
Course id Number 4 are: 14
Course id Number 5 are: 13

***********************************************
 What do you want to change
 1: Student ID Number
 2: Student First Name
 3: Student Last Name
 4: Student GPA
 5: Student Course ID

 Enter the Number: 1
 Enter the New Student ID: 10

***************************************************
* [INFO] Student ID Number 10 Updated Successfully
***************************************************

***********************************************
********** Student Data After updated *********
***********************************************
Student ID: 10
Student First Name: Khaled
Student Last Name: Alaa
Student GPA Number: 2.70
Course id Number 1 are: 12
Course id Number 2 are: 16
Course id Number 3 are: 20
Course id Number 4 are: 14
Course id Number 5 are: 13
```

3- update Student id failed



```
                              ******************************************
                              * 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 Students Details by Roll Number

 8- Update the students Details by Roll Number

 9- Show All Information

 10- Exit

 Enter your Choice to Perform the task: 8

=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

 Enter Student id to update data: 10

               ***************
               * Student Data *
               ***************
*********************************************
Student ID: 10
Student First Name: Khaled
Student Last Name: Alaa
Student GPA Number: 2.70
Course id Number 1 are: 12
Course id Number 2 are: 16
Course id Number 3 are: 20
Course id Number 4 are: 14
Course id Number 5 are: 13

*********************************************
 What do you want to change
 1: Student ID Number
 2: Student First Name
 3: Student Last Name
 4: Student GPA
 5: Student Course ID

 Enter the Number: 1
 Enter the New Student ID: 3
=============================================
* [ERROR] Roll Number 3 is Exist
=============================================
```

4- update Student First Name



```
                              ******************************************
                              * 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 Students Details by Roll Number

 8- Update the students Details by Roll Number

 9- Show All Information

 10- Exit

 Enter your Choice to Perform the task: 8

=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

 Enter Student id to update data: 4

                ***************
                * Student Data *
                ***************
**********************************************
Student ID: 4
Student First Name: Mahmoed
Student Last Name: nour
Student GPA Number: 3.30
Course id Number 1 are: 19
Course id Number 2 are: 20
Course id Number 3 are: 12
Course id Number 4 are: 15
Course id Number 5 are: 16

**********************************************
 What do you want to change
 1: Student ID Number
 2: Student First Name
 3: Student Last Name
 4: Student GPA
 5: Student Course ID

 Enter the Number: 2
 Enter The New Student First Name: Osama

**********************************************
* [INFO] Student Osama Updated Successfully
**********************************************

**********************************************
********** Student Data After update **********
**********************************************
Student ID: 4
Student First Name: Osama
Student Last Name: nour
Student GPA Number: 3.30
Course id Number 1 are: 19
Course id Number 2 are: 20
Course id Number 3 are: 12
Course id Number 4 are: 15
Course id Number 5 are: 16
```

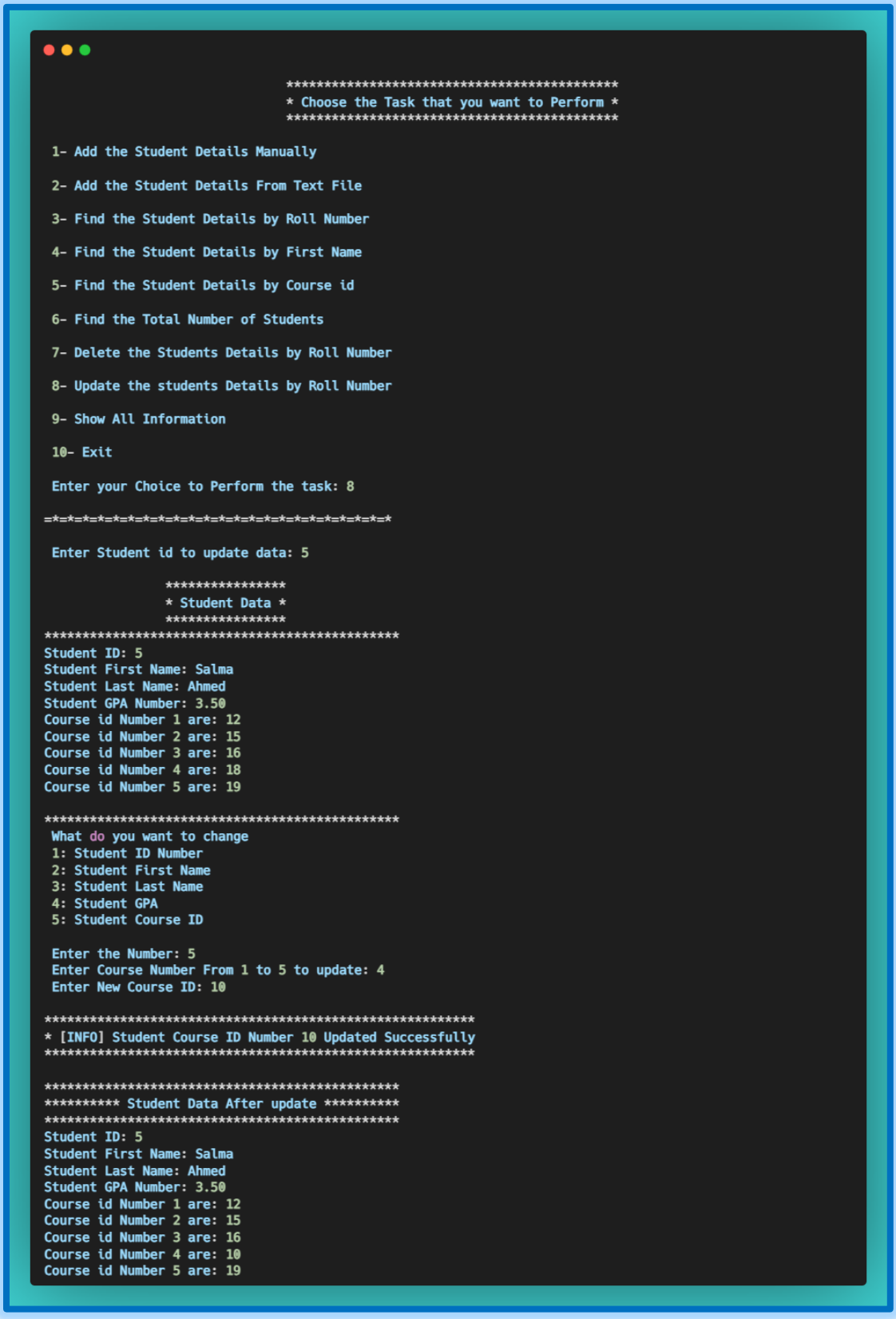
5- update Student Last Name

```
                          ****************************************
                          * 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 Students Details by Roll Number

 8- Update the students Details by Roll Number

 9- Show All Information

 10- Exit

 Enter your Choice to Perform the task: 8

=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

 Enter Student id to update data: 4

               ***************
               * Student Data *
               ***************
**********************************************
Student ID: 4
Student First Name: Osama
Student Last Name: nour
Student GPA Number: 3.30
Course id Number 1 are: 19
Course id Number 2 are: 20
Course id Number 3 are: 12
Course id Number 4 are: 15
Course id Number 5 are: 16

**********************************************
 What do you want to change
 1: Student ID Number
 2: Student First Name
 3: Student Last Name
 4: Student GPA
 5: Student Course ID

 Enter the Number: 3
 Enter The New Student Last Name: Ali

**************************************************
* [INFO] Student Last Name Ali Updated Successfully
**************************************************

**********************************************
********** Student Data After update **********
**********************************************
Student ID: 4
Student First Name: Osama
Student Last Name: Ali
Student GPA Number: 3.30
Course id Number 1 are: 19
Course id Number 2 are: 20
Course id Number 3 are: 12
Course id Number 4 are: 15
Course id Number 5 are: 16
```

6- Update Student GPA

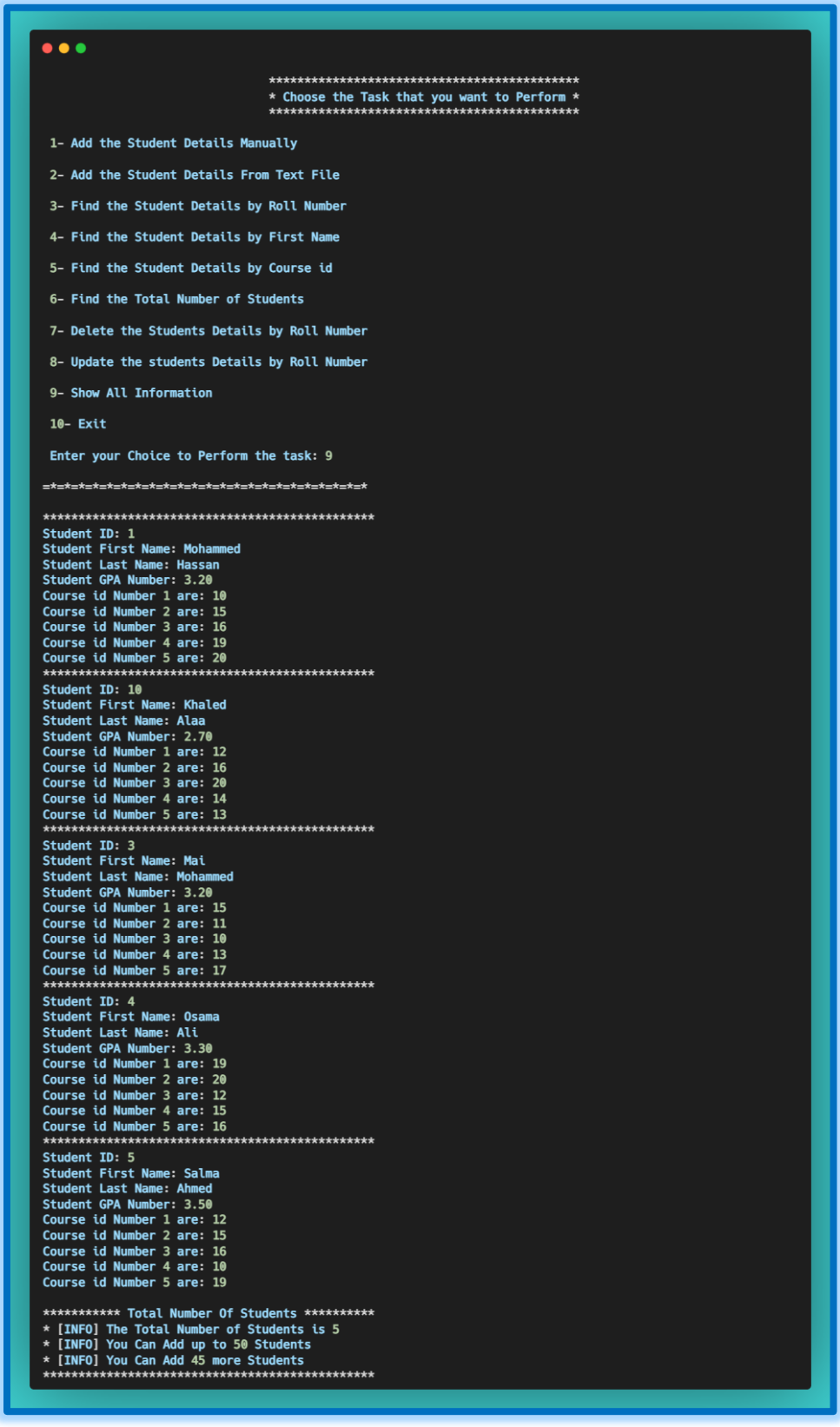
```
                              ****************************************
                              * Choose the Task that you want to Perform *
                              ****************************************

 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 Students Details by Roll Number

 8- Update the students Details by Roll Number

 9- Show All Information

 10- Exit

 Enter your Choice to Perform the task: 8

=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

 Enter Student id to update data: 3

               ***************
               * Student Data *
               ***************
*********************************************
Student ID: 3
Student First Name: Mai
Student Last Name: Mohammed
Student GPA Number: 2.50
Course id Number 1 are: 15
Course id Number 2 are: 11
Course id Number 3 are: 10
Course id Number 4 are: 13
Course id Number 5 are: 17

*********************************************
 What do you want to change
 1: Student ID Number
 2: Student First Name
 3: Student Last Name
 4: Student GPA
 5: Student Course ID

 Enter the Number: 4
 Enter The New Student GPA: 3.2

*********************************************
* [INFO] Student GPA 3.20 Updated Successfully
*********************************************

*********************************************
********** Student Data After update **********
*********************************************
Student ID: 3
Student First Name: Mai
Student Last Name: Mohammed
Student GPA Number: 3.20
Course id Number 1 are: 15
Course id Number 2 are: 11
Course id Number 3 are: 10
Course id Number 4 are: 13
Course id Number 5 are: 17
```

7- update Student Course ID Number



```
                              ****************************************
                              * 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 Students Details by Roll Number

 8- Update the students Details by Roll Number

 9- Show All Information

 10- Exit

 Enter your Choice to Perform the task: 8

=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

 Enter Student id to update data: 5

               ***************
               * Student Data *
               ***************
*********************************************
Student ID: 5
Student First Name: Salma
Student Last Name: Ahmed
Student GPA Number: 3.50
Course id Number 1 are: 12
Course id Number 2 are: 15
Course id Number 3 are: 16
Course id Number 4 are: 18
Course id Number 5 are: 19

*********************************************
 What do you want to change
 1: Student ID Number
 2: Student First Name
 3: Student Last Name
 4: Student GPA
 5: Student Course ID

 Enter the Number: 5
 Enter Course Number From 1 to 5 to update: 4
 Enter New Course ID: 10

*********************************************************
* [INFO] Student Course ID Number 10 Updated Successfully
*********************************************************

*********************************************
********** Student Data After update **********
*********************************************
Student ID: 5
Student First Name: Salma
Student Last Name: Ahmed
Student GPA Number: 3.50
Course id Number 1 are: 12
Course id Number 2 are: 15
Course id Number 3 are: 16
Course id Number 4 are: 10
Course id Number 5 are: 19
```

8-Show All Details information of 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 Students Details by Roll Number

 8- Update the students Details by Roll Number

 9- Show All Information

 10- Exit

 Enter your Choice to Perform the task: 9

=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

**********************************************
Student ID: 1
Student First Name: Mohammed
Student Last Name: Hassan
Student GPA Number: 3.20
Course id Number 1 are: 10
Course id Number 2 are: 15
Course id Number 3 are: 16
Course id Number 4 are: 19
Course id Number 5 are: 20
**********************************************
Student ID: 10
Student First Name: Khaled
Student Last Name: Alaa
Student GPA Number: 2.70
Course id Number 1 are: 12
Course id Number 2 are: 16
Course id Number 3 are: 20
Course id Number 4 are: 14
Course id Number 5 are: 13
**********************************************
Student ID: 3
Student First Name: Mai
Student Last Name: Mohammed
Student GPA Number: 3.20
Course id Number 1 are: 15
Course id Number 2 are: 11
Course id Number 3 are: 10
Course id Number 4 are: 13
Course id Number 5 are: 17
**********************************************
Student ID: 4
Student First Name: Osama
Student Last Name: Ali
Student GPA Number: 3.30
Course id Number 1 are: 19
Course id Number 2 are: 20
Course id Number 3 are: 12
Course id Number 4 are: 15
Course id Number 5 are: 16
**********************************************
Student ID: 5
Student First Name: Salma
Student Last Name: Ahmed
Student GPA Number: 3.50
Course id Number 1 are: 12
Course id Number 2 are: 15
Course id Number 3 are: 16
Course id Number 4 are: 10
Course id Number 5 are: 19

********** Total Number Of Students **********
* [INFO] The Total Number of Students is 5
* [INFO] You Can Add up to 50 Students
* [INFO] You Can Add 45 more Students
**********************************************
```

9- Exit the Program

```
**********************************************
* 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 Students Details by Roll Number

8- Update the students Details by Roll Number

9- Show All Information

10- Exit

Enter your Choice to Perform the task: 10

=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

                  ****************** Thank you ******************
```