MASTERING EMBEDDED SYSTEM ONLINE DIPLOMA

First Term

PROJECT 2

ENG. OMAR SHAWKY MOHAMED

https://www.learn-in-depth.com/online-diploma/omarshawky9@gmail.com

STUDENT DATABASE

1 PROBLEM STATEMENT

This project is mainly about implementing a software system to manage the students' information regarding the following: (First Name, Last Name, GPA , Unique Roll Number, Current Enrolled Courses).

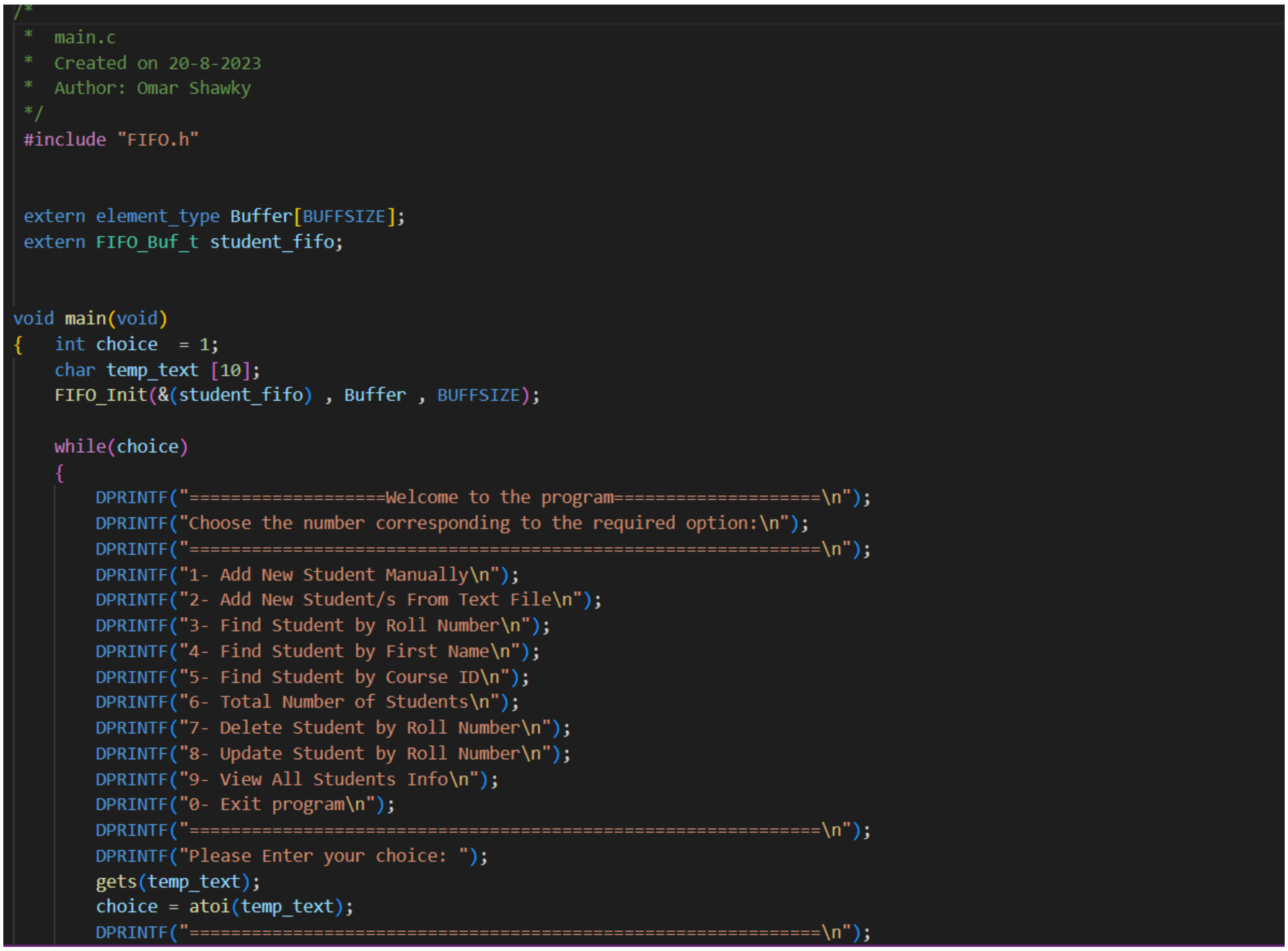
2 Functions to implement

The idea is to form an individual function for each operation. All the functions are unified to form a software system. The functions needed to be implemented are expected to be as following:

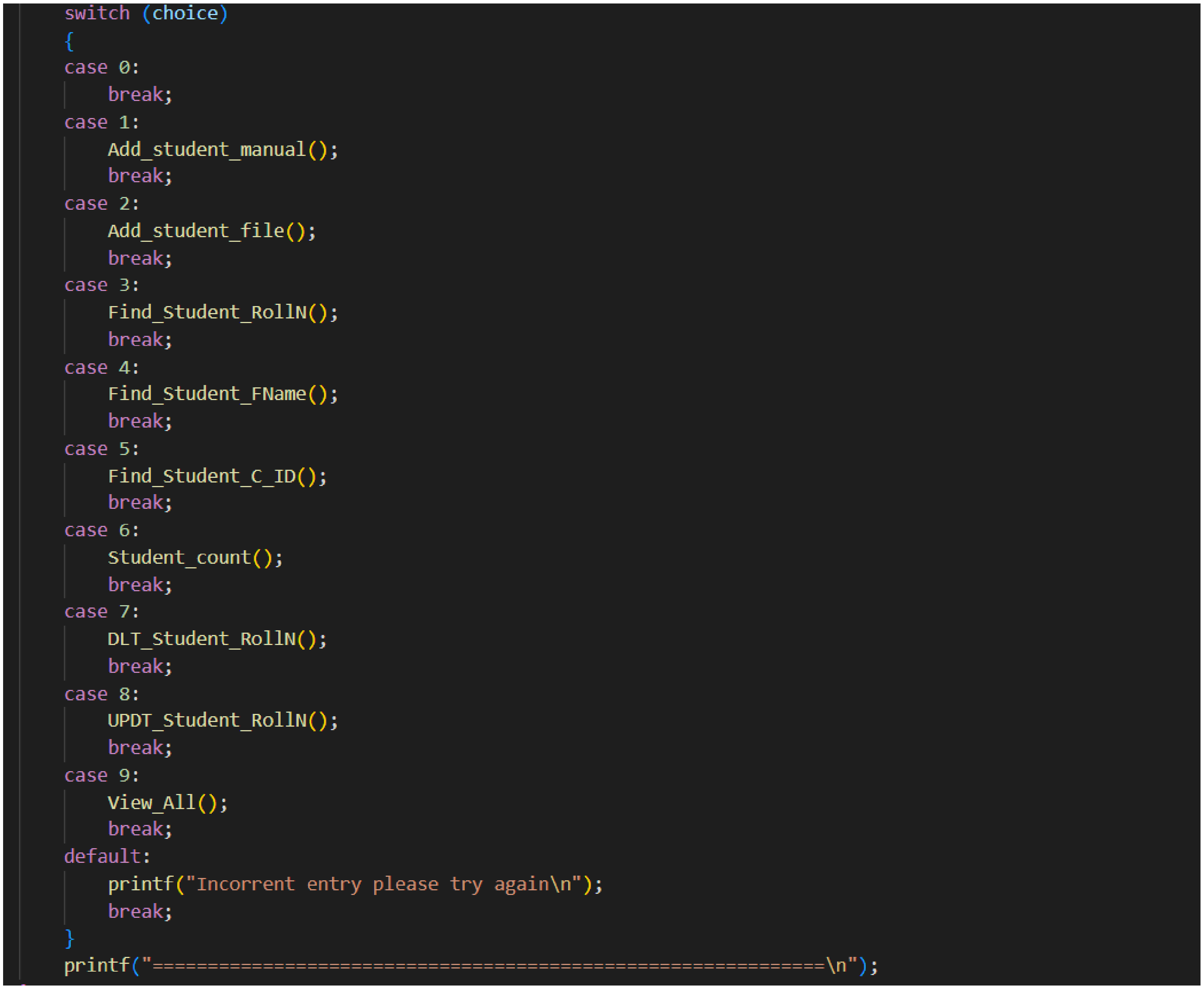
1. Add Student Details from File.
2. Add Student Details manually.
3. Find Student by given Roll Number.
4. Find Student by given First Name.
5. Find Students enrolled in a course.
6. Count of students.
7. Delete a student.
8. Update a student.
9. View all info.

3 IMPLEMENTATION

3.1 main.c

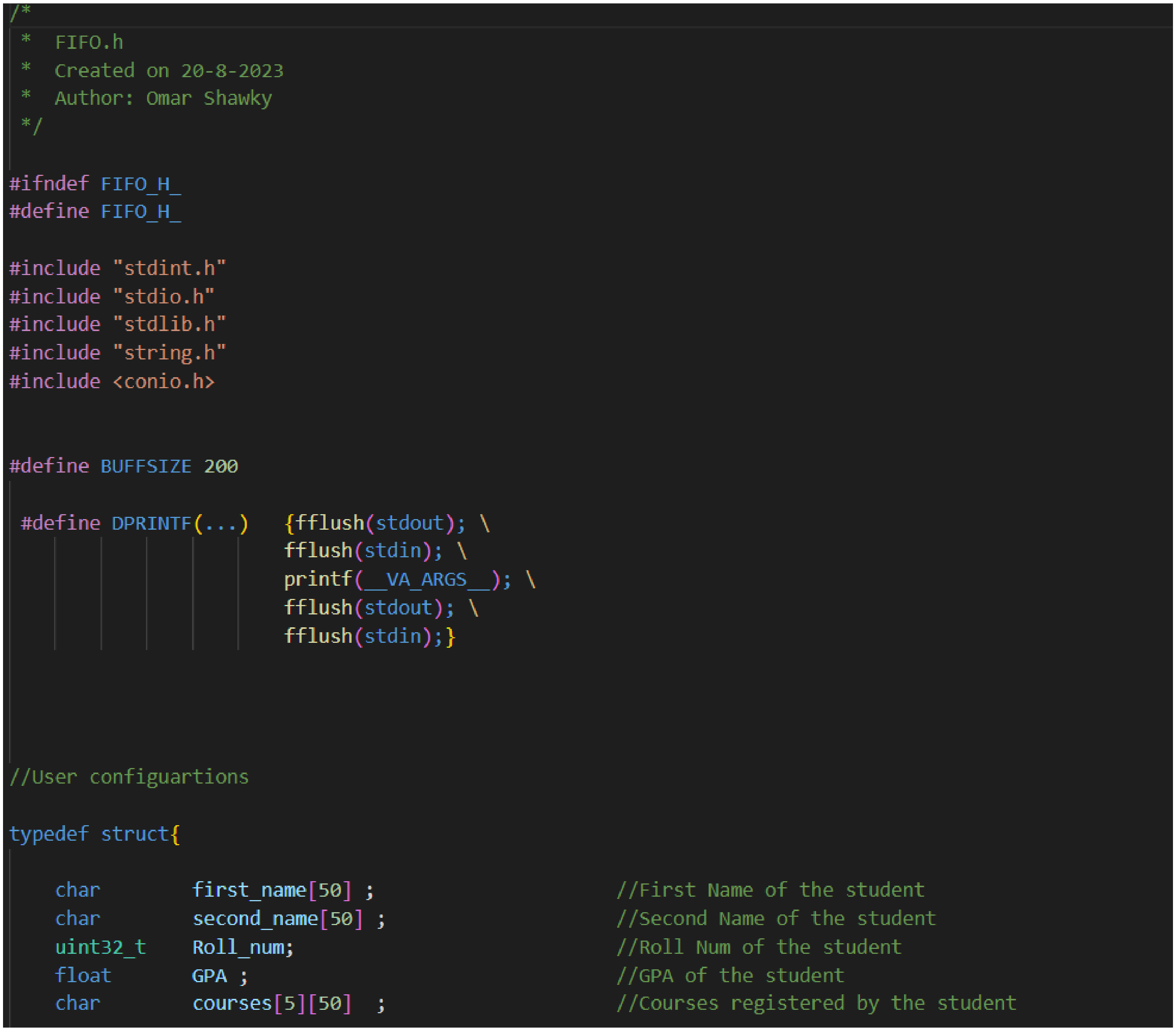


```c
/*
 *  main.c
 *  Created on 20-8-2023
 *  Author: Omar Shawky
 */
#include "FIFO.h"


extern element_type Buffer[BUFFSIZE];
extern FIFO_Buf_t student_fifo;


void main(void)
{   int choice  = 1;
    char temp_text [10];
    FIFO_Init(&(student_fifo) , Buffer , BUFFSIZE);

    while(choice)
    {
        DPRINTF("==================Welcome to the program===================\n");
        DPRINTF("Choose the number corresponding to the required option:\n");
        DPRINTF("=============================================================\n");
        DPRINTF("1- Add New Student Manually\n");
        DPRINTF("2- Add New Student/s From Text File\n");
        DPRINTF("3- Find Student by Roll Number\n");
        DPRINTF("4- Find Student by First Name\n");
        DPRINTF("5- Find Student by Course ID\n");
        DPRINTF("6- Total Number of Students\n");
        DPRINTF("7- Delete Student by Roll Number\n");
        DPRINTF("8- Update Student by Roll Number\n");
        DPRINTF("9- View All Students Info\n");
        DPRINTF("0- Exit program\n");
        DPRINTF("=============================================================\n");
        DPRINTF("Please Enter your choice: ");
        gets(temp_text);
        choice = atoi(temp_text);
        DPRINTF("=============================================================\n");
```

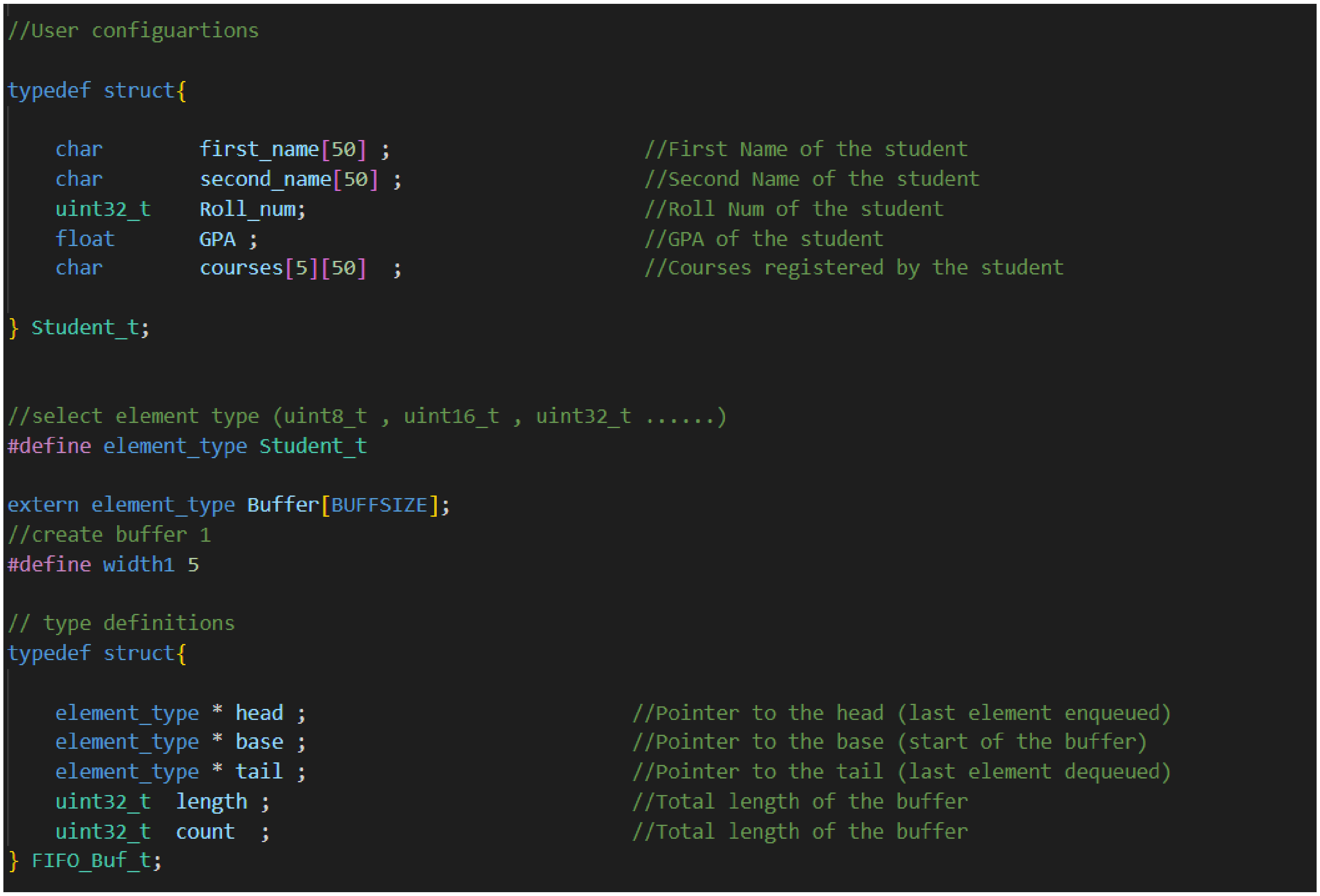


```c
    switch (choice)
    {
    case 0:
        break;
    case 1:
        Add_student_manual();
        break;
    case 2:
        Add_student_file();
        break;
    case 3:
        Find_Student_RollN();
        break;
    case 4:
        Find_Student_FName();
        break;
    case 5:
        Find_Student_C_ID();
        break;
    case 6:
        Student_count();
        break;
    case 7:
        DLT_Student_RollN();
        break;
    case 8:
        UPDT_Student_RollN();
        break;
    case 9:
        View_All();
        break;
    default:
        printf("Incorrent entry please try again\n");
        break;
    }
    printf("=============================================================\n");
```

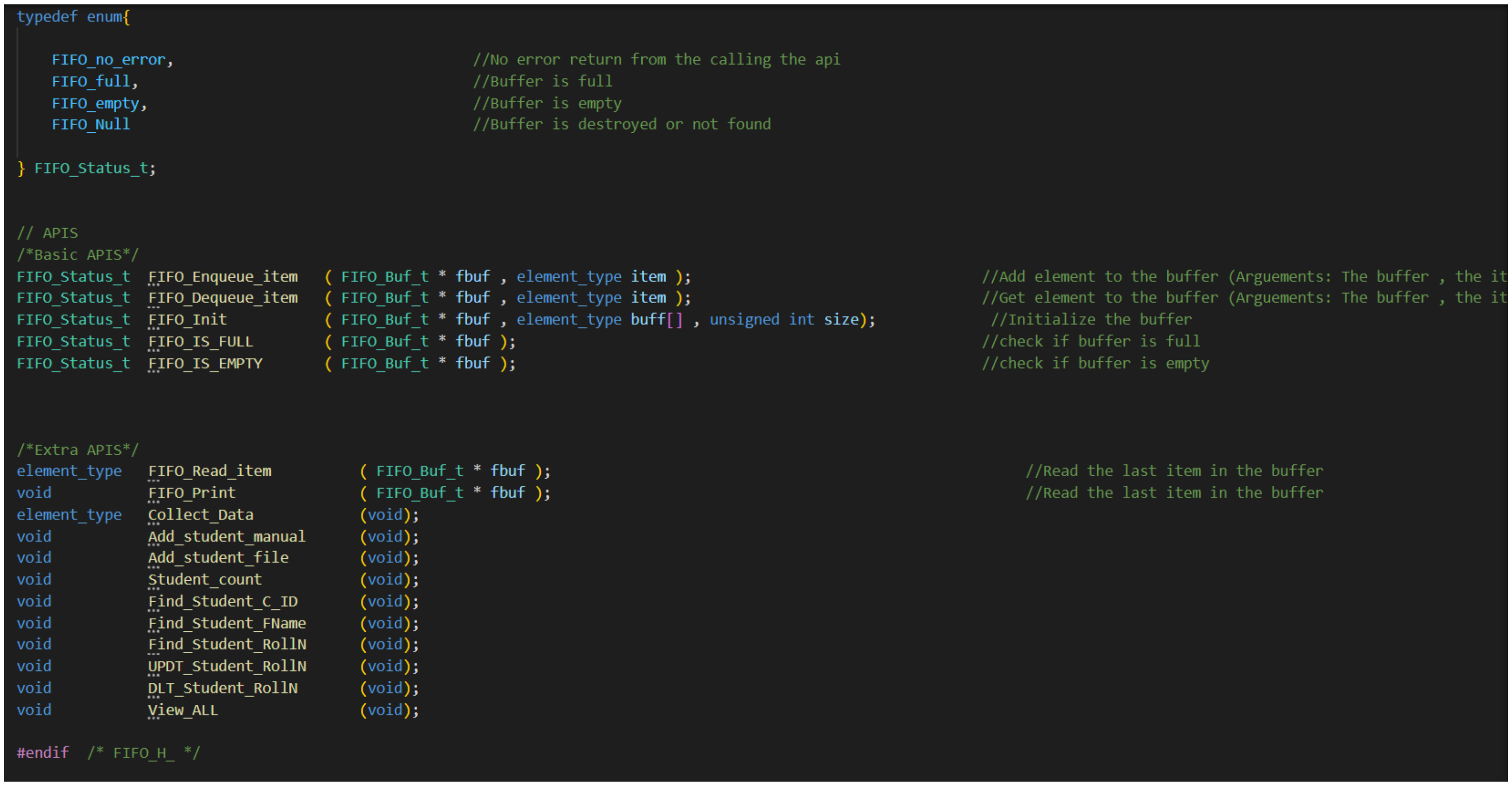
3.2 FIFO.h



```c
/*
 *   FIFO.h
 *   Created on 20-8-2023
 *   Author: Omar Shawky
 */

#ifndef FIFO_H_
#define FIFO_H_

#include "stdint.h"
#include "stdio.h"
#include "stdlib.h"
#include "string.h"
#include <conio.h>


#define BUFFSIZE 200

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




//User configuartions

typedef struct{

    char        first_name[50] ;                    //First Name of the student
    char        second_name[50] ;                   //Second Name of the student
    uint32_t    Roll_num;                           //Roll Num of the student
    float       GPA ;                               //GPA of the student
    char        courses[5][50]   ;                  //Courses registered by the student
```



```c
//User configuartions

typedef struct{

    char        first_name[50] ;                    //First Name of the student
    char        second_name[50] ;                   //Second Name of the student
    uint32_t    Roll_num;                           //Roll Num of the student
    float       GPA ;                               //GPA of the student
    char        courses[5][50]  ;                   //Courses registered by the student

} Student_t;


//select element type (uint8_t , uint16_t , uint32_t ......)
#define element_type Student_t

extern element_type Buffer[BUFFSIZE];
//create buffer 1
#define width1 5

// type definitions
typedef struct{

    element_type * head ;                          //Pointer to the head (last element enqueued)
    element_type * base ;                          //Pointer to the base (start of the buffer)
    element_type * tail ;                          //Pointer to the tail (last element dequeued)
    uint32_t  length ;                             //Total length of the buffer
    uint32_t  count  ;                             //Total length of the buffer
} FIFO_Buf_t;
```

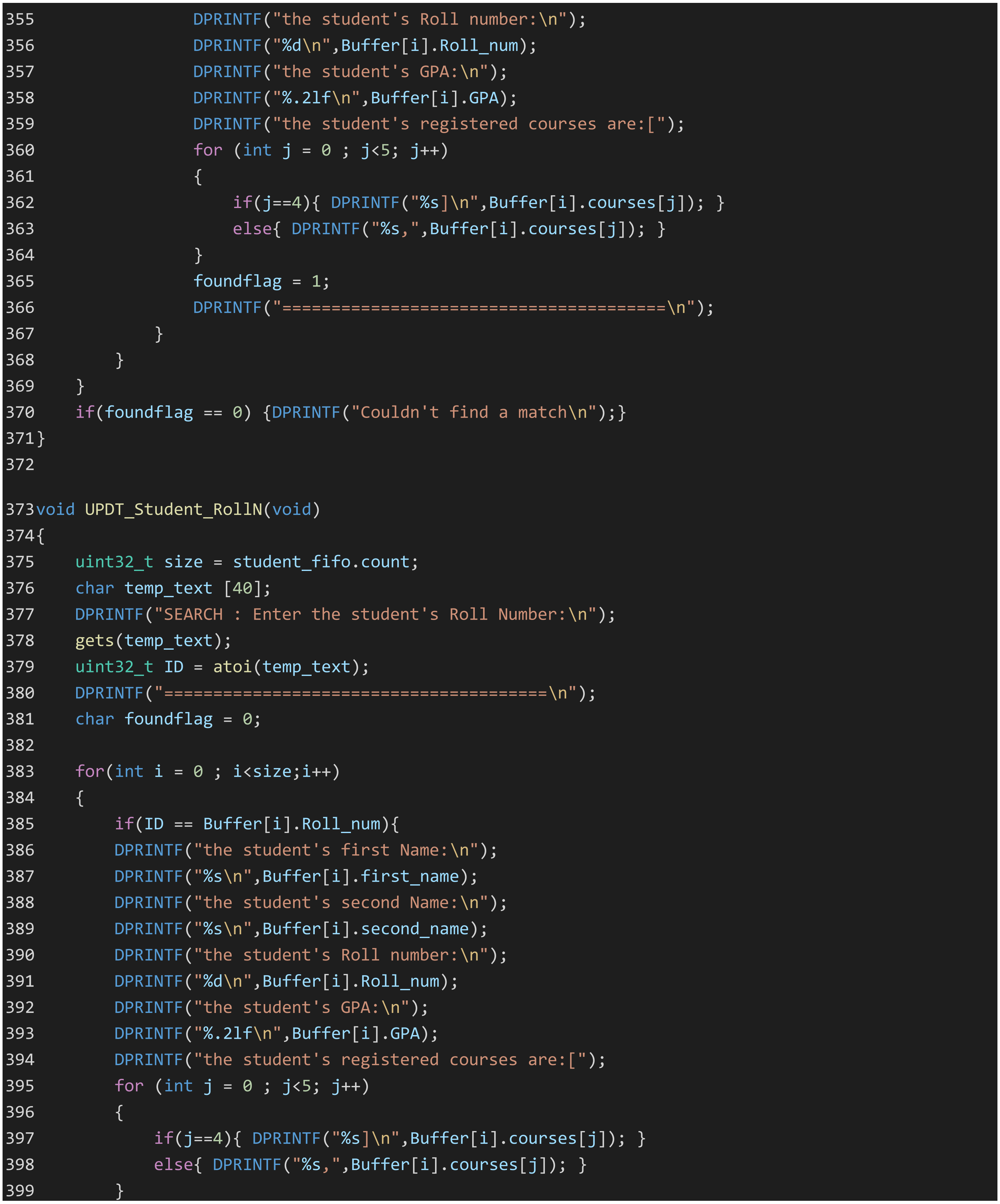


```c
typedef enum{

    FIFO_no_error,                              //No error return from the calling the api
    FIFO_full,                                  //Buffer is full
    FIFO_empty,                                 //Buffer is empty
    FIFO_Null                                   //Buffer is destroyed or not found

} FIFO_Status_t;


// APIS
/*Basic APIS*/
FIFO_Status_t  FIFO_Enqueue_item   ( FIFO_Buf_t * fbuf , element_type item );                        //Add element to the buffer (Arguements: The buffer , the it
FIFO_Status_t  FIFO_Dequeue_item   ( FIFO_Buf_t * fbuf , element_type item );                        //Get element to the buffer (Arguements: The buffer , the it
FIFO_Status_t  FIFO_Init           ( FIFO_Buf_t * fbuf , element_type buff[] , unsigned int size);   //Initialize the buffer
FIFO_Status_t  FIFO_IS_FULL        ( FIFO_Buf_t * fbuf );                                            //check if buffer is full
FIFO_Status_t  FIFO_IS_EMPTY       ( FIFO_Buf_t * fbuf );                                            //check if buffer is empty



/*Extra APIS*/
element_type   FIFO_Read_item          ( FIFO_Buf_t * fbuf );                                            //Read the last item in the buffer
void           FIFO_Print              ( FIFO_Buf_t * fbuf );                                            //Read the last item in the buffer
element_type   Collect_Data            (void);
void           Add_student_manual      (void);
void           Add_student_file        (void);
void           Student_count           (void);
void           Find_Student_C_ID       (void);
void           Find_Student_FName      (void);
void           Find_Student_RollN      (void);
void           UPDT_Student_RollN      (void);
void           DLT_Student_RollN       (void);
void           View_ALL                (void);

#endif  /* FIFO_H_ */
```

3.3 FIFO.c

```c
/*
 *  LIFO.c
 *  Created on 20-8-2023
 *  Author: Omar Shawky
 */

 #include "FIFO.h"

 #ifndef NULL
 #define NULL 0
 #endif

 element_type Buffer[BUFFSIZE] ;            //Static allocation of 5*(sizeof(unsigned int)) = 20 byte
 FIFO_Buf_t student_fifo;

// APIS
/*Basic APIS*/
FIFO_Status_t  FIFO_enqueue_item   ( FIFO_Buf_t * fbuf , element_type  item )            //Add element to the buffer (Arguements: The buffer , the item to be added)
{
    //Check if the FIFO is valid
    if ( ! fbuf->head || ! fbuf->base || ! fbuf->tail  ){
        return FIFO_Null;
    }
    //Check if the LIFO is full
    if (  fbuf->count ==  fbuf->length ){
        return FIFO_full;
    }
    //queue the value (Cicular Queue)
    *(fbuf->head)   = item;
    fbuf->count ++;

    if(fbuf->head == (fbuf->base + fbuf->length))
    {
        fbuf->head = fbuf->base;
    }
    else{
    fbuf->head ++;
    }

    return FIFO_no_error ;
}

```

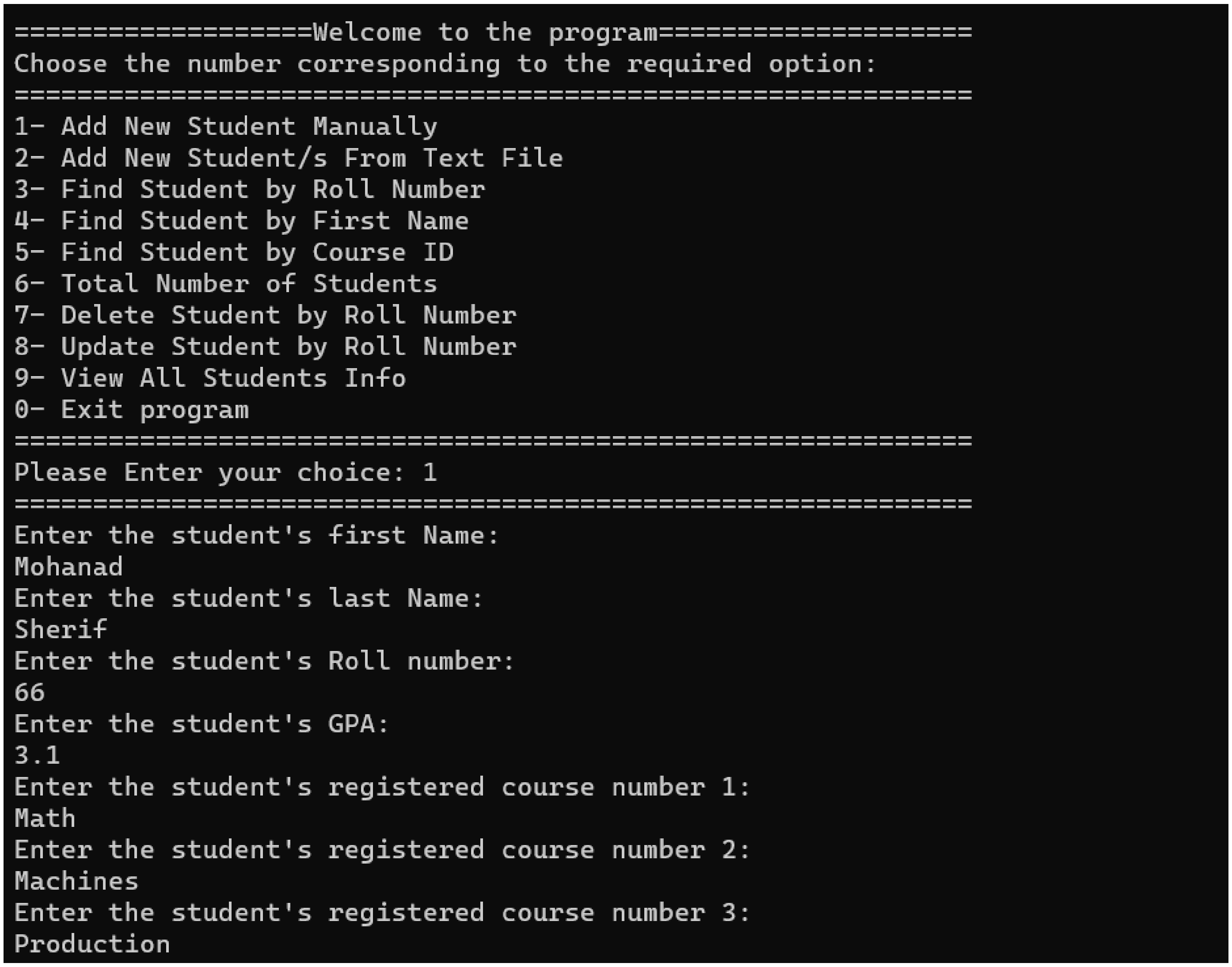
```c
FIFO_Status_t  FIFO_dequeue_item    ( FIFO_Buf_t * fbuf , element_type item )             //Get element to the buffer (Arguements: The buffer , the item to be gotten)
{
    //Check if the LIFO is valid
    if ( ! fbuf->head || ! fbuf->base || ! fbuf->tail  ){
        return FIFO_Null;
    }
    //Check if the LIFO is empty
    if (  fbuf->count ==  0 ){
        return FIFO_empty;
    }
    //dequeue the value (Cicular Queue)
    (item)  = *(fbuf->tail);
    fbuf->count --;
    if(fbuf->tail == (fbuf->base + fbuf->length))
    {
        fbuf->tail = fbuf->base;
    }
    else{
    fbuf->tail ++;
    }

    return FIFO_no_error ;
}


FIFO_Status_t  FIFO_Init        ( FIFO_Buf_t * fbuf , element_type buff[] , unsigned int size)              //Initialize the buffer
{
    if( buff == NULL ) {
        return FIFO_Null;
    }

    fbuf->base = buff;
    fbuf->head = buff;
    fbuf->tail = buff;
    fbuf->length = size;
    fbuf->count = 0;

    return FIFO_no_error ;
}

FIFO_Status_t  FIFO_IS_FULL        ( FIFO_Buf_t * fbuf )                                                  //check if buffer is full
{
    if ( ! fbuf->head || ! fbuf->base || ! fbuf->tail  ){
        return FIFO_Null;
    }
    //Check if the LIFO is full
    if (  fbuf->count ==  fbuf->length ){
        return FIFO_full;
    }
    else{
        return FIFO_no_error;
    }
}

FIFO_Status_t  FIFO_IS_EMPTY       ( FIFO_Buf_t * fbuf )                    //check if buffer is empty
{
    if ( ! fbuf->head || ! fbuf->base || ! fbuf->tail  ){
        return FIFO_Null;
    }
    //Check if the LIFO is empty
    if (  fbuf->count ==  0 ){
        return FIFO_empty;
    }
    else{
        return FIFO_no_error;
    }
}

/*Extra APIS*/
element_type   FIFO_Read_item   ( FIFO_Buf_t * fbuf )                              //Read the last item in the buffer
{
    //Check if the LIFO is valid
    if ( ! fbuf->head || ! fbuf->base ){
        //return FIFO_Null;
    }
    //Check if the LIFO is empty
    if (  fbuf->count ==  0 ){
        //return FIFO_empty;
    }

    return (element_type)*(fbuf->head) ;
}

void            FIFO_Print      ( FIFO_Buf_t * fbuf )                              //Read the last item in the buffer
{
    element_type * temp ;
    int i ;
    //Check if the FIFO is empty
    if (  fbuf->count ==  0 ){
        printf("Fifo is empty \n");
    }
    else {
        temp = fbuf->tail ;
        printf("=========fifo print=========\n");
        for(i=0;i<(fbuf->count);i++)
        {
            printf("%d \n",*temp);
        if( temp == (fbuf->base + fbuf->length))
            {
                temp = fbuf->base;
            }
        else{
                temp++;
            }
        }
        printf("==========fifo end==========\n");
    }
}

void Add_student_manual(void)
{
    Student_t s ;
    char temp_text [40];
    DPRINTF("Enter the student's first Name:\n");
    gets(temp_text);
    //DPRINTF("%s\n",temp_text);
    strcpy(s.first_name , temp_text);

    DPRINTF("Enter the student's last Name:\n");
    gets(temp_text);
    strcpy(s.second_name , temp_text);

    DPRINTF("Enter the student's Roll number:\n");
    gets(temp_text);
    s.Roll_num = atoi(temp_text);

    DPRINTF("Enter the student's GPA:\n");
    gets(temp_text);
    s.GPA = atof(temp_text);

    for(int i=0 ; i<5 ; i++)
    {
        DPRINTF("Enter the student's registered course number %d:\n",i+1);
        gets(temp_text);
        strcpy(s.courses[i] , temp_text);

    }
    DPRINTF("#######Data filled successfully########\n");
    FIFO_enqueue_item ( &student_fifo ,   s );
}

void Add_student_file()
{

    FILE *fp;
    fp = fopen("StudentFile.txt", "r");

    if(fp == NULL){
        DPRINTF("\n[ERROR] Failed to open file");
        return;
    }

    char line[100];
    char *token;

    while(fgets(line, sizeof(line), fp)){

        int counter = 1;
        int course = 0;
        Student_t new_s;
        token = strtok(line, " ");

        while(token != NULL){
            switch(counter){
            case 1:
                new_s.Roll_num = atoi(token);
                break;
            case 2:
                strcpy(new_s.first_name, token);
                break;
            case 3:
                strcpy(new_s.second_name, token);
                break;
            case 4:
                new_s.GPA = atof(token);
                break;
            case 5:
                strcpy(new_s.courses[course],token);
                course++;
                counter--;
                break;
            }
            token = strtok(NULL, " ");
            counter++;
        }
        *(student_fifo.head) = new_s;
        student_fifo.count++;

        if(student_fifo.head == (student_fifo.base + (student_fifo.length)))
            student_fifo.head = student_fifo.base;
        else
            student_fifo.head++;
    }
    fclose(fp);
    DPRINTF("\n[INFO] Student Info added successfully\n");
}

void Student_count(void)
{
    DPRINTF("DataBase size is %d students\n",student_fifo.count);
}

void View_All(void)
{
    int size = student_fifo.count;
    for(int i = 0 ; i<size;i++)
    {
        DPRINTF("the student's first Name:\n");
        DPRINTF("%s\n",Buffer[i].first_name);
        DPRINTF("the student's second Name:\n");
        DPRINTF("%s\n",Buffer[i].second_name);
        DPRINTF("the student's Roll number:\n");
        DPRINTF("%d\n",Buffer[i].Roll_num);
        DPRINTF("the student's GPA:\n");
        DPRINTF("%.2lf\n",Buffer[i].GPA);
        DPRINTF("the student's registered courses are:[");
        for (int j = 0 ; j<5; j++)
        {
            if(j==4){ DPRINTF("%s]\n",Buffer[i].courses[j]); }
            else{ DPRINTF("%s,",Buffer[i].courses[j]); }
        }
        DPRINTF("=======================================\n");
    }
}


void Find_Student_FName(void)
{
    uint32_t size = student_fifo.count;
    char temp_text [40];
    DPRINTF("SEARCH : Enter the student's first Name:\n");
    gets(temp_text);
    DPRINTF("=======================================\n");
    char foundflag = 0;

    for(int i = 0 ; i<size;i++)
    {
        if(strcmp(temp_text,Buffer[i].first_name)==0){
        DPRINTF("the student's first Name:\n");
        DPRINTF("%s\n",Buffer[i].first_name);
        DPRINTF("the student's second Name:\n");
        DPRINTF("%s\n",Buffer[i].second_name);
        DPRINTF("the student's Roll number:\n");
        DPRINTF("%d\n",Buffer[i].Roll_num);
        DPRINTF("the student's GPA:\n");
        DPRINTF("%.2lf\n",Buffer[i].GPA);
        DPRINTF("the student's registered courses are:[");
        for (int j = 0 ; j<5; j++)
        {
            if(j==4){ DPRINTF("%s]\n",Buffer[i].courses[j]); }
            else{ DPRINTF("%s,",Buffer[i].courses[j]); }
        }
        foundflag = 1;
        DPRINTF("=======================================\n");
        }
    }
    if(foundflag == 0) {DPRINTF("Couldn't find a match\n");}
}

void Find_Student_RollN(void)
{
    uint32_t size = student_fifo.count;
    char temp_text [40];
    DPRINTF("SEARCH : Enter the student's Roll Number:\n");
    gets(temp_text);
    uint32_t ID = atoi(temp_text);
    DPRINTF("=======================================\n");
    char foundflag = 0;

    for(int i = 0 ; i<size;i++)
    {
        if(ID == Buffer[i].Roll_num){
        DPRINTF("the student's first Name:\n");
        DPRINTF("%s\n",Buffer[i].first_name);
        DPRINTF("the student's second Name:\n");
        DPRINTF("%s\n",Buffer[i].second_name);
        DPRINTF("the student's Roll number:\n");
        DPRINTF("%d\n",Buffer[i].Roll_num);
        DPRINTF("the student's GPA:\n");
        DPRINTF("%.2lf\n",Buffer[i].GPA);
        DPRINTF("the student's registered courses are:[");
        for (int j = 0 ; j<5; j++)
        {
            if(j==4){ DPRINTF("%s]\n",Buffer[i].courses[j]); }
            else{ DPRINTF("%s,",Buffer[i].courses[j]); }
        }
        foundflag = 1;
        DPRINTF("=======================================\n");
        }
    }
    if(foundflag == 0) {DPRINTF("Couldn't find a match\n");}
}

void Find_Student_C_ID(void)
{
    uint32_t size = student_fifo.count;
    char temp_text [40];
    DPRINTF("SEARCH : Enter the student's Course:\n");
    gets(temp_text);
    DPRINTF("=======================================\n");
    char foundflag = 0;

    for(int i = 0 ; i<size;i++)
    {
        for(int z = 0 ; z < 5 ; z++){
            if(strcmp(temp_text,Buffer[i].courses[z]) == 0)
            {
                DPRINTF("the student's first Name:\n");
                DPRINTF("%s\n",Buffer[i].first_name);
                DPRINTF("the student's second Name:\n");
                DPRINTF("%s\n",Buffer[i].second_name);
```



```c
                DPRINTF("the student's Roll number:\n");
                DPRINTF("%d\n",Buffer[i].Roll_num);
                DPRINTF("the student's GPA:\n");
                DPRINTF("%.2lf\n",Buffer[i].GPA);
                DPRINTF("the student's registered courses are:[");
                for (int j = 0 ; j<5; j++)
                {
                    if(j==4){ DPRINTF("%s]\n",Buffer[i].courses[j]); }
                    else{ DPRINTF("%s,",Buffer[i].courses[j]); }
                }
                foundflag = 1;
                DPRINTF("=======================================\n");
            }
        }
    }
    if(foundflag == 0) {DPRINTF("Couldn't find a match\n");}
}

void UPDT_Student_RollN(void)
{
    uint32_t size = student_fifo.count;
    char temp_text [40];
    DPRINTF("SEARCH : Enter the student's Roll Number:\n");
    gets(temp_text);
    uint32_t ID = atoi(temp_text);
    DPRINTF("=======================================\n");
    char foundflag = 0;

    for(int i = 0 ; i<size;i++)
    {
        if(ID == Buffer[i].Roll_num){
        DPRINTF("the student's first Name:\n");
        DPRINTF("%s\n",Buffer[i].first_name);
        DPRINTF("the student's second Name:\n");
        DPRINTF("%s\n",Buffer[i].second_name);
        DPRINTF("the student's Roll number:\n");
        DPRINTF("%d\n",Buffer[i].Roll_num);
        DPRINTF("the student's GPA:\n");
        DPRINTF("%.2lf\n",Buffer[i].GPA);
        DPRINTF("the student's registered courses are:[");
        for (int j = 0 ; j<5; j++)
        {
            if(j==4){ DPRINTF("%s]\n",Buffer[i].courses[j]); }
            else{ DPRINTF("%s,",Buffer[i].courses[j]); }
        }
        foundflag = 1;
        DPRINTF("=======================================\n");
        DPRINTF("UPDATING INFO PLEASE FILL THE FOLLOWING:\n");
        DPRINTF("=======================================\n");
        DPRINTF("Enter the student's first Name:\n");
        gets(temp_text);
        //DPRINTF("%s\n",temp_text);
        strcpy(Buffer[i].first_name , temp_text);
        DPRINTF("Enter the student's last Name:\n");
        gets(temp_text);
        strcpy(Buffer[i].second_name , temp_text);

        DPRINTF("Enter the student's Roll number:\n");
        gets(temp_text);
        Buffer[i].Roll_num = atoi(temp_text);

        DPRINTF("Enter the student's GPA:\n");
        gets(temp_text);
        Buffer[i].GPA = atof(temp_text);
        for(int i=0 ; i<5 ; i++)
            {
                DPRINTF("Enter the student's registered course number %d:\n",i+1);
                gets(temp_text);
                strcpy(Buffer[i].courses[i] , temp_text);

            }

        }
    }
    if(foundflag == 0) {DPRINTF("Couldn't find a match\n");}
}

void DLT_Student_RollN(void)
{
    char temp_text [40];
    DPRINTF("SEARCH : Enter the student's Roll Number:\n");
    gets(temp_text);
    uint32_t ID = atoi(temp_text);

    uint32_t size = student_fifo.count;

    Student_t * s ;
    Student_t * curr_s ;
    Student_t * next_s ;
    s = student_fifo.tail;
    DPRINTF("=======================================\n");
    char foundflag = 0;

    for(int i = 0 ; i<size;i++)
    {
        if(ID == Buffer[i].Roll_num){
            for(int j = i ; j<size;j++)
            {
                curr_s = s;
                next_s = ++s;
                *curr_s = *next_s;
            }
            student_fifo.count-- ;
            if(student_fifo.head == student_fifo.base)
            {
                student_fifo.head = student_fifo.base + student_fifo.length;
            }
            else
            {
                student_fifo.head--;
            }
            foundflag = 1;
            DPRINTF("DELETED SUCCESSFULLY\n");
        }
        s++;
    }
    if(foundflag == 0) {DPRINTF("Couldn't find a match\n");}
}

```

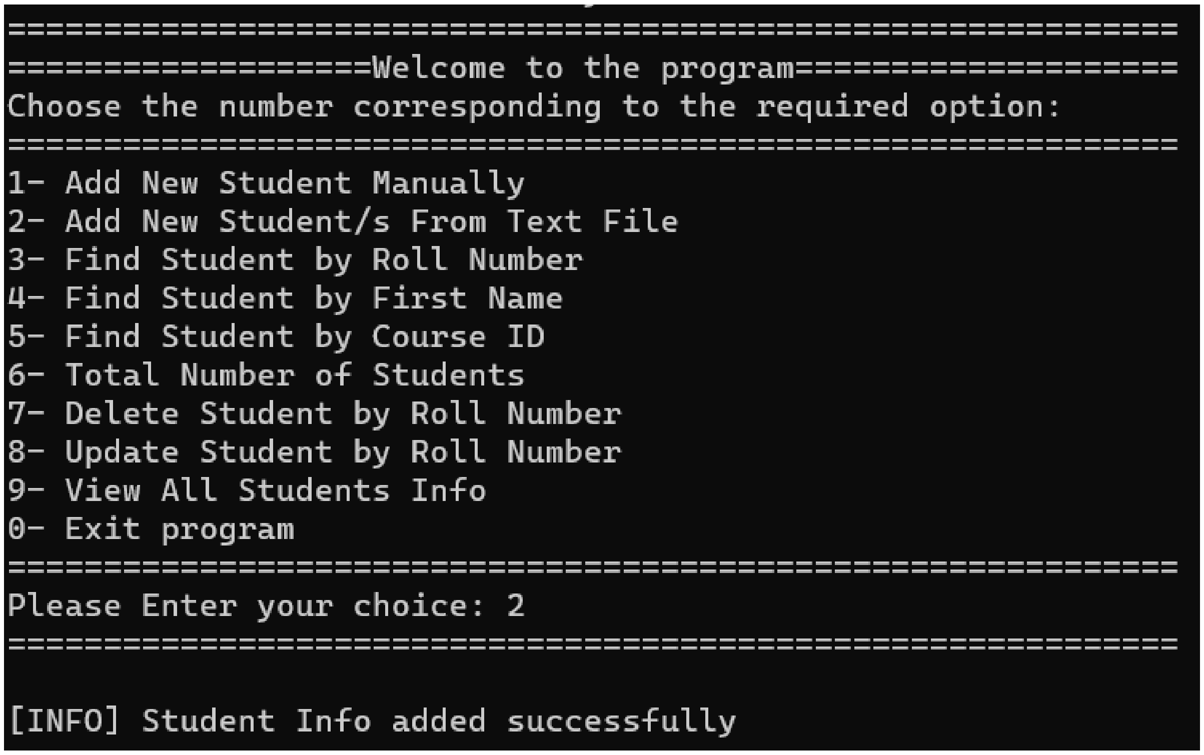
4 OUTPUT

1)Add Manual

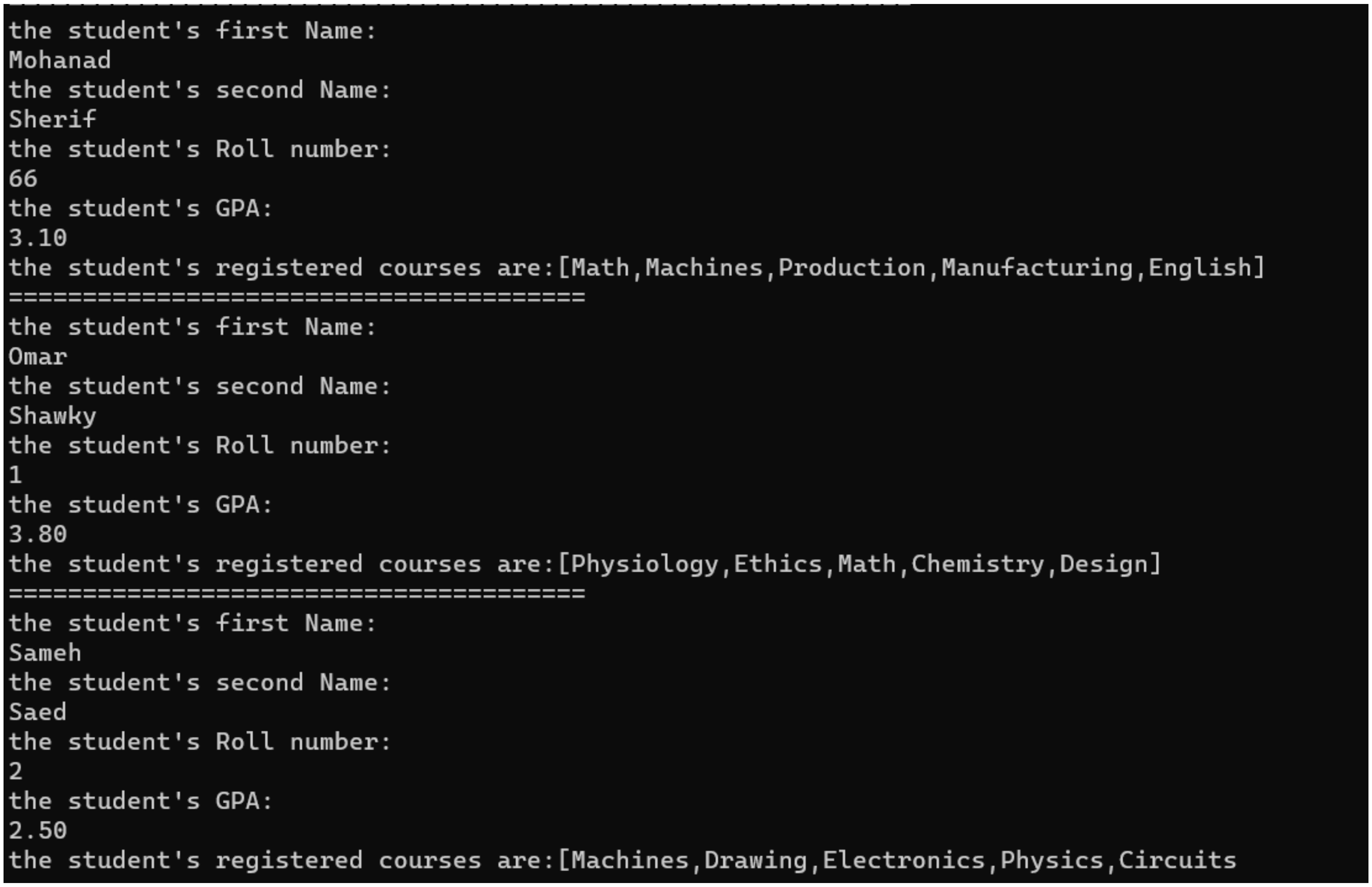


```
===================Welcome to the program====================
Choose the number corresponding to the required option:
=============================================================
1- Add New Student Manually
2- Add New Student/s From Text File
3- Find Student by Roll Number
4- Find Student by First Name
5- Find Student by Course ID
6- Total Number of Students
7- Delete Student by Roll Number
8- Update Student by Roll Number
9- View All Students Info
0- Exit program
=============================================================
Please Enter your choice: 1
=============================================================
Enter the student's first Name:
Mohanad
Enter the student's last Name:
Sherif
Enter the student's Roll number:
66
Enter the student's GPA:
3.1
Enter the student's registered course number 1:
Math
Enter the student's registered course number 2:
Machines
Enter the student's registered course number 3:
Production
```

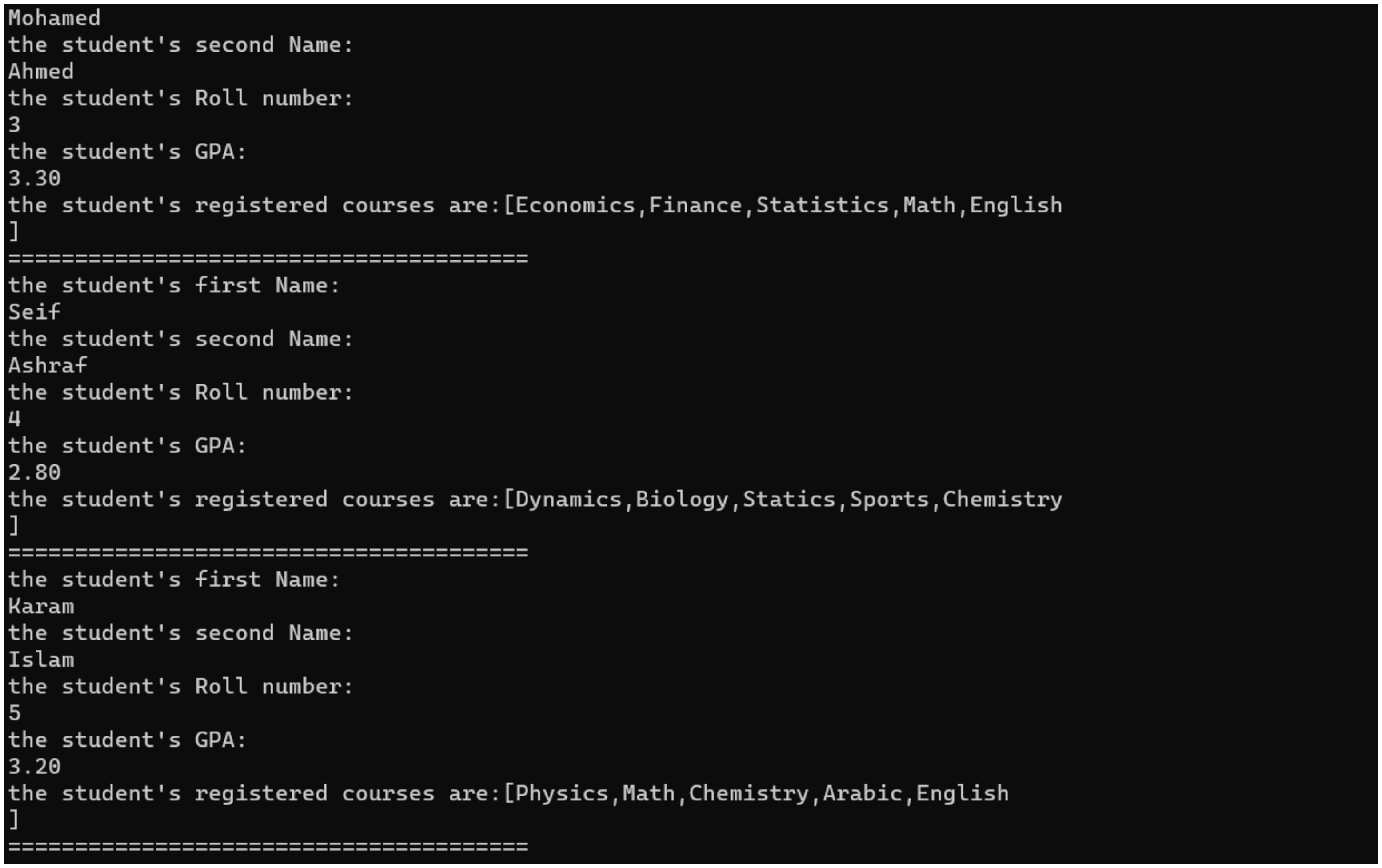
2)Add from text



```
==============================================================
===================Welcome to the program=====================
Choose the number corresponding to the required option:
==============================================================
1- Add New Student Manually
2- Add New Student/s From Text File
3- Find Student by Roll Number
4- Find Student by First Name
5- Find Student by Course ID
6- Total Number of Students
7- Delete Student by Roll Number
8- Update Student by Roll Number
9- View All Students Info
0- Exit program
==============================================================
Please Enter your choice: 2
==============================================================

[INFO] Student Info added successfully
```

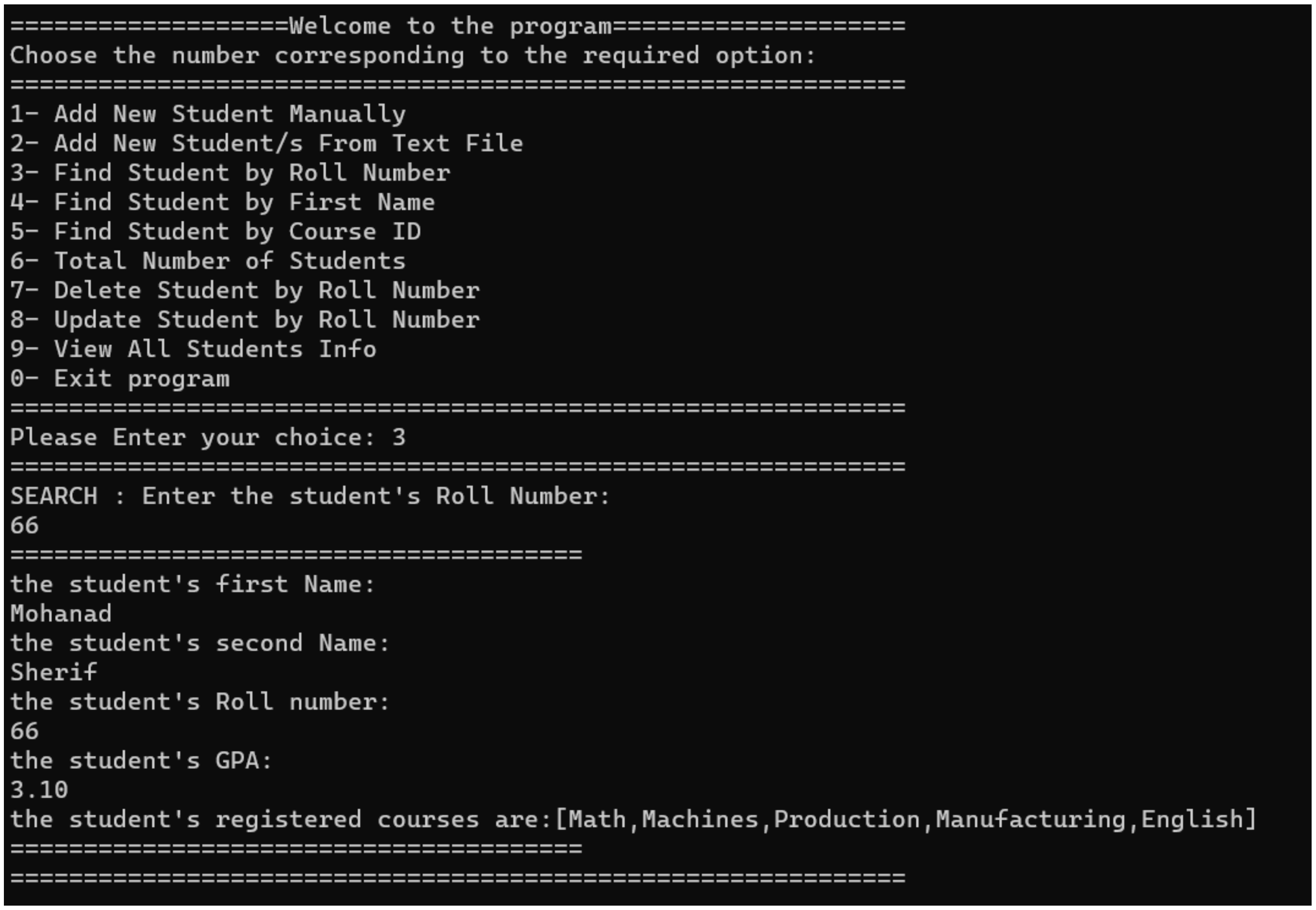


```
the student's first Name:
Mohanad
the student's second Name:
Sherif
the student's Roll number:
66
the student's GPA:
3.10
the student's registered courses are:[Math,Machines,Production,Manufacturing,English]
=======================================
the student's first Name:
Omar
the student's second Name:
Shawky
the student's Roll number:
1
the student's GPA:
3.80
the student's registered courses are:[Physiology,Ethics,Math,Chemistry,Design]
=======================================
the student's first Name:
Sameh
the student's second Name:
Saed
the student's Roll number:
2
the student's GPA:
2.50
the student's registered courses are:[Machines,Drawing,Electronics,Physics,Circuits
```



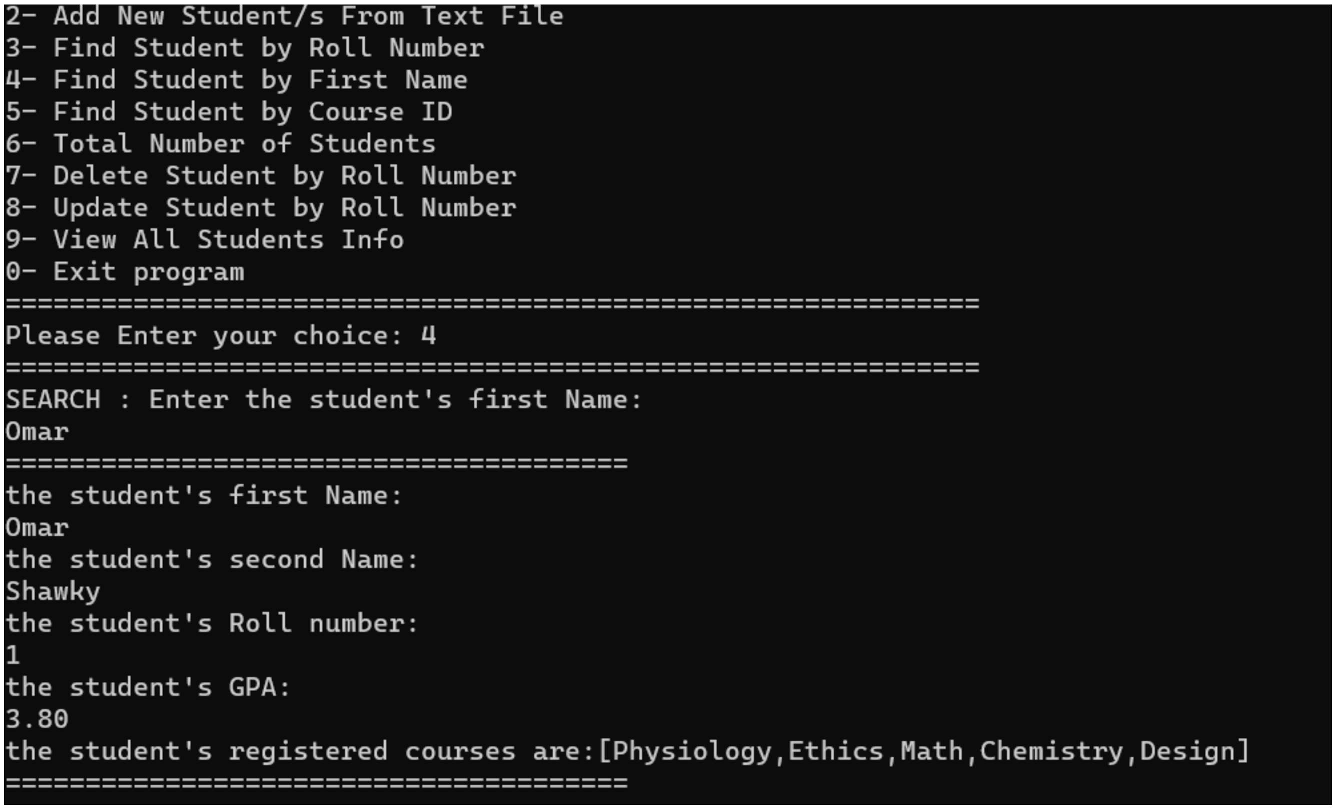
```
Mohamed
the student's second Name:
Ahmed
the student's Roll number:
3
the student's GPA:
3.30
the student's registered courses are:[Economics,Finance,Statistics,Math,English
]
=======================================
the student's first Name:
Seif
the student's second Name:
Ashraf
the student's Roll number:
4
the student's GPA:
2.80
the student's registered courses are:[Dynamics,Biology,Statics,Sports,Chemistry
]
=======================================
the student's first Name:
Karam
the student's second Name:
Islam
the student's Roll number:
5
the student's GPA:
3.20
the student's registered courses are:[Physics,Math,Chemistry,Arabic,English
]
=======================================
```

3)Find by Roll



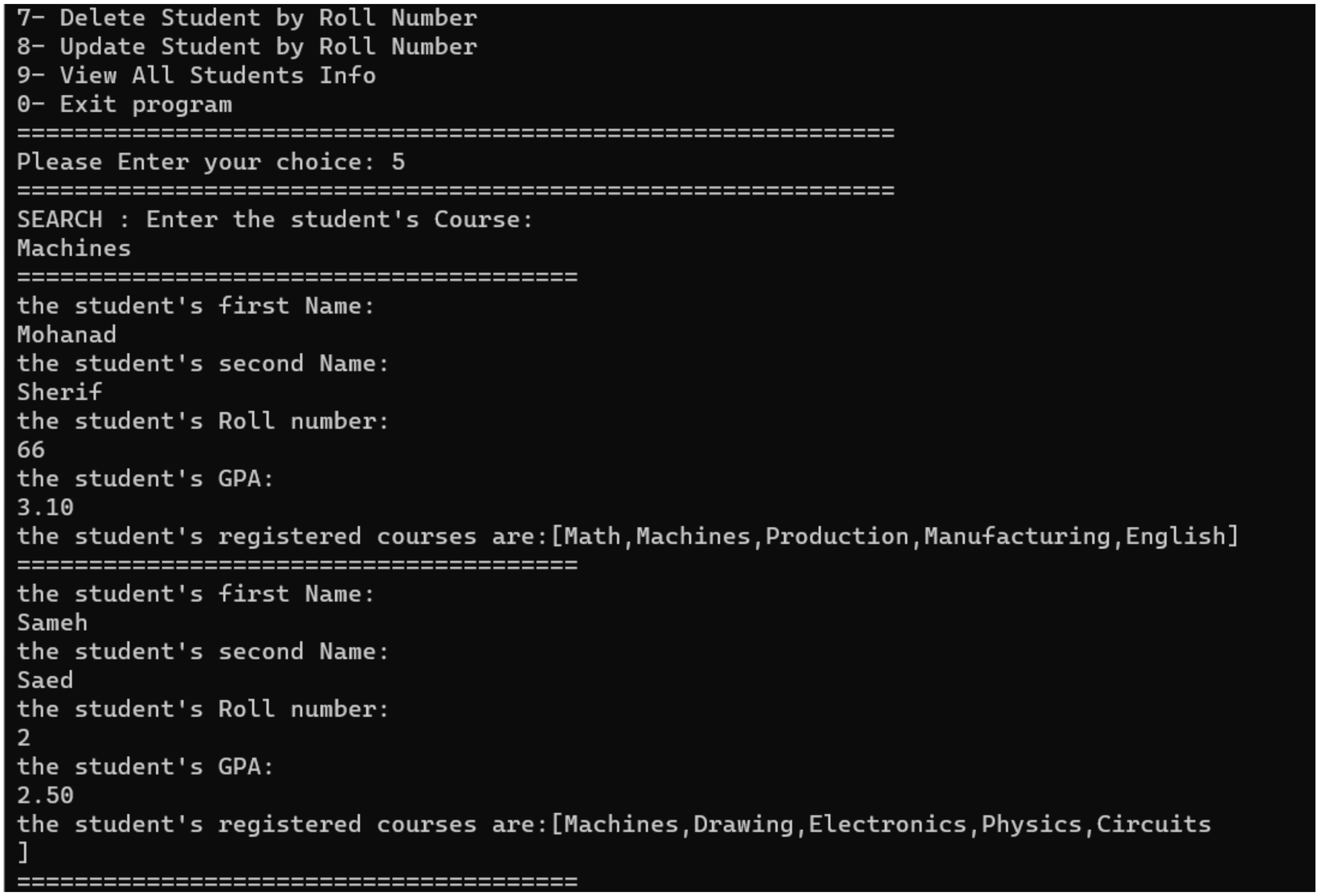
```
===================Welcome to the program====================
Choose the number corresponding to the required option:
=============================================================
1- Add New Student Manually
2- Add New Student/s From Text File
3- Find Student by Roll Number
4- Find Student by First Name
5- Find Student by Course ID
6- Total Number of Students
7- Delete Student by Roll Number
8- Update Student by Roll Number
9- View All Students Info
0- Exit program
=============================================================
Please Enter your choice: 3
=============================================================
SEARCH : Enter the student's Roll Number:
66
=======================================
the student's first Name:
Mohanad
the student's second Name:
Sherif
the student's Roll number:
66
the student's GPA:
3.10
the student's registered courses are:[Math,Machines,Production,Manufacturing,English]
=======================================
=============================================================
```

4)Find by First Name



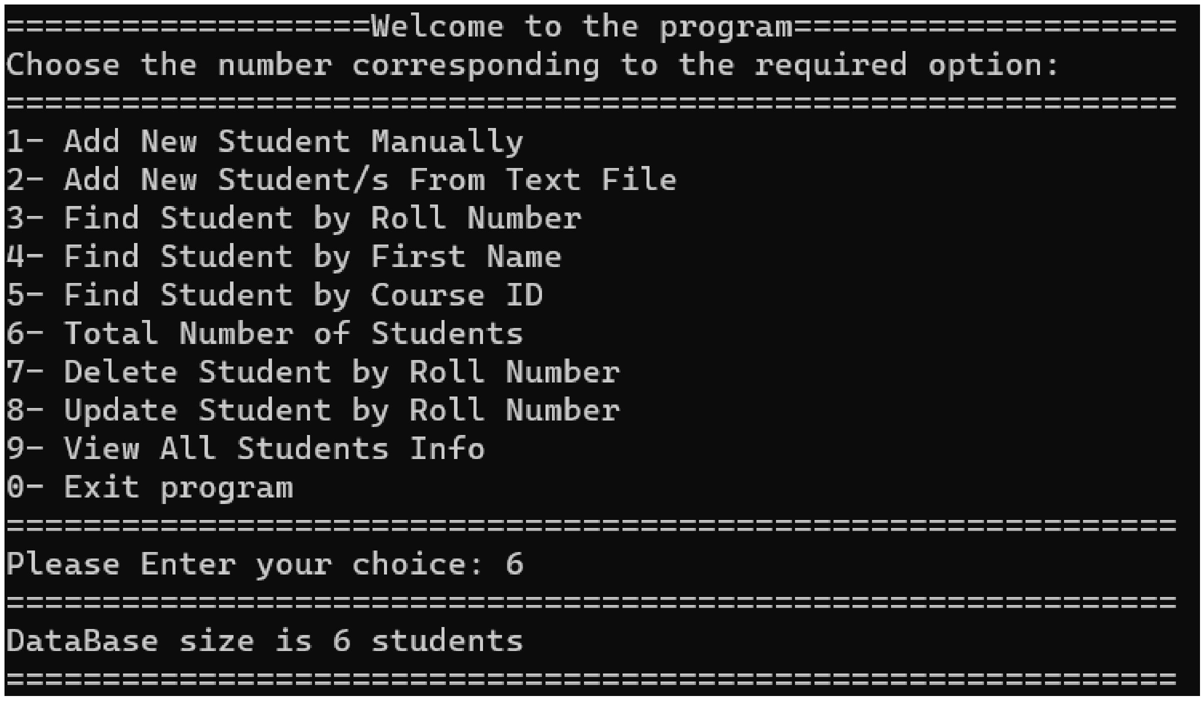
```
2- Add New Student/s From Text File
3- Find Student by Roll Number
4- Find Student by First Name
5- Find Student by Course ID
6- Total Number of Students
7- Delete Student by Roll Number
8- Update Student by Roll Number
9- View All Students Info
0- Exit program
=============================================================
Please Enter your choice: 4
=============================================================
SEARCH : Enter the student's first Name:
Omar
=======================================
the student's first Name:
Omar
the student's second Name:
Shawky
the student's Roll number:
1
the student's GPA:
3.80
the student's registered courses are:[Physiology,Ethics,Math,Chemistry,Design]
=======================================
```

5)Find by Course



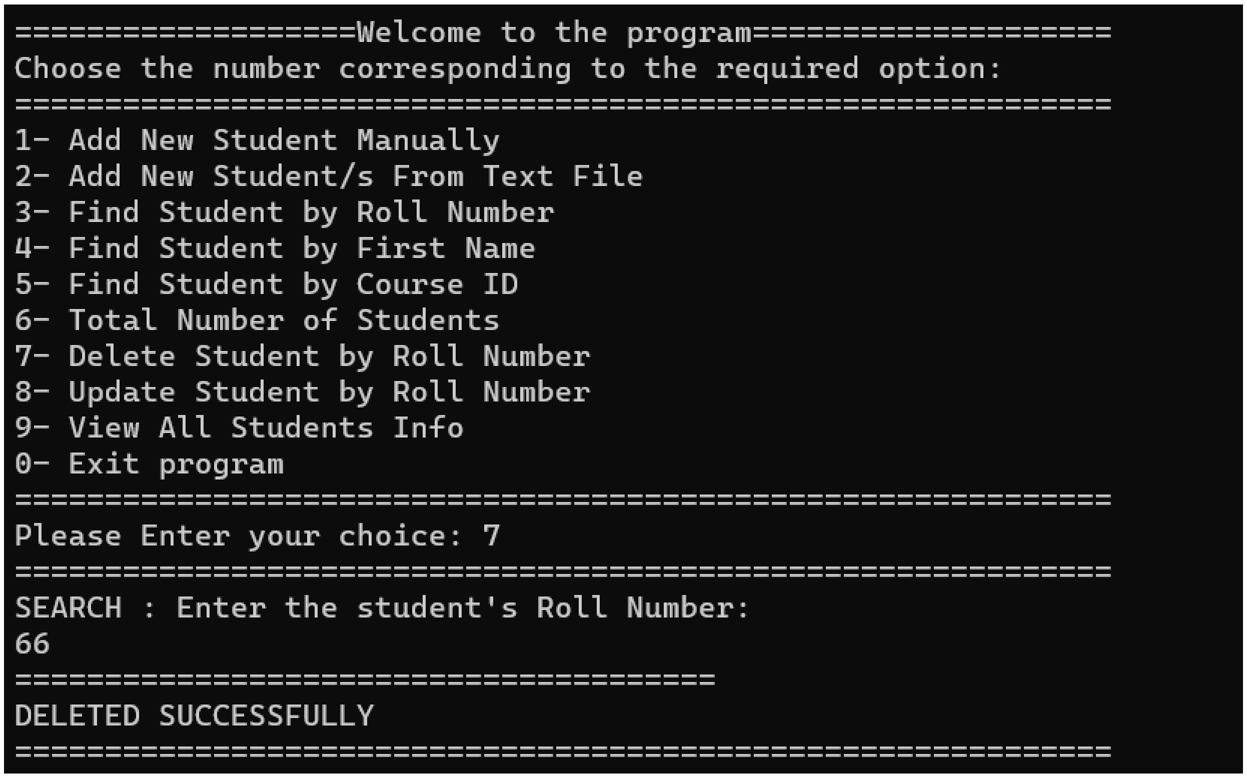
```
7- Delete Student by Roll Number
8- Update Student by Roll Number
9- View All Students Info
0- Exit program
==============================================================
Please Enter your choice: 5
==============================================================
SEARCH : Enter the student's Course:
Machines
=======================================
the student's first Name:
Mohanad
the student's second Name:
Sherif
the student's Roll number:
66
the student's GPA:
3.10
the student's registered courses are:[Math,Machines,Production,Manufacturing,English]
=======================================
the student's first Name:
Sameh
the student's second Name:
Saed
the student's Roll number:
2
the student's GPA:
2.50
the student's registered courses are:[Machines,Drawing,Electronics,Physics,Circuits
]
=======================================
```

6)Print total number of students in the database

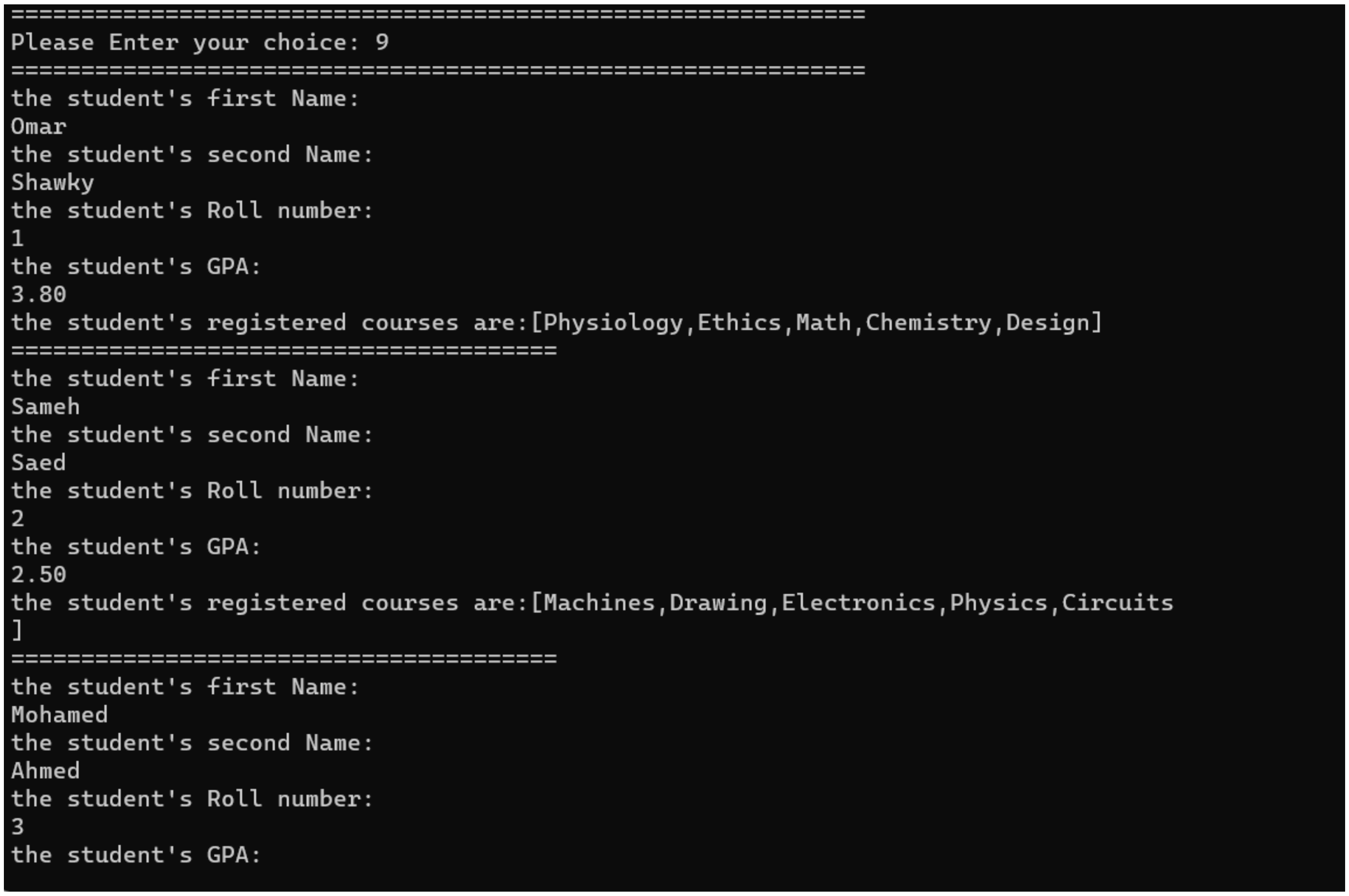


```
==================Welcome to the program====================
Choose the number corresponding to the required option:
============================================================
1- Add New Student Manually
2- Add New Student/s From Text File
3- Find Student by Roll Number
4- Find Student by First Name
5- Find Student by Course ID
6- Total Number of Students
7- Delete Student by Roll Number
8- Update Student by Roll Number
9- View All Students Info
0- Exit program
============================================================
Please Enter your choice: 6
============================================================
DataBase size is 6 students
============================================================
```

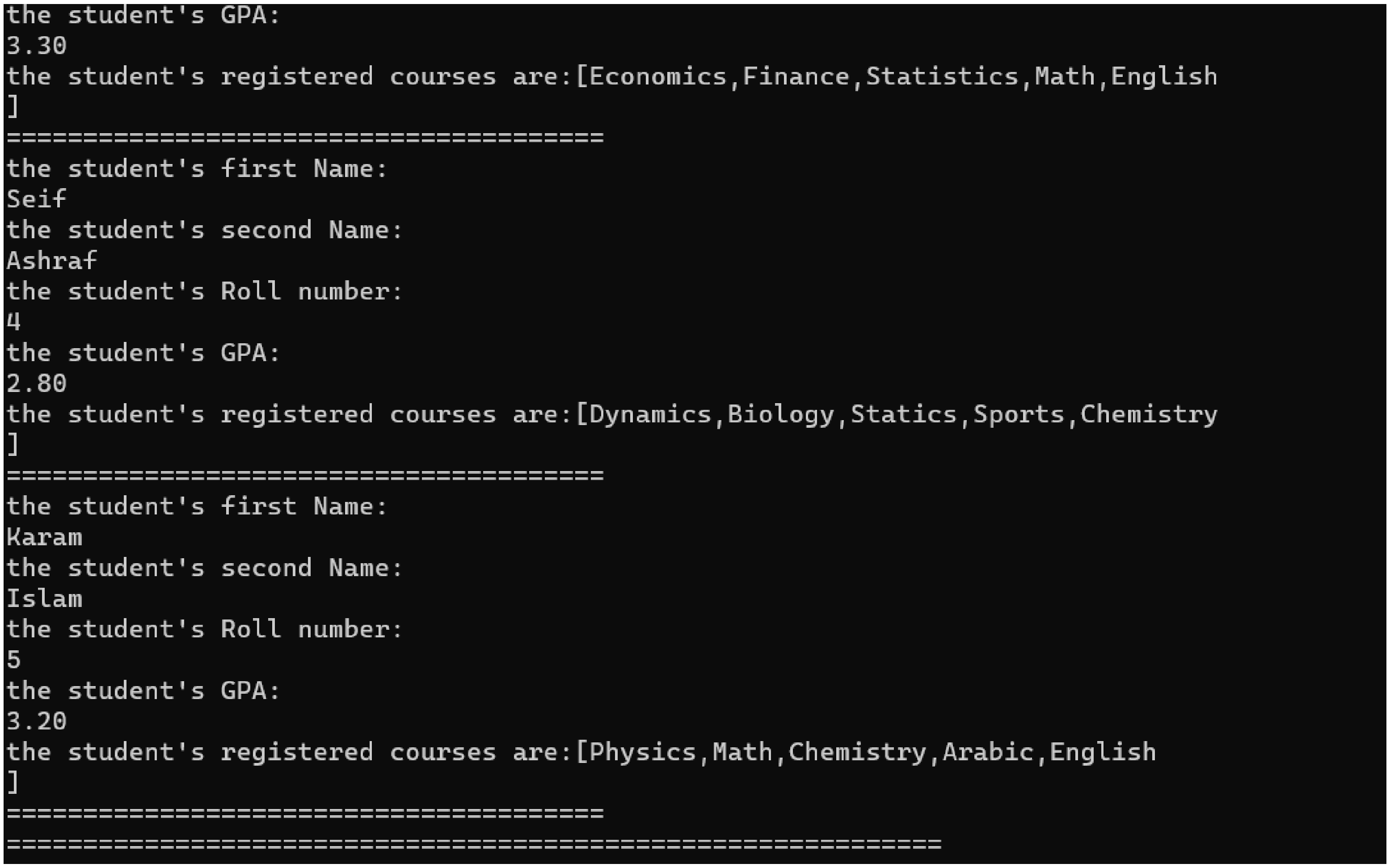
7)Delete student using roll number



```
===================Welcome to the program====================
Choose the number corresponding to the required option:
=============================================================
1- Add New Student Manually
2- Add New Student/s From Text File
3- Find Student by Roll Number
4- Find Student by First Name
5- Find Student by Course ID
6- Total Number of Students
7- Delete Student by Roll Number
8- Update Student by Roll Number
9- View All Students Info
0- Exit program
=============================================================
Please Enter your choice: 7
=============================================================
SEARCH : Enter the student's Roll Number:
66
=======================================
DELETED SUCCESSFULLY
=============================================================
```

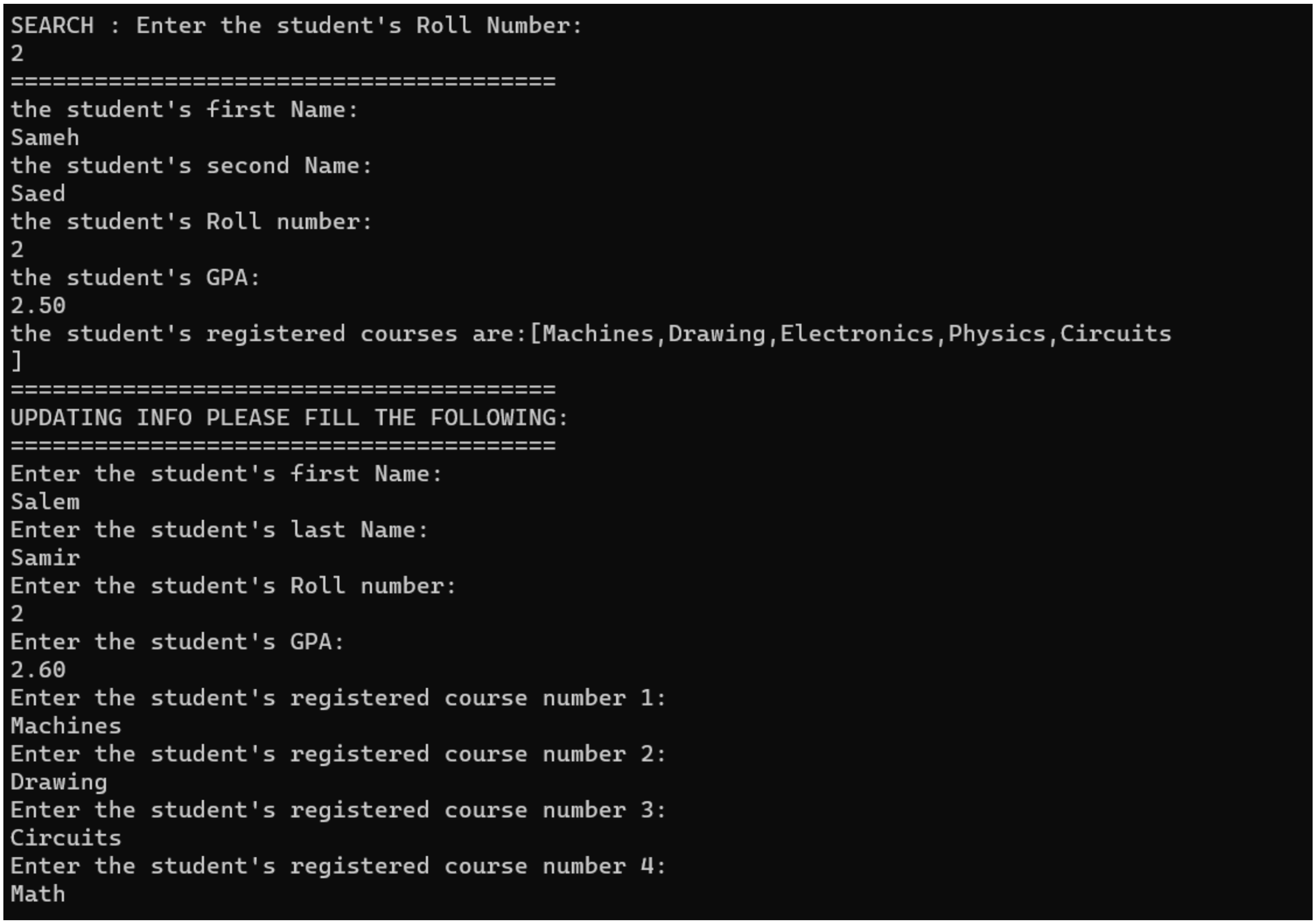


```
=============================================================
Please Enter your choice: 9
=============================================================
the student's first Name:
Omar
the student's second Name:
Shawky
the student's Roll number:
1
the student's GPA:
3.80
the student's registered courses are:[Physiology,Ethics,Math,Chemistry,Design]
=======================================
the student's first Name:
Sameh
the student's second Name:
Saed
the student's Roll number:
2
the student's GPA:
2.50
the student's registered courses are:[Machines,Drawing,Electronics,Physics,Circuits
]
=======================================
the student's first Name:
Mohamed
the student's second Name:
Ahmed
the student's Roll number:
3
the student's GPA:
```

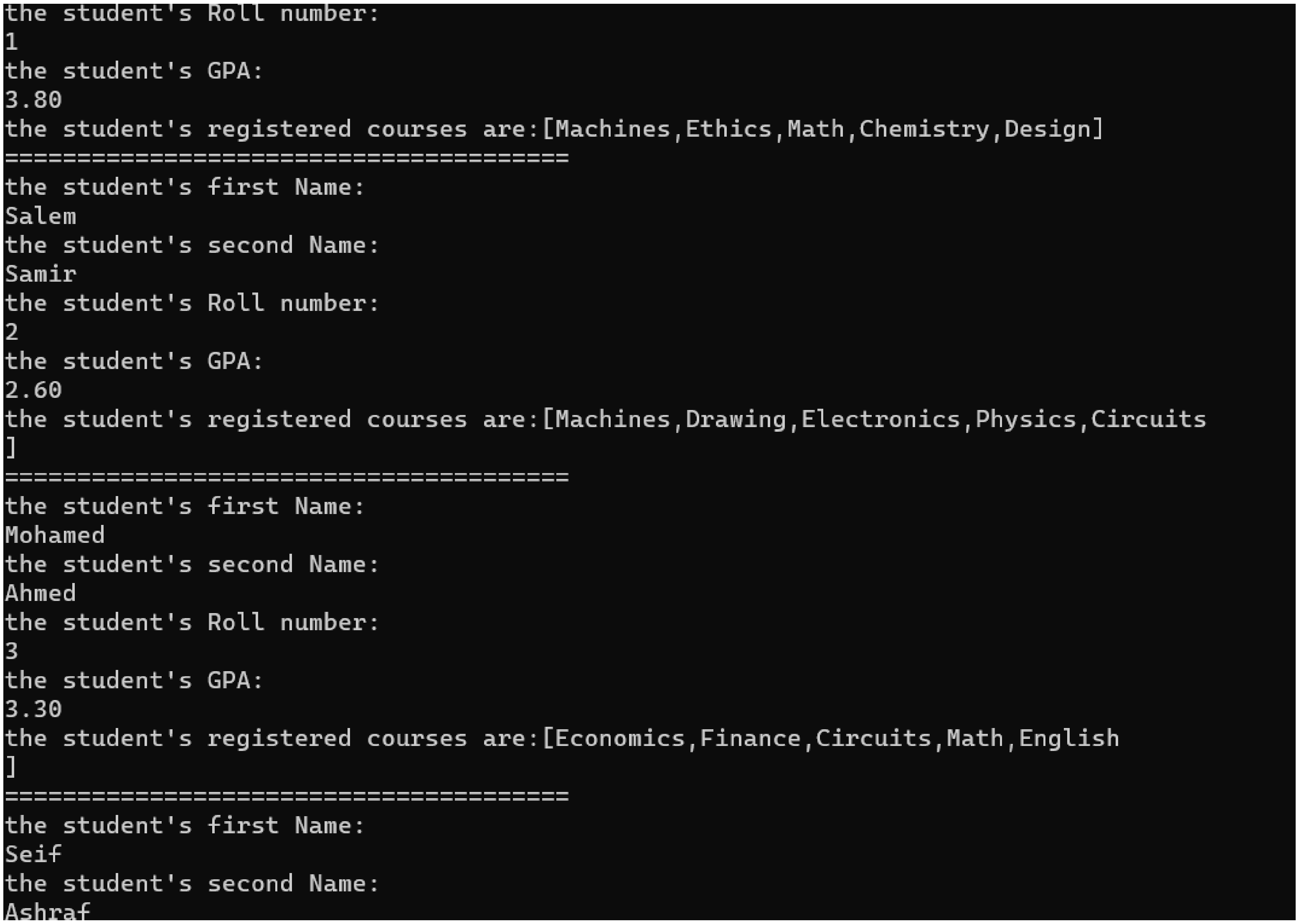


```
the student's GPA:
3.30
the student's registered courses are:[Economics,Finance,Statistics,Math,English
]
=======================================
the student's first Name:
Seif
the student's second Name:
Ashraf
the student's Roll number:
4
the student's GPA:
2.80
the student's registered courses are:[Dynamics,Biology,Statics,Sports,Chemistry
]
=======================================
the student's first Name:
Karam
the student's second Name:
Islam
the student's Roll number:
5
the student's GPA:
3.20
the student's registered courses are:[Physics,Math,Chemistry,Arabic,English
]
=======================================
=============================================================
```

8)Update student using roll number

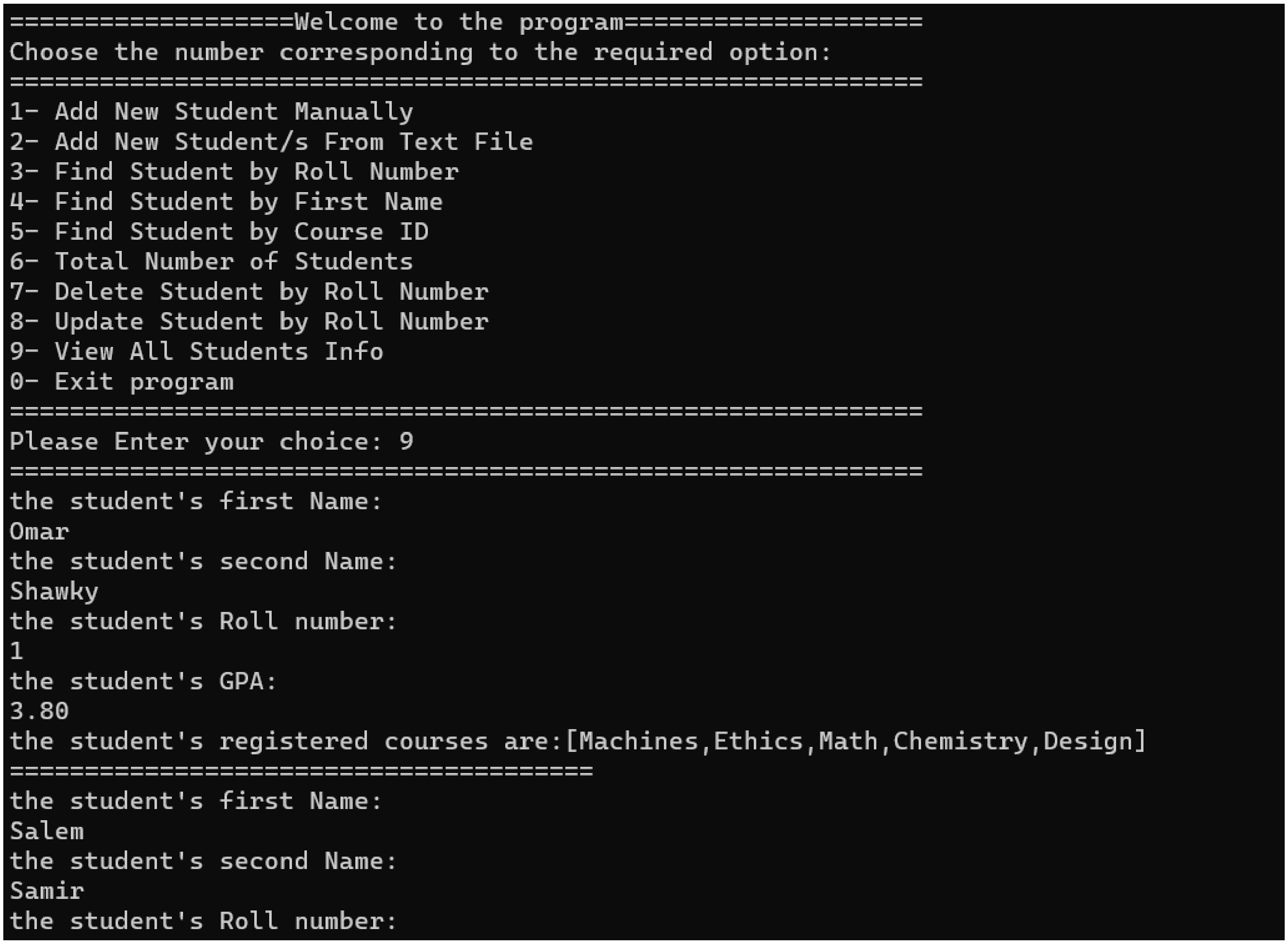


```
SEARCH : Enter the student's Roll Number:
2
=======================================
the student's first Name:
Sameh
the student's second Name:
Saed
the student's Roll number:
2
the student's GPA:
2.50
the student's registered courses are:[Machines,Drawing,Electronics,Physics,Circuits
]
=======================================
UPDATING INFO PLEASE FILL THE FOLLOWING:
=======================================
Enter the student's first Name:
Salem
Enter the student's last Name:
Samir
Enter the student's Roll number:
2
Enter the student's GPA:
2.60
Enter the student's registered course number 1:
Machines
Enter the student's registered course number 2:
Drawing
Enter the student's registered course number 3:
Circuits
Enter the student's registered course number 4:
Math
```

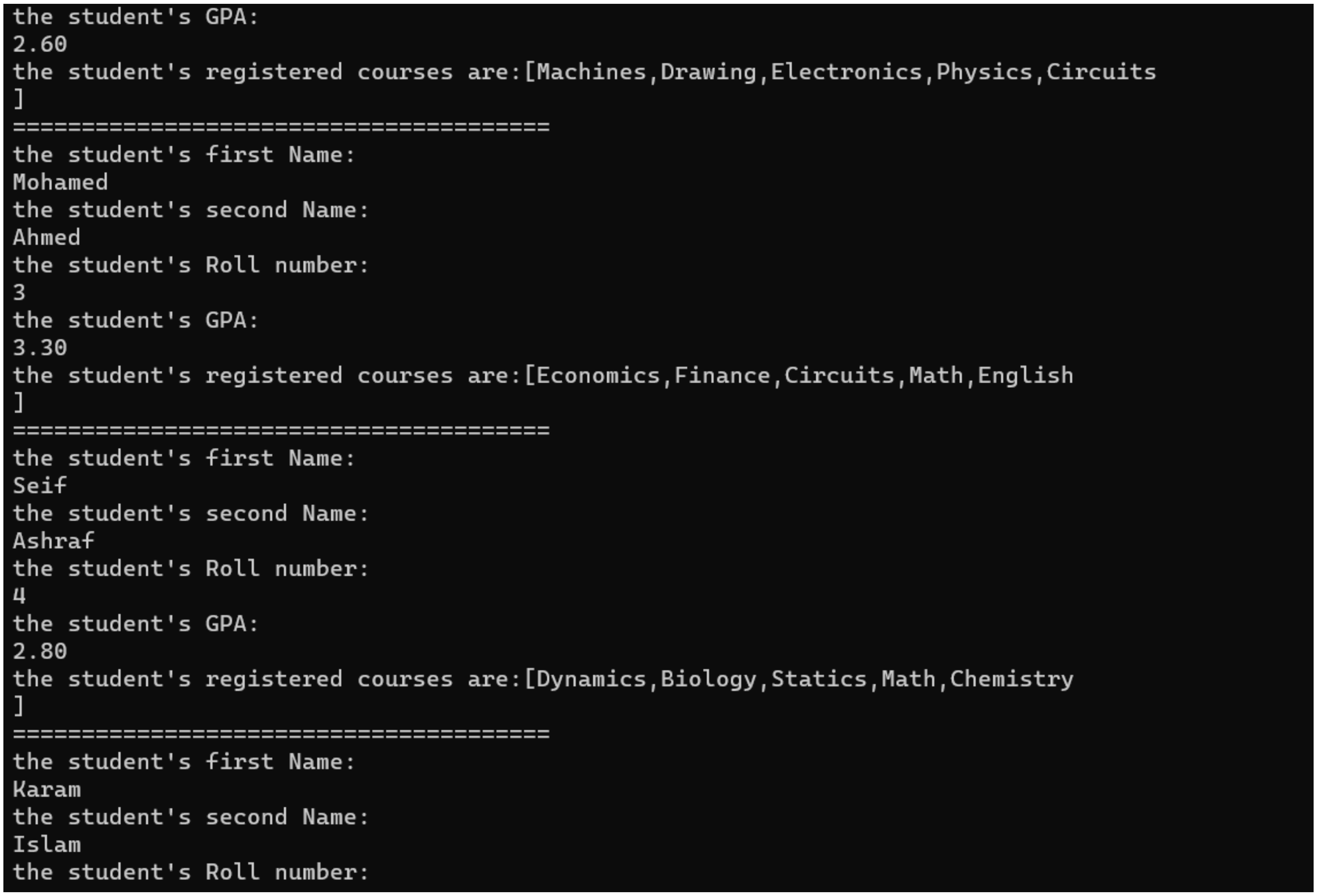


```
the student's Roll number:
1
the student's GPA:
3.80
the student's registered courses are:[Machines,Ethics,Math,Chemistry,Design]
=======================================
the student's first Name:
Salem
the student's second Name:
Samir
the student's Roll number:
2
the student's GPA:
2.60
the student's registered courses are:[Machines,Drawing,Electronics,Physics,Circuits
]
=======================================
the student's first Name:
Mohamed
the student's second Name:
Ahmed
the student's Roll number:
3
the student's GPA:
3.30
the student's registered courses are:[Economics,Finance,Circuits,Math,English
]
=======================================
the student's first Name:
Seif
the student's second Name:
Ashraf
```

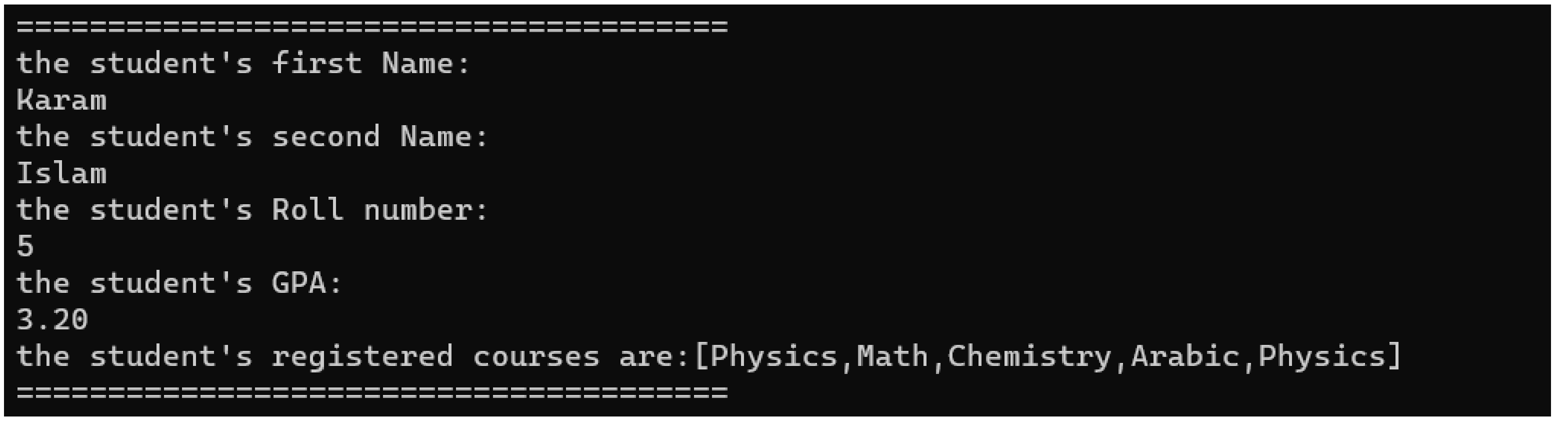
9)View the entire Data base



```
===================Welcome to the program====================
Choose the number corresponding to the required option:
=============================================================
1- Add New Student Manually
2- Add New Student/s From Text File
3- Find Student by Roll Number
4- Find Student by First Name
5- Find Student by Course ID
6- Total Number of Students
7- Delete Student by Roll Number
8- Update Student by Roll Number
9- View All Students Info
0- Exit program
=============================================================
Please Enter your choice: 9
=============================================================
the student's first Name:
Omar
the student's second Name:
Shawky
the student's Roll number:
1
the student's GPA:
3.80
the student's registered courses are:[Machines,Ethics,Math,Chemistry,Design]
=======================================
the student's first Name:
Salem
the student's second Name:
Samir
the student's Roll number:
```



```
the student's GPA:
2.60
the student's registered courses are:[Machines,Drawing,Electronics,Physics,Circuits
]
=======================================
the student's first Name:
Mohamed
the student's second Name:
Ahmed
the student's Roll number:
3
the student's GPA:
3.30
the student's registered courses are:[Economics,Finance,Circuits,Math,English
]
=======================================
the student's first Name:
Seif
the student's second Name:
Ashraf
the student's Roll number:
4
the student's GPA:
2.80
the student's registered courses are:[Dynamics,Biology,Statics,Math,Chemistry
]
=======================================
the student's first Name:
Karam
the student's second Name:
Islam
the student's Roll number:
```



```
=======================================
the student's first Name:
Karam
the student's second Name:
Islam
the student's Roll number:
5
the student's GPA:
3.20
the student's registered courses are:[Physics,Math,Chemistry,Arabic,Physics]
=======================================
```