BAHRIA UNIVERSITY,
(Karachi Campus)

Department of Software Engineering

ASSIGNMENT #: 2 – Fall 2022

COURSE TITLE:	**System Programming**	COURSE CODE:	**CEN-449**
Class:	**BSE - 5(B)**	Shift:	**Morning**
Course Instructor:	**Engr. Rizwan Fazal**	Date:	**09-DEC-2022**
Due Date:	**23-December-2022**	Max. Marks:	**10 Points**

Instructions:

1. **This is a *Design and Implementation* project which will enable you to understand the System Software development using few important system aspects.**
2. **It is given as an individual task.**
3. **A project report needs to be submitted.**
4. **Attach the title page as given at page #. 3 of this document with completely filled information as required.**
5. **The given deadline is final and will not be extended in any case, therefore, it is highly recommended to start it as soon as possible. Strictly follow the instructions please!**

Project Description: [CLO-3]

In this project work, you have to consider the following design aspects:

- Pointers to Functions
- Abstract Data Type (like Structures in C/C++)
- Dynamic Memory Allocation

Project Operation:

In this project work, you have to implement the same application as you did in the previous assignment (Assignment #. 1), but this time you have to use the concept of 'Pointers to Functions'. The idea is to decompose the whole single application into multiple modules (functions) which will improve application's flexibility and reliability.

Implement individual functions for the below cited options and just place function calls in the main application where appropriate. It is important to note that all the functions must be called using 'Pointer' notation. You also have the option to use 'Array of Function Pointers' if it is believed to be a viable option. So, you can choose the implementation scheme considering the requirements of the application.

*** Student Database System ***

1. *Enter Student Information*
2. *Display Student Information*
3. *Edit Student Information*
4. *Delete Student Information*

The user should be able to do any of the above by pressing the corresponding 'number' on the screen. So, once the above menu is displayed on the screen, the program must wait for an input from the user to respond.

As the memory is allocated dynamically, so it is very important to release the acquired memory back to the system to avoid 'Memory Leakage'.

The program also keeps track of the number of students whose information has been added in the record. For instance, if the user initially mentioned that there will be 50 students but provide the information for only 10 students at that time, so it must be known to the program itself to avoid overwriting.

There will be a bound check also in the program like if user enters '50' in response to a message 'Number of Students', it shouldn't go beyond that point which can possibly lead to application crash.

Report

You have to submit the Assignment on the LMS which must include the following.

1. The title page as given here on the next page
2. The source code (either in C or C++ languages), preferably with comments
3. Screen shots of program execution of all the four options mentioned above
4. Recommendations for improvement in the program in any way

BAHRIA UNIVERSITY,
(Karachi Campus)

Department of Software Engineering

ASSIGNMENT #: 2 – Fall 2022

COURSE TITLE:	**System Programming**	COURSE CODE:	**CEN-449**
Class:	**BSE - 5(B)**	Shift:	**Morning**
Course Instructor:	**Engr. Rizwan Fazal**	Date:	**09-Dec-2022**
Due Date:	**23-December-2022**	Max. Marks:	**10 Points**

ASSIGNMENT #. 2

Name: M Muaz Shahzad

Enrolment #: 02-131202-081

Class: BSE 5th B

Software Engineering Department
Bahria University, Karachi Campus

Code

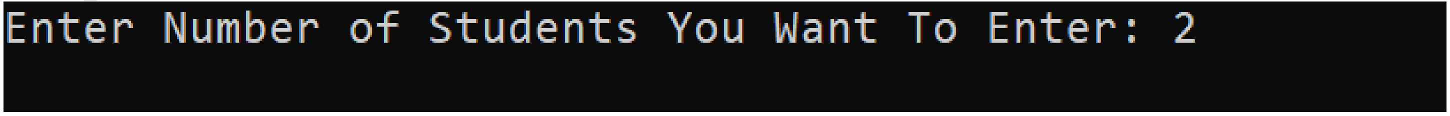
```
// ConsoleApplication1.cpp : This file contains the 'main' function. Program execution begins and ends
there.
//

#include <iostream>
#include <stdlib.h>
#include <string>
using namespace std;
struct Student {
        string Name;
        string Enrollment;
        int Semester;
        double CGPA;
public:
        Student()
        {
                CGPA = 0.0;
                Semester = 0;
        }
};
void Menu()
{
        system("CLS");
        cout << "\t\t-------------"<< endl;
        cout << "\t\tDisplay Menu" << endl;
        cout << "\t\t-------------" << endl << endl;
        cout << "1. Add Student Information " << endl;
        cout << "2. Display Student Information " << endl;
        cout << "3. Edit Student Information " << endl;
        cout << "4. Delete Student Information " << endl;
        cout << "5. Exit " << endl;
}
void addingStudent(Student* std, int& noOfStudents)
{
        cout << "\nEnter Student's Name: ";
        cin.ignore(); // ignore one or more characters from input buffer
        getline(cin, std[noOfStudents].Name); // string with spaces
        cout << "Enter Semester No: ";
        cin >> std[noOfStudents].Semester;
        cout << "Enter Enrollment: ";
        cin >> std[noOfStudents].Enrollment;
        cout << "Enter CGPA: ";
        cin >> std[noOfStudents].CGPA;
        ++noOfStudents;
}
void displayStudentData(Student* std, int& noOfStudents)
{
        system("CLS");
        for (int i = 0; i < noOfStudents; i++)
        {
                if (std[i].Enrollment == "0000") // skip the data of deleted student
                {
                        continue;
                }
                cout << "Name: " << std[i].Name << endl;
                cout << "Enrollment: " << std[i].Enrollment << endl;
                cout << "Semester: " << std[i].Semester << endl;
                cout << "CGPA: " << std[i].CGPA << endl;
```

```cpp
        cout << endl;
    }
    system("pause");
}
void editStudentData(Student* std, int& noOfStudents)
{
    system("CLS");
    string student;
    char entry;
    cout << "Student's Enroll Num To Edit Data: ";
    cin.ignore();
    getline(cin, student);
    for (int i = 0; i < noOfStudents; i++)
    {
        if (std[i].Enrollment == student)
        {
            cout << "Data of " << std[i].Name << endl << endl;
            cout << "What Do You Want To Edit ? " << endl;
            cout << "1. Edit Student's Name " << endl;
            cout << "2. Edit Student's Enrollment " << endl;
            cout << "3. Edit Student's Semester " << endl;
            cout << "4. Edit Student's CGPA " << endl;
            cout << "\n Enter Your Choice: ";
            cin >> entry;
            system("CLS");
            if (entry == '1')
            {
                cout << "Name of Student: " << std[i].Name << endl;
                cout << "Enter New Name: ";
                cin >> std[i].Name;
                cout << "Name " << std[i].Name << " Updated!" << endl;
            }
            else if (entry == '2')
            {
                cout << "Enroll of Student: " << std[i].Enrollment
                    << endl;
                cout << "Enter New Enroll: ";
                cin >> std[i].Enrollment;
                cout << "Enrollment " << std[i].Name << " Updated !" << endl;
            }
            else if (entry == '3')
            {
                cout << "Semester of Student: " << std[i].Semester <<
                    endl;
                cout << "Enter New Semester No.: ";
                cin >> std[i].Semester;
                cout << "Semester " << std[i].Name << " Updated !" << endl;
            }
            else if (entry == '4')
            {
                cout << "CGPA of Student: " << std[i].CGPA << endl;
                cout << "Enter New CGPA: ";
                cin >> std[i].CGPA;
                cout << "GPA of " << std[i].Name << " Updated !" << endl;
            }
            else
                cout << "Wrong Input: " << endl;
            system("pause");
        }
    }
}
void deleteStudentData(Student* std, int& noOfStudents)
{
    system("CLS");
```

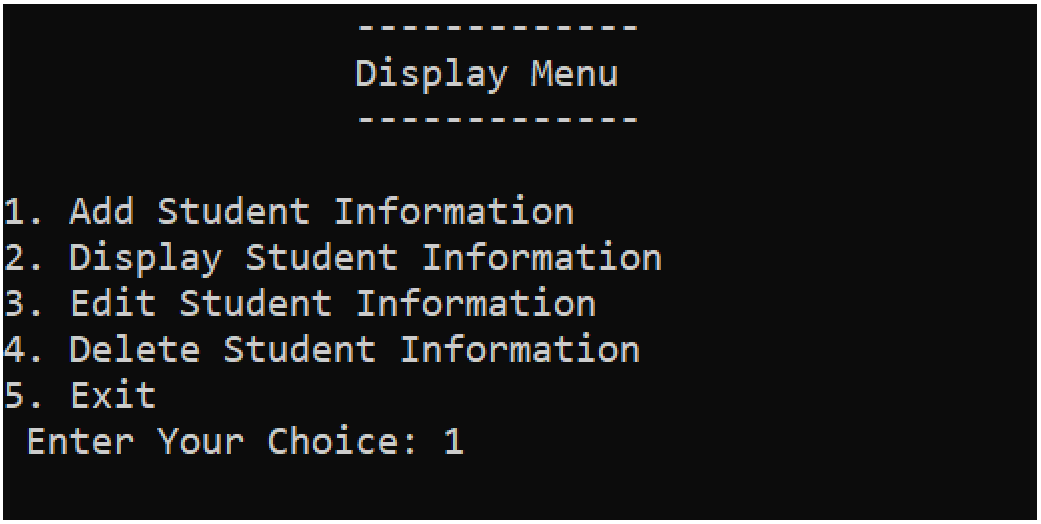
```
	string student;
	cout << "Enter Enroll Num To Delete Student's Data: ";
	cin.ignore();
	getline(cin, student);
	for (int i = 0; i < noOfStudents; i++)
	{
		if (std[i].Enrollment == student)
		{
			std[i].Enrollment = "0000"; // 0000 indicates that the student is deleted it over
rides the previous enrollment
			cout << std[i].Name << "Deleted Successfully!" << endl;
			--noOfStudents;
			system("pause>nul");
		}
	}
}
int main()
{
	int totalNoOfStudents;
	int noOfStudents = 0;
	Student* std;
	int choice = 0;
	int count = 0;
	void (*displayfunctionptr)() = Menu;
	void (*funcptrarr[])(Student*, int&) = {
	addingStudent,displayStudentData,editStudentData,deleteStudentData };
	cout << "Enter Number of Students You Want To Enter: ";
	cin >> totalNoOfStudents;
	int stdNo = 0;
	std = new Student[totalNoOfStudents];
	while (choice != 5)
	{
		system("CLS");
		displayfunctionptr();
		cout << " Enter Your Choice: ";
		cin >> choice;
		if (choice == 1)
		{
			system("CLS");
			cout << "\t\tTotal Num of Students That Can Be Added: " <<
				totalNoOfStudents << endl;
			if (noOfStudents != totalNoOfStudents)
			{
				stdNo++;
				cout << "\n\t\tEnter Data of Student No." << stdNo << endl;
				(*funcptrarr[0])(std, noOfStudents);
			}
			else
			{
				cout << "\n\t\tCan't Insert Any More Data (No Storage Left)!" << endl;
				system("pause>nul");
			}
		}
		else if (choice == 2)
		{
			system("CLS");
			if (noOfStudents != 0)
			{
				(*funcptrarr[1])(std, noOfStudents);
			}
			else
			{
				cout << "Empty Student Data...!!" << endl;
				system("pause>nul");
```

```
            }
        }
        else if (choice == 3)
        {
            system("CLS");
            if (noOfStudents != 0)
            {
                (*funcptrarr[2])(std, noOfStudents);
            }
            else
            {
                cout << "Empty Student Data...!!" << endl;
                system("pause>nul");
            }
        }
        else if (choice == 4)
        {
            system("CLS");
            if (noOfStudents != 0)
            {
                cout << "\t\tNo of Students Present in the DB are: " <<
                    count << endl;
                system("pause>nul");
                (*funcptrarr[3])(std, noOfStudents);
                count--;
                cout << "Now The No of Students Present in the DB Are: " << count <<
                    endl;
                system("pause");
            }
            else
            {
                cout << "Empty Student Data...!!" << endl;
                system("pause>nul");
            }
        }
        else if (choice == 5)
        {
            system("CLS");
            cout << "Deallocating Memory....!!" << endl;
            delete[] std;
        }
        else
        {
            system("CLS");
            cout << "Invalid Input" << endl;
        }
    }
    return 0;
}
```

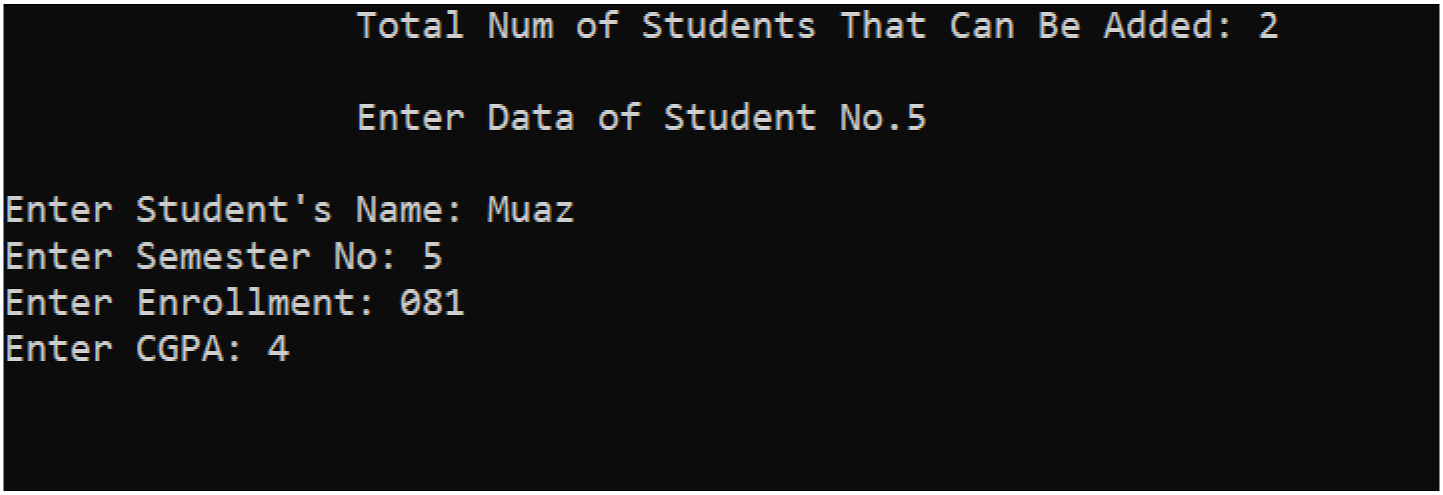
Output



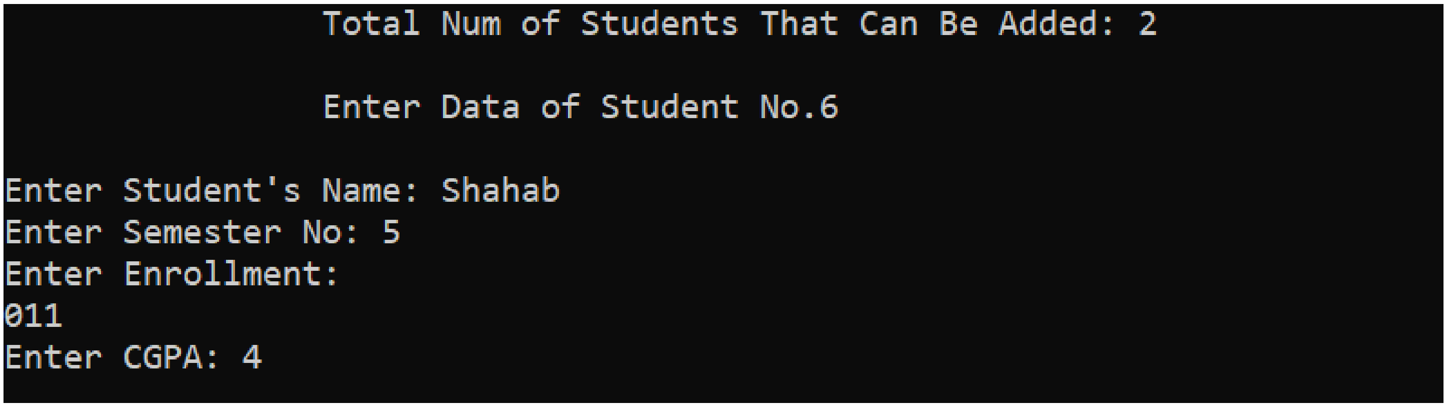
```
Enter Number of Students You Want To Enter: 2
```



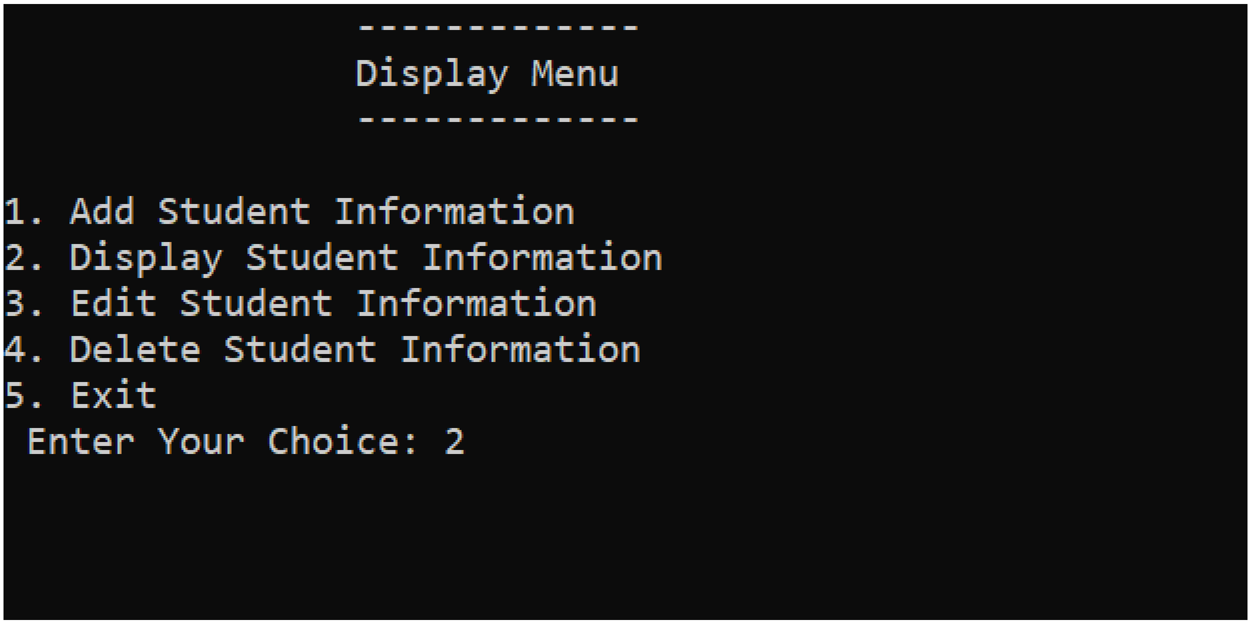
```
                -------------
                Display Menu
                -------------

1. Add Student Information
2. Display Student Information
3. Edit Student Information
4. Delete Student Information
5. Exit
 Enter Your Choice: 1
```



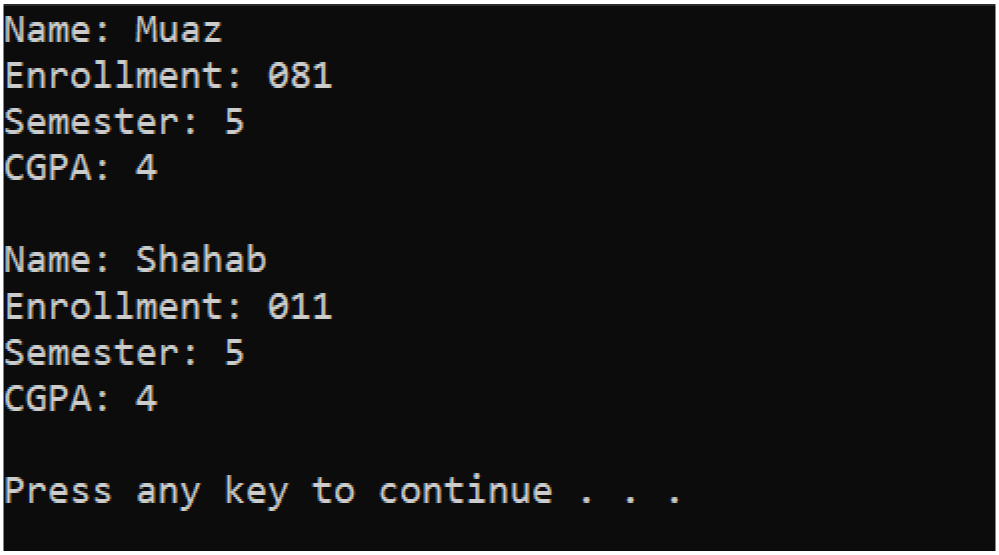
```
                Total Num of Students That Can Be Added: 2

                Enter Data of Student No.5

Enter Student's Name: Muaz
Enter Semester No: 5
Enter Enrollment: 081
Enter CGPA: 4
```



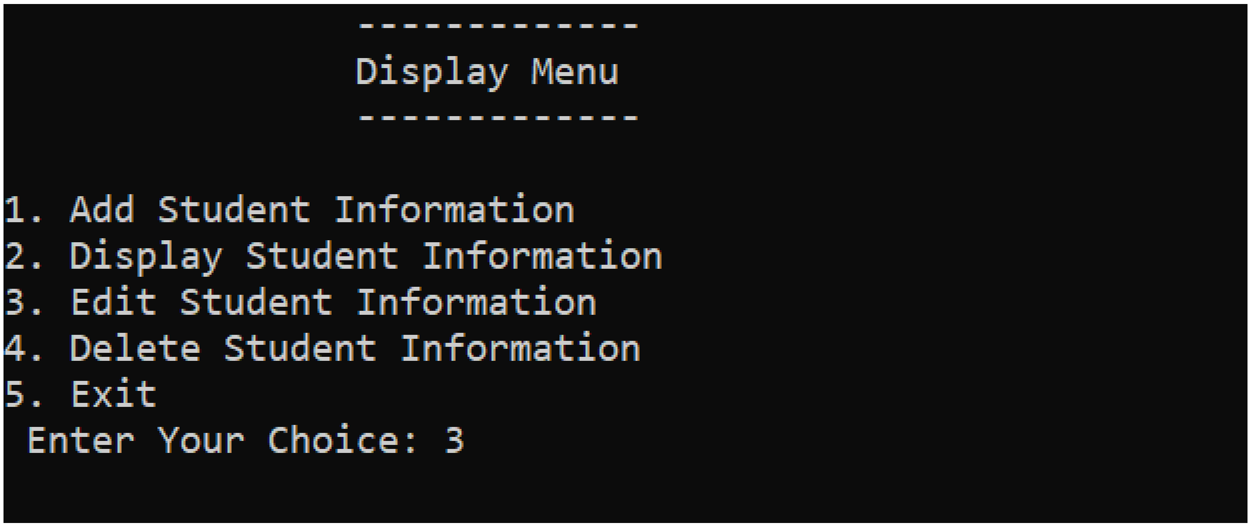
```
                Total Num of Students That Can Be Added: 2

                Enter Data of Student No.6

Enter Student's Name: Shahab
Enter Semester No: 5
Enter Enrollment:
011
Enter CGPA: 4
```



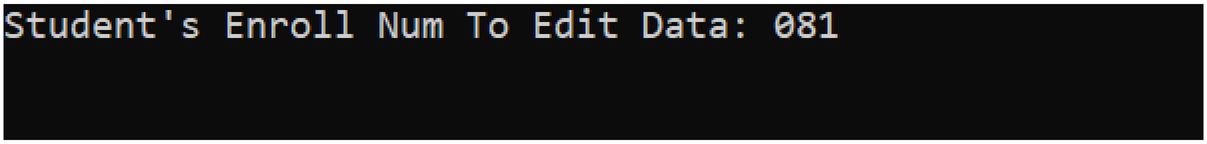
```
                -------------
                Display Menu
                -------------

1. Add Student Information
2. Display Student Information
3. Edit Student Information
4. Delete Student Information
5. Exit
 Enter Your Choice: 2
```



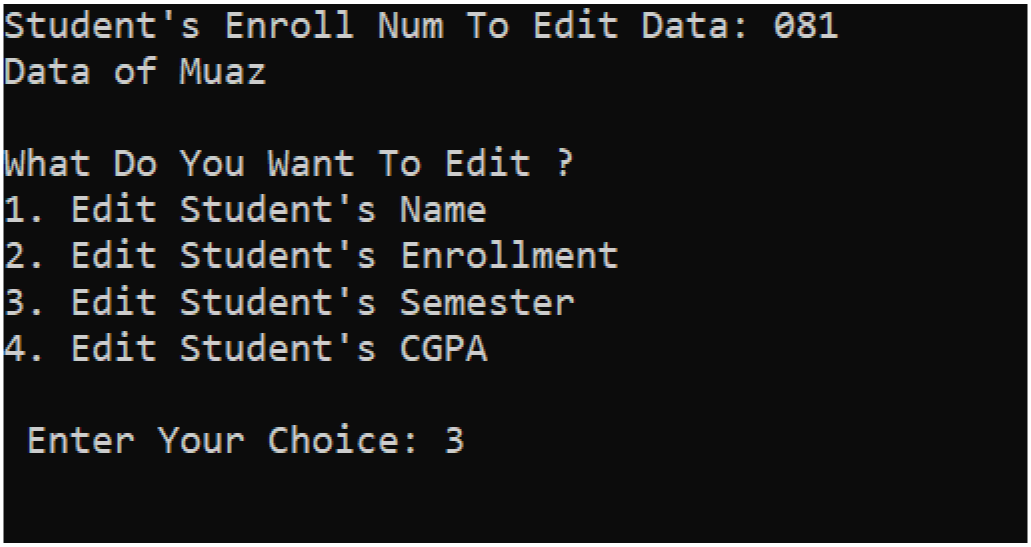
```
Name: Muaz
Enrollment: 081
Semester: 5
CGPA: 4

Name: Shahab
Enrollment: 011
Semester: 5
CGPA: 4

Press any key to continue . . .
```



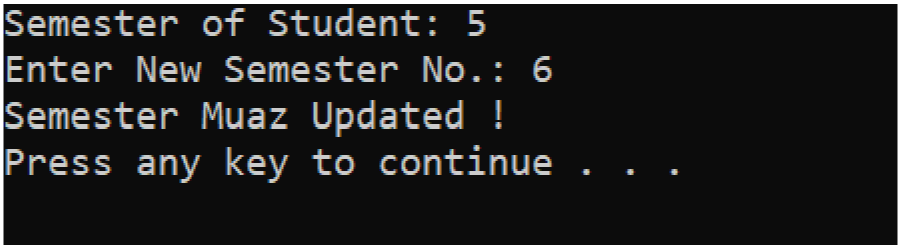
```
                -------------
                Display Menu
                -------------

1. Add Student Information
2. Display Student Information
3. Edit Student Information
4. Delete Student Information
5. Exit
 Enter Your Choice: 3
```



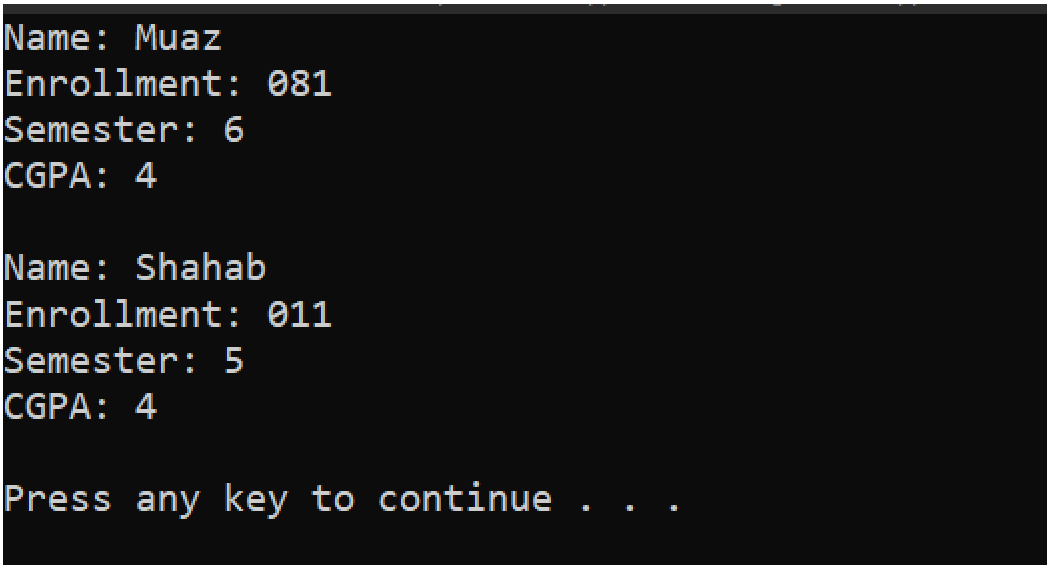
```
Student's Enroll Num To Edit Data: 081
```



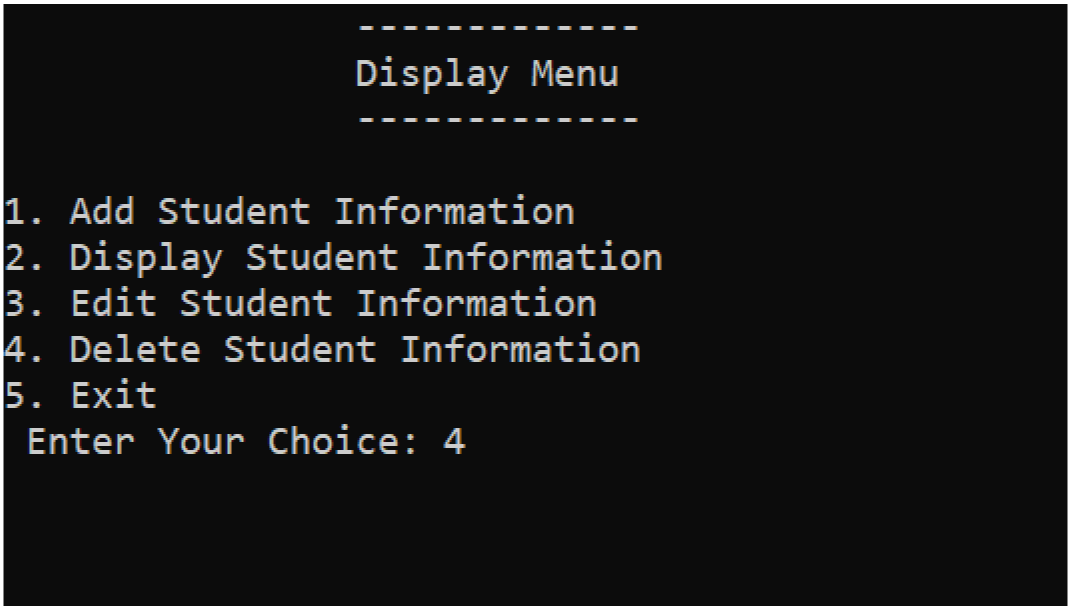
```
Student's Enroll Num To Edit Data: 081
Data of Muaz

What Do You Want To Edit ?
1. Edit Student's Name
2. Edit Student's Enrollment
3. Edit Student's Semester
4. Edit Student's CGPA

 Enter Your Choice: 3
```



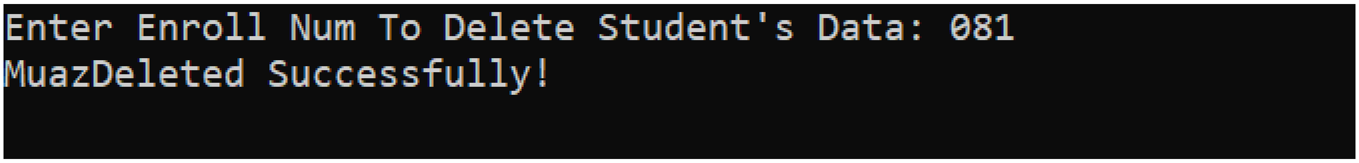
```
Semester of Student: 5
Enter New Semester No.: 6
Semester Muaz Updated !
Press any key to continue . . .
```



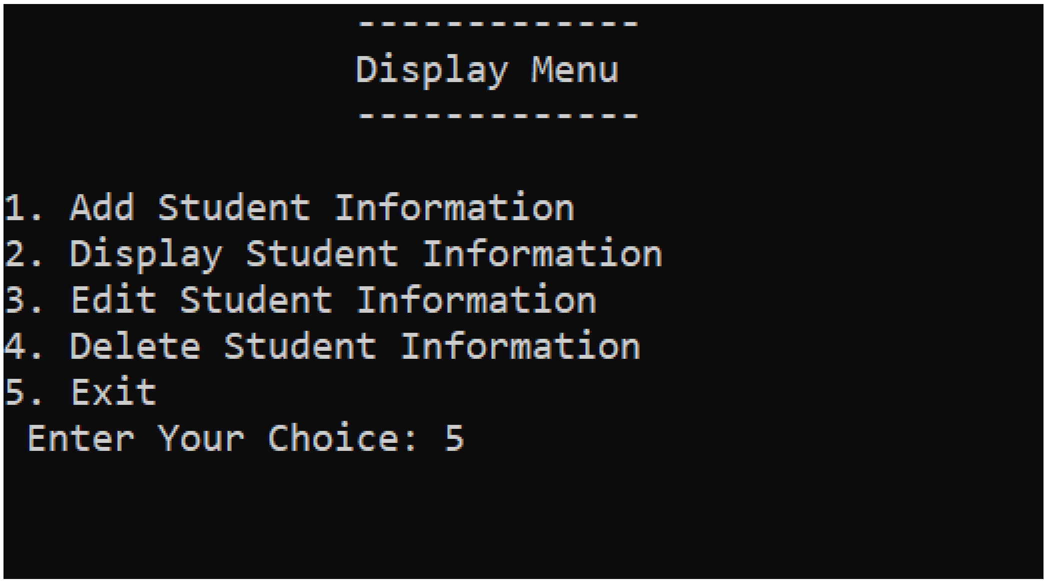
```
Name: Muaz
Enrollment: 081
Semester: 6
CGPA: 4

Name: Shahab
Enrollment: 011
Semester: 5
CGPA: 4

Press any key to continue . . .
```



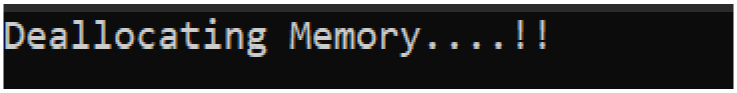
```
                -------------
                Display Menu
                -------------

1. Add Student Information
2. Display Student Information
3. Edit Student Information
4. Delete Student Information
5. Exit
 Enter Your Choice: 4
```



```
Enter Enroll Num To Delete Student's Data: 081
MuazDeleted Successfully!
```



```
                -------------
                Display Menu
                -------------

1. Add Student Information
2. Display Student Information
3. Edit Student Information
4. Delete Student Information
5. Exit
 Enter Your Choice: 5
```



```
Deallocating Memory....!!
```