"Question 01:" Develop a Student Management System that stores and processes student records using structures, functions, and arrays? The source code is available below?

1. Use An Array Of Structures To Store Multiple Student Records.

2. Use Functions To Modularize The Program.

3. The System Should Handle A Maximum Of 50 Students.

4. Marks Should Be Stored In An Array Inside The Student Structure (Each Student Has Marks For 5 Subjects).

5. The Program Should Include Menu-Driven Functionality Allowing The User To Perform Multiple Operations.

6. System Should Allow To Add Students, Remove Students, View Students And Calculate Avg, Highest And Lowest.

"SOLUTION:"

System Overview:

The system utilizes an array of structures to store student records, with each structure containing student information and an array of marks for 5 subjects. Functions are employed to modularize the program, ensuring efficient management of student data.

Key Features:

1. Stores up to 50 student records
2. Allows adding, removing, and viewing student records
3. Calculates average, highest, and lowest marks for each student

Implementation:

The system is implemented using C++, structures, functions, and arrays to manage student data efficiently.

```cpp
#include <iostream>
#include <string>
using namespace std;

struct Student {
    string name;
    int id;
    int marks[5];
    float average;
};

const int MAX_STUDENTS = 50;
Student students[MAX_STUDENTS];
int totalStudents = 0;


void addStudent() {
    if (totalStudents >= MAX_STUDENTS) {
        cout << "Maximum students "<<endl;
        return;
    }

    cout << "\n student name"<<endl;
    cin.ignore();
    getline(cin, students[totalStudents].name);

    cout << " student ID "<<endl;
    cin >> students[totalStudents].id;

    cout << " marks for 5 subjects"<<endl;
    int sum = 0;
    for (int i = 0; i < 5; i++) {
        cout << "Subject "<<endl;
        cout << i + 1<<endl;
        cin >> students[totalStudents].marks[i];
        sum += students[totalStudents].marks[i];
    }
    students[totalStudents].average = sum / 5.0;

    totalStudents++;
    cout << "Student added successfully"<<endl;
}
```

```
void displayStudents() {
    if (totalStudents == 0) {
        cout << "No students"<<endl;
        return;
    }

    cout << " Student Records "<<endl;
    for (int i = 0; i < totalStudents; i++) {
        cout << "Student " << i + 1 << endl;
        cout << "Name: " << students[i].name << endl;
        cout << "ID: " << students[i].id << endl;
        cout << "Marks: ";
        for (int j = 0; j < 5; j++) {
            cout << students[i].marks[j] << " ";
        }
        cout << "\nAverage Marks" << students[i].average << endl;
    }
}

void removeStudent() {
    if (totalStudents == 0) {
        cout << "No students to remove"<<endl;
        return;
    }

    int id;
    cout << "student ID to remove"<<endl;
    cin >> id;

    bool found = false;
    for (int i = 0; i < totalStudents; i++) {
        if (students[i].id == id) {
            found = true;

            for (int j = i; j < totalStudents - 1; j++) {
                students[j] = students[j + 1];
            }
            totalStudents--;
            cout << "Student removed successfully"<<endl;
            break;
        }
    }

    if (!found) {
        cout << "Student ID not found"<<endl;
    }
```

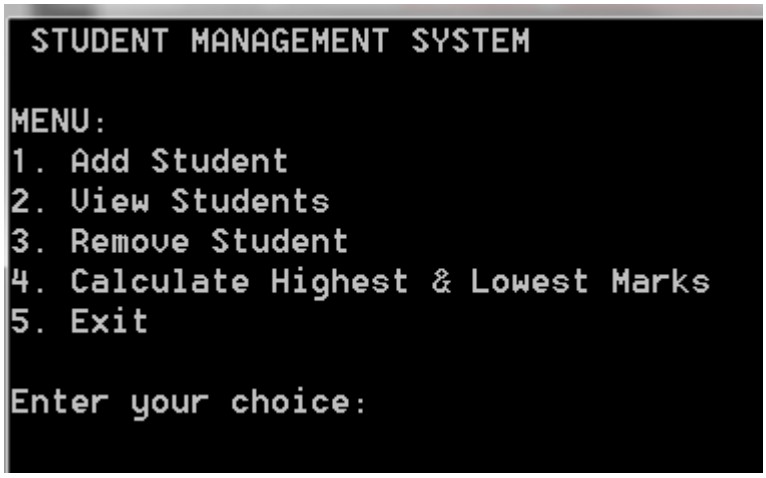
```
}

void calculateMarks() {
    if (totalStudents == 0) {
        cout << "No calculate marks"<<endl;
        return;
    }

    float highest = students[0].average;
    float lowest = students[0].average;

    for (int i = 1; i < totalStudents; i++) {
        if (students[i].average > highest)
            highest = students[i].average;
        if (students[i].average < lowest)
            lowest = students[i].average;
    }

    cout << "Highest Average  " << highest << endl;
    cout << "Lowest Average " << lowest << endl;
}


int main() {
    cout <<  " STUDENT MANAGEMENT SYSTEM\n";

    int choice;
    do {
        cout << "\nMENU:"<<endl;
        cout << "1. Add Student"<<endl;
        cout << "2. View Students"<<endl;
        cout << "3. Remove Student"<<endl;
        cout << "4. Calculate Highest & Lowest Marks"<<endl;
        cout << "5. Exit"<<endl;
        cout<<endl;
        cout << "Enter your choice: ";
        cin >> choice;
```

```
}

void calculateMarks() {
    if (totalStudents == 0) {
        cout << "No calculate marks"<<endl;
        return;
    }

    float highest = students[0].average;
    float lowest = students[0].average;

    for (int i = 1; i < totalStudents; i++) {
        if (students[i].average > highest)
            highest = students[i].average;
        if (students[i].average < lowest)
            lowest = students[i].average;
    }

    cout << "Highest Average  " << highest << endl;
    cout << "Lowest Average " << lowest << endl;
}


int main() {
    cout <<  " STUDENT MANAGEMENT SYSTEM\n";

    int choice;
    do {
        cout << "\nMENU:"<<endl;
        cout << "1. Add Student"<<endl;
        cout << "2. View Students"<<endl;
        cout << "3. Remove Student"<<endl;
        cout << "4. Calculate Highest & Lowest Marks"<<endl;
        cout << "5. Exit"<<endl;
        cout<<endl;
        cout << "Enter your choice: ";
        cin >> choice;
```

```
switch (choice) {
    case 1:
        addStudent();
            break;
    case 2:
        displayStudents();
            break;
    case 3:
        removeStudent();
            break;
```

```
            case 4:
                  calculateMarks();
                        break;
            case 5:
                  cout << "Exiting program"<<endl;
                              break;
            default:
                        cout << "Invalid choice"<<endl;
        }
    }
    while (choice != 5);

    return 0;
}
```

“OUTPUT:”



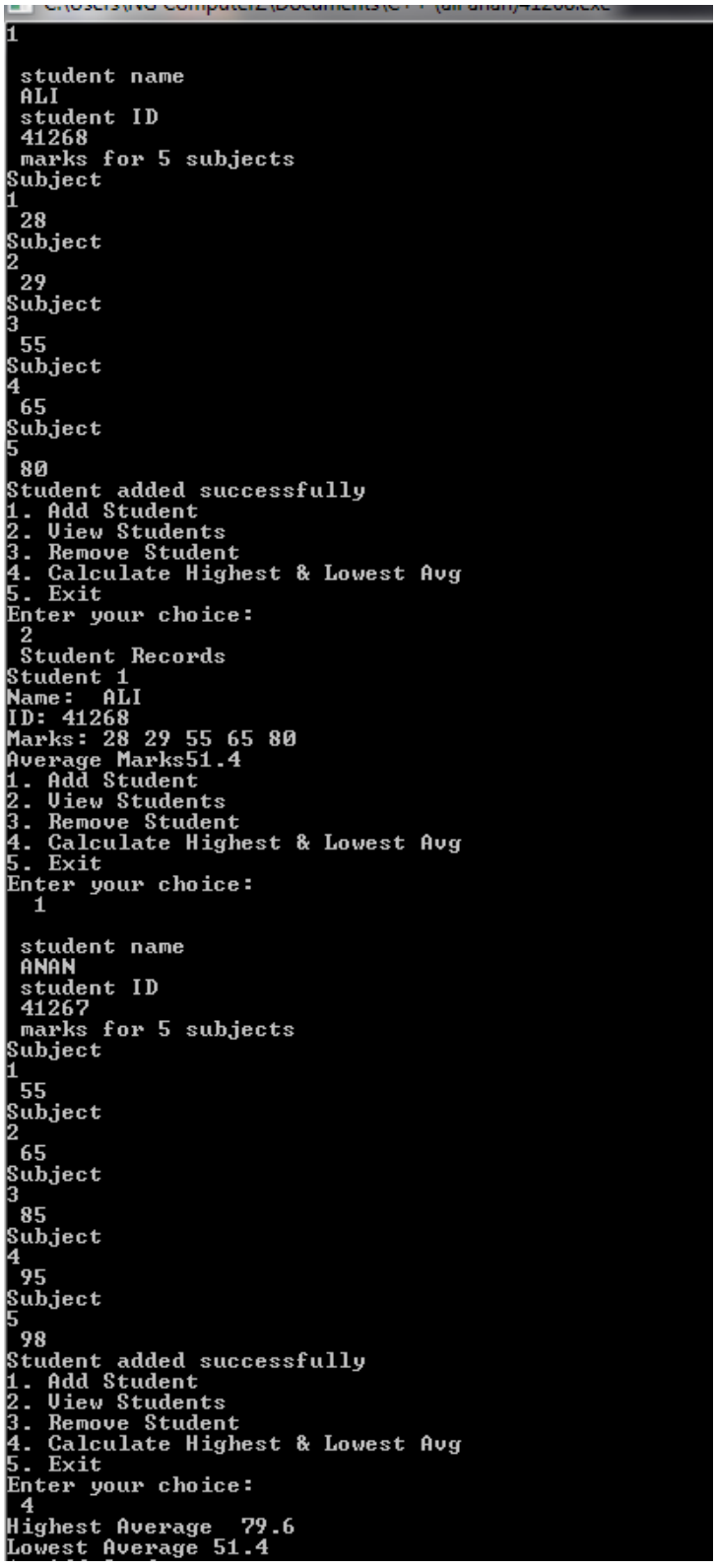
1.ADD STUDENT:

```
1. Add Student
2. View Students
3. Remove Student
4. Calculate Highest & Lowest Avg
5. Exit
Enter your choice:
3
student ID to remove
41266
Student ID not found
1. Add Student
2. View Students
3. Remove Student
4. Calculate Highest & Lowest Avg
5. Exit
Enter your choice:
3
student ID to remove
41268
Student removed successfully
```

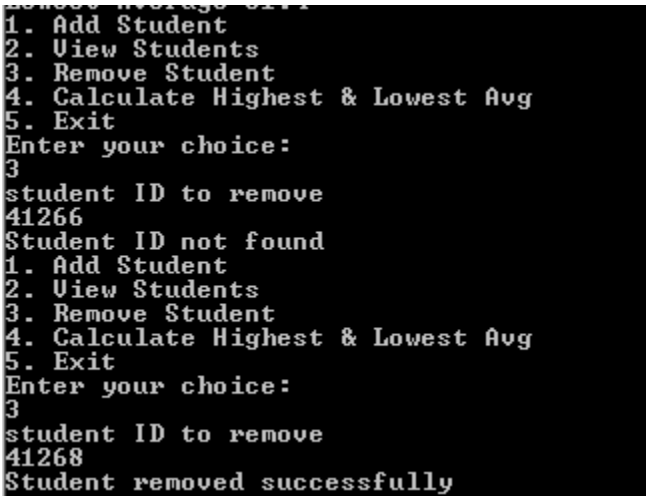
2.VIEW STUDENT:

```
1. Add Student
2. View Students
3. Remove Student
4. Calculate Highest & Lowest Avg
5. Exit
Enter your choice:
 2
 Student Records
Student 1
Name:  ALI
ID: 41268
Marks: 28 29 55 65 80
Average Marks51.4
```

4.CAL(HIGH OR LOW AVG):



3.REMOVE STUDENT:



Source Code of student mangment system

```
 #include <iostream>
#include <string>
using namespace std;

struct Student {
   string name;
   int id;
   int marks[5];
   float average;
};

const int MAX_STUDENTS = 50;
```

```
Student students[MAX_STUDENTS];
int totalStudents = 0;



void addStudent() {
   if (totalStudents >= MAX_STUDENTS) {
      cout << "Max students "<<endl;
      return;
   }

   cout << "\n student name"<<endl;
   cin.ignore();
   getline(cin, students[totalStudents].name);

   cout << " student ID "<<endl;
   cin >> students[totalStudents].id;

   cout << " marks for 5 subjects"<<endl;
   int sum = 0;
   for (int i = 0; i < 5; i++) {
```

```
        cout << "Subject " << i + 1<<endl;
        cin >> students[totalStudents].marks[i];
        sum += students[totalStudents].marks[i];
    }
    students[totalStudents].average = sum / 5.0;

    totalStudents++;
    cout << "Student added successfully"<<endl;
}

void displayStudents() {
    if (totalStudents == 0) {
        cout << "No students"<<endl;
        return;
    }

    cout << " Student Records "<<endl;
    for (int i = 0; i < totalStudents; i++) {
        cout << "Student " << i + 1 << endl;
        cout << "Name: " << students[i].name << endl;
```

```
        cout << "ID: " << students[i].id << endl;
        cout << "Marks: ";
        for (int j = 0; j < 5; j++) {
            cout << students[i].marks[j] << " ";
        }
        cout << "\nAverage Marks" << students[i].average << endl;
    }
}

void removeStudent() {
    if (totalStudents == 0) {
        cout << "No students to remove"<<endl;
        return;
    }

    int id;
    cout << "Enter student ID to remove"<<endl;
    cin >> id;

    bool found = false;
```

```
    for (int i = 0; i < totalStudents; i++) {
        if (students[i].id == id) {
            found = true;

            for (int j = i; j < totalStudents - 1; j++) {
                students[j] = students[j + 1];
            }
            totalStudents--;
            cout << "Student removed successfully"<<endl;
            break;
        }
    }

    if (!found) {
        cout << "Student ID not found"<<endl;
    }
}

void calculateMarks() {
    if (totalStudents == 0) {
```

```
        cout << "No calculate marks"<<endl;
        return;
    }

    float highest = students[0].average;
    float lowest = students[0].average;

    for (int i = 1; i < totalStudents; i++) {
        if (students[i].average > highest)
            highest = students[i].average;
        if (students[i].average < lowest)
            lowest = students[i].average;
    }

    cout << "Highest Average  " << highest << endl;
    cout << "Lowest Average " << lowest << endl;
}


int main() {
```

```
cout <<  " STUDENT MANAGEMENT SYSTEM\n";


int choice;
do {
   cout << "\nMENU:"<<endl;
   cout << "1. Add Student"<<endl;
   cout << "2. View Students"<<endl;
   cout << "3. Remove Student"<<endl;
   cout << "4. Calculate Highest & Lowest Marks"<<endl;
   cout << "5. Exit"<<endl;
   cout<<endl;
   cout << "Enter your choice: ";
   cin >> choice;

   switch (choice) {
      case 1:
          addStudent();
            break;
      case 2:
           displayStudents();
```

```
            break;
        case 3:
            removeStudent();
            break;
        case 4:
            calculateMarks();
            break;
        case 5:
            cout << "Exiting program"<<endl;
                    break;
        default:
            cout << "Invalid choice, try again"<<endl;
      }
   } while (choice != 5);

   return 0;
}
```