Assignment 01

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

//Declaration of Structure.
struct studentInfo
{
    string name;
    int roll;
    float sgpa;
};

// User Input
void getStudentInfo(struct studentInfo s[], int n)
{
    for (int i = 0; i < n; i++)
    {
        cout << "Student " << (i + 1) << endl;
        cout << "Enter Name: ";
        cin >> s[i].name;
        cout << "Enter Roll No: ";
        cin >> s[i].roll;
        cout << "Enter SGPA: ";
        cin >> s[i].sgpa;
        cout << endl;
    }
}


// Display Output
void displayInfo(struct studentInfo s[], int n)
{
    for (int i = 0; i < n; i++)
    {
        cout << "Name: " << s[i].name << endl;
        cout << "Roll No: " << s[i].roll << endl;
        cout << "SGPA: " << s[i].sgpa << endl;
        cout << endl;
    }
}


// Sorting Students w.r.t. Roll Numbers using Bubble Sort.
void bubbleSort(struct studentInfo s[], int n)
{
    for (int i = 0; i < (n - 1); i++)
    {
        for (int j = 0; j < (n - i - 1); j++)
        {
            if (s[j].roll > s[j + 1].roll)
            {
                swap(s[j].roll, s[j + 1].roll);
                swap(s[j].name, s[j + 1].name);
                swap(s[j].sgpa, s[j + 1].sgpa);
            }
        }
    }
}


// Sorting Student w.r.t. SGPA using Bubble Sort.
void bubbleSort2(struct studentInfo s[], int n)
{
    for (int i = 0; i < (n - 1); i++)
    {
        for (int j = 0; j < (n - i - 1); j++)
        {
            if (s[j].sgpa < s[j + 1].sgpa)
            {
                swap(s[j].roll, s[j + 1].roll);
                swap(s[j].name, s[j + 1].name);
                swap(s[j].sgpa, s[j + 1].sgpa);
            }
        }
    }
    if (n >= 10)
    {
        cout << "List of Top 10 Students According to SGPA\n\n";
        for (int i = 0; i < 10; i++)
        {
            cout << "Rank : " << (i + 1) << endl;
            cout << "Name: " << s[i].name << endl;
            cout << "Roll No: " << s[i].roll << endl;
            cout << "SGPA: " << s[i].sgpa << endl;
            cout << endl;
        }
    }
    else
    {
        cout << "List of Top Students According to SGPA\n\n";
        for (int i = 0; i < n; i++)
        {
            cout << "Rank : " << (i + 1) << endl;
            cout << "Name: " << s[i].name << endl;
            cout << "Roll No: " << s[i].roll << endl;
            cout << "SGPA: " << s[i].sgpa << endl;
            cout << endl;
        }
    }
}

// Sorting Students Alphabetically.
void sortingAlphabetically(struct studentInfo s[], int n)
{
    for (int i = 0; i < (n - 1); i++)
    {
        for (int j = 0; j < (n - i - 1); j++)
        {
            if (s[j].name > s[j + 1].name)
            {
                swap(s[j].roll, s[j + 1].roll);
                swap(s[j].name, s[j + 1].name);
                swap(s[j].sgpa, s[j + 1].sgpa);
            }
        }
    }
}

// Searching Student w.r.t. SGPA using Linear Search.
void searchStudent(struct studentInfo s[], int n, float key)
{
    for (int i = 0; i < n; i++)
    {
        if (s[i].sgpa == key)
        {
            cout << "Student Info: \n"
                 << endl;
            cout << "Name: " << s[i].name << endl;
            cout << "Roll No: " << s[i].roll << endl;
            cout << "SGPA: " << s[i].sgpa << endl;
            cout << endl;
        }
    }
}

// Searching Student w.r.t. Name using Binary Search.
int binarySearch(struct studentInfo s[], int n, string key)
{
    int start = 0;
    int end = n - 1;
    while (start <= end)
    {
        int mid = (start + end) / 2;
        if (s[mid].name == key)
        {
            return mid;
        }
        else if (s[mid].name < key)
        {
            start = mid + 1;
        }
        else
        {
            end = mid - 1;
        }
    }
}

int main()
{
    cout << "Student Information.\n"
         << endl;
    int n;
```

```
cout << "Enter Number of Students: ";
cin >> n;

struct studentInfo s[n];

getStudentInfo(s, n);

while (true)
{
    // Menu of the Program.
    cout << "Enter" << endl;
    cout << "1.Arrange List of Student According to Roll Numbers." << endl;
    cout << "2.Arrange List of Student Alphabetically." << endl;
    cout << "3.Arrange List of Student to Find Out First 10 Toppers of Class." << endl;
    cout << "4.Search Student According to SGPA." << endl;
    cout << "5.Search Student According to Name." << endl;
    cout << "6.Exit" << endl;
    cout << "\n";

    int ope;
    cout << "Enter Your Choice: ";
    cin >> ope;
    cout << "\n";

    // List of Student According to Roll Numbers.
    if (ope == 1)
    {
        cout << "list of The Students According to Roll Numbers.\n\n";
        bubbleSort(s, n);
        displayInfo(s, n);
    }
    // List of Student According to their Name.
    else if (ope == 2)
    {
        cout << "List of Top Students According to Their Name\n\n";
        sortingAlphabetically(s, n);
        displayInfo(s, n);
    }
    // List of Student to Find Out First 10 Toppers of Class.
    else if (ope == 3)
    {
        bubbleSort2(s, n);
    }
    //Search Student According to SGPA.
    else if (ope == 4)
    {
        cout << "Searching Student with SGPA.\n";
        float key;
        cout << "Enter SGPA: ";
        cin >> key;
        cout << "\n";
        searchStudent(s, n, key);
    }
    //Search Student According to Name.
    else if (ope == 5)
```

```
        {
            sortingAlphabetically(s, n);
            cout << "Searching Student with Name.\n";
            string key2;
            cout << "Enter Name: ";
            cin >> key2;
            cout << "\n";

            int index = binarySearch(s, n, key2);
            cout << "Student Info: \n"
                 << endl;
            cout << "Name: " << s[index].name << endl;
            cout << "Roll No: " << s[index].roll << endl;
            cout << "SGPA: " << s[index].sgpa << endl;
            cout << endl;
        }
        // Exiting the Program.
        else if (ope == 6)
        {
            cout << "Exiting The Program...";
            break;
        }
        // Invalid Input Case.
        else
        {
            cout << "Enter Valid Input..\n";
        }
    }
    return 0;
}
```

Output :

```
Student Information.

Enter Number of Students: 10

Student 1
Enter Name: Parth
Enter Roll No: 67
Enter SGPA: 10

Student 2
Enter Name: Sahil
Enter Roll No: 46
Enter SGPA: 9

Student 3
Enter Name: Ayush
Enter Roll No: 51
Enter SGPA: 7

Student 4
Enter Name: Hitesh
Enter Roll No: 52
Enter SGPA: 9

Student 5
Enter Name: Aditya
Enter Roll No: 57
Enter SGPA: 10

Student 6
Enter Name: Arohi
Enter Roll No: 34
Enter SGPA: 6

Student 7
Enter Name: Sam
Enter Roll No: 60
Enter SGPA: 10

Student 8
Enter Name: Pranav
Enter Roll No: 9
Enter SGPA: 10

Student 9
Enter Name: Pratik
Enter Roll No: 45
Enter SGPA: 8

Student 10
Enter Name: Darpan
Enter Roll No: 48
Enter SGPA: 5

Enter
1.Arrange List of Student According to Roll Numbers.
2.Arrange List of Student Alphabetically.
3.Arrange List of Student to Find Out First 10 Toppers of Class.
4.Search Student According to SGPA.
5.Search Student According to Name.
6.Exit

Enter Your Choice: 1

list of The Students According to Roll Numbers.

Name: Pranav
Roll No: 9
SGPA: 10

Name: Arohi
Roll No: 34
SGPA: 6

Name: Pratik
Roll No: 45
```

```
SGPA: 8

Name: Sahil
Roll No: 46
SGPA: 9

Name: Darpan
Roll No: 48
SGPA: 5

Name: Ayush
Roll No: 51
SGPA: 7

Name: Hitesh
Roll No: 52
SGPA: 9

Name: Aditya
Roll No: 57
SGPA: 10

Name: Sam
Roll No: 60
SGPA: 10

Name: Parth
Roll No: 67
SGPA: 10

Enter
1.Arrange List of Student According to Roll Numbers.
2.Arrange List of Student Alphabetically.
3.Arrange List of Student to Find Out First 10 Toppers of Class.
4.Search Student According to SGPA.
5.Search Student According to Name.
6.Exit

Enter Your Choice: 2

List of Top Students According to Their Name

Name: Aditya
Roll No: 57
SGPA: 10

Name: Arohi
Roll No: 34
SGPA: 6

Name: Ayush
Roll No: 51
SGPA: 7

Name: Darpan
```

```
Roll No: 48
SGPA: 5

Name: Hitesh
Roll No: 52
SGPA: 9

Name: Parth
Roll No: 67
SGPA: 10

Name: Pranav
Roll No: 9
SGPA: 10

Name: Pratik
Roll No: 45
SGPA: 8

Name: Sahil
Roll No: 46
SGPA: 9

Name: Sam
Roll No: 60
SGPA: 10

Enter
1.Arrange List of Student According to Roll Numbers.
2.Arrange List of Student Alphabetically.
3.Arrange List of Student to Find Out First 10 Toppers of Class.
4.Search Student According to SGPA.
5.Search Student According to Name.
6.Exit

Enter Your Choice: 3

List of Top 10 Students According to SGPA

Rank : 1
Name: Aditya
Roll No: 57
SGPA: 10

Rank : 2
Name: Parth
Roll No: 67
SGPA: 10

Rank : 3
Name: Pranav
Roll No: 9
SGPA: 10

Rank : 4
```

```
Name: Sam
Roll No: 60
SGPA: 10

Rank : 5
Name: Hitesh
Roll No: 52
SGPA: 9

Rank : 6
Name: Sahil
Roll No: 46
SGPA: 9

Rank : 7
Name: Pratik
Roll No: 45
SGPA: 8

Rank : 8
Name: Ayush
Roll No: 51
SGPA: 7

Rank : 9
Name: Arohi
Roll No: 34
SGPA: 6

Rank : 10
Name: Darpan
Roll No: 48
SGPA: 5

Enter
1.Arrange List of Student According to Roll Numbers.
2.Arrange List of Student Alphabetically.
3.Arrange List of Student to Find Out First 10 Toppers of Class.
4.Search Student According to SGPA.
5.Search Student According to Name.
6.Exit

Enter Your Choice: 4

Searching Student with SGPA.
Enter SGPA: 6

Student Info:

Name: Arohi
Roll No: 34
SGPA: 6

Enter
1.Arrange List of Student According to Roll Numbers.
```

```
2.Arrange List of Student Alphabetically.
3.Arrange List of Student to Find Out First 10 Toppers of Class.
4.Search Student According to SGPA.
5.Search Student According to Name.
6.Exit

Enter Your Choice: 5

Searching Student with Name.
Enter Name: Parth

Student Info:

Name: Parth
Roll No: 67
SGPA: 10

Enter
1.Arrange List of Student According to Roll Numbers.
2.Arrange List of Student Alphabetically.
3.Arrange List of Student to Find Out First 10 Toppers of Class.
4.Search Student According to SGPA.
5.Search Student According to Name.
6.Exit

Enter Your Choice: 6

Exiting The Program..
```