```
#include<iostream>
#include <string.h>
#include <ctype.h>
#define max 100
using namespace std;

// Data structure to store student information
class data{
    public:
    int roll_no;
    string name;
    double sgpa;
    string department;
};

data arr[max]; // Array to hold student data
int n = 0; // Number of students entered

void ask();
void display();

void enter(){
    // Input the details for each student
    cout << "Total number of students you want to enter = ";
    cin >> n;
    for(int i = 0; i < n; i++){
        cout << "\nEnter Details of student - " << i+1 << endl;
        cout << "Roll no = ";
        cin >> arr[i].roll_no;
        cout << "Name = ";
```

```
        cin >> arr[i].name;
        cout << "SGPA = ";
        cin >> arr[i].sgpa;

        // Assign department based on user choice
        cout << "Department:\n1. Computer\n2. Software\n3. ENTC\n4. Mechanical\n5. Design\nEnter choice = ";
        int ch;
        cin >> ch;
        switch(ch){
            case 1: arr[i].department = "Computer"; break;
            case 2: arr[i].department = "Software"; break;
            case 3: arr[i].department = "ENTC"; break;
            case 4: arr[i].department = "Mechanical"; break;
            case 5: arr[i].department = "Design"; break;
            default: arr[i].department = "Computer"; cout << "Invalid choice! Defaulting to Computer department...\n";
        }
        cout << "Student " << i+1 << " details saved successfully!\n";
    }
    ask();
}

void display(){
    // Display details for each student
    for(int i = 0; i < n; i++){
        cout << "\nRoll no: " << arr[i].roll_no << endl;
        cout << "Name: " << arr[i].name << endl;
        cout << "SGPA: " << arr[i].sgpa << endl;
        cout << "Department: " << arr[i].department << endl;
    }
```

```
    ask();
}

// Bubble Sort by Roll Number
void Bub_sort(){
    for(int i = 0; i < n - 1; i++){
        for(int j = 0; j < n - 1 - i; j++){
            if(arr[j].roll_no > arr[j+1].roll_no){
                // Swap if current roll number is greater than the next
                data temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
            }
        }
    }
    display();
}

// Insertion Sort by Name
void Ins_sort(){
    for(int i = 1; i < n; i++){
        data temp = arr[i];
        int j = i - 1;
        while(j >= 0 && arr[j].name > temp.name){
            arr[j + 1] = arr[j];
            j--;
        }
        arr[j + 1] = temp;
    }
}
```

```
// Quick Sort by SGPA (for top 10)
int partition(int low, int high){
    double pivot = arr[high].sgpa;
    int i = low - 1;
    for(int j = low; j < high; j++){
        if(arr[j].sgpa >= pivot) { // Sort in descending order
            i++;
            swap(arr[i], arr[j]);
        }
    }
    swap(arr[i + 1], arr[high]);
    return (i + 1);
}

void Quick_sort(int low, int high){
    if(low < high){
        int pi = partition(low, high);
        Quick_sort(low, pi - 1);
        Quick_sort(pi + 1, high);
    }
}

void display_first_10(){
    // Display top 10 students by SGPA
    cout << "First 10 toppers are: \n";
    if(n <= 10){
        display();
    } else {
        for(int i = 0; i < 10; i++){
```

```
            cout << "\nRoll no: " << arr[i].roll_no << endl;
            cout << "Name: " << arr[i].name << endl;
            cout << "SGPA: " << arr[i].sgpa << endl;
            cout << "Department: " << arr[i].department << endl;
        }
        ask();
    }
}

// Helper to display student details by index
void disp_search(int i){
    cout << "\nRoll no: " << arr[i].roll_no << endl;
    cout << "Name: " << arr[i].name << endl;
    cout << "SGPA: " << arr[i].sgpa << endl;
    cout << "Department: " << arr[i].department << endl;
}

// Linear Search by SGPA
void Lin_search(){
    double search_sgpa;
    cout << "Enter SGPA to search: ";
    cin >> search_sgpa;
    bool found = false;
    for(int i = 0; i < n; i++){
        if(arr[i].sgpa == search_sgpa){
            disp_search(i);
            found = true;
        }
    }
    if(!found) cout << "\nNo student scored that SGPA!\n";
```

```
    ask();
}

// Binary Search by Name (after sorting by name)
void Bin_search(){
    Ins_sort(); // Ensure the array is sorted by name for binary search
    char search_name[100];
    cout << "Enter Name to search: ";
    cin >> search_name;

    // Convert search_name to lowercase for case-insensitive search
    strlwr(search_name);

    int low = 0, high = n - 1;
    bool found = false;

    while (low <= high){
        int mid = (low + high) / 2;

        char student_name[100];
        strcpy(student_name, arr[mid].name.c_str());
        strlwr(student_name);

        if(strcmp(student_name, search_name) == 0){
            disp_search(mid);
            found = true;
            break;
        }
        else if(strcmp(student_name, search_name) < 0){
            low = mid + 1;
```

```
        }
        else{
            high = mid - 1;
        }
    }
    if(!found) cout << "\nNo student found with that name!\n";
    ask();
}

// Sort array by name for further use
void sortByName(){
    for(int i = 0; i < n - 1; i++){
        for(int j = i + 1; j < n; j++){
            if(arr[i].name > arr[j].name){
                data temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
}

// Helper function to convert a char array to lowercase
void toLowerStr(char str[]){
    for (int i = 0; str[i]; i++){
        str[i] = tolower(str[i]);
    }
}

// Fibonacci Search to check if a student belongs to the Computer department
```

```
void Fibo_search(){
    sortByName(); // Sort array by name for search
    char search_name[50];
    cout << "Enter the student's name to check if they belong to the Computer department: ";
    cin >> search_name;

    toLowerStr(search_name);

    int fibMMm2 = 0;
    int fibMMm1 = 1;
    int fibM = fibMMm2 + fibMMm1;

    while (fibM < n){
        fibMMm2 = fibMMm1;
        fibMMm1 = fibM;
        fibM = fibMMm2 + fibMMm1;
    }

    int offset = -1;
    bool found = false;

    while(fibM > 1){
        int i = min(offset + fibMMm2, n - 1);

        char name_lower[50];
        strcpy(name_lower, arr[i].name.c_str());
        toLowerStr(name_lower);

        if(strcmp(name_lower, search_name) < 0){
            fibM = fibMMm1;
```

```
            fibMMm1 = fibMMm2;
            fibMMm2 = fibM - fibMMm1;
            offset = i;
        }
        else if(strcmp(name_lower, search_name) > 0){
            fibM = fibMMm2;
            fibMMm1 -= fibMMm2;
            fibMMm2 = fibM - fibMMm1;
        }
        else{
            if(strcmp(arr[i].department.c_str(), "Computer") == 0){
                cout << "\nStudent " << arr[i].name << " belongs to the Computer department.\n";
            }
            else{
                cout << "\nStudent " << arr[i].name << " does not belong to the Computer
department.\n";
            }
            found = true;
            break;
        }
    }

    if(!found && fibMMm1 && offset + 1 < n){
        char last_name_lower[50];
        strcpy(last_name_lower, arr[offset + 1].name.c_str());
        toLowerStr(last_name_lower);

        if(strcmp(last_name_lower, search_name) == 0){
            if(strcmp(arr[offset + 1].department.c_str(), "Computer") == 0){
                cout << "\nStudent " << arr[offset + 1].name << " belongs to the Computer
department.\n";
```

```
            }
            else{
                cout << "\nStudent " << arr[offset + 1].name << " does not belong to the Computer department.\n";
            }
        }
        else{
            cout << "\nNo student found with that name.\n";
        }
    }

    ask();
}

void ask(){
    int ch = 0;
    cout << "\nWhat you want to do?\n";
    cout << "1. Display data\n";
    cout << "2. Sort - Roll no\n";
    cout << "3. Sort - Name\n";
    cout << "4. Sort by SGPA and Display Top 10\n";
    cout << "5. Search by SGPA\n";
    cout << "6. Search by Name\n";
    cout << "7. Search whether a student belongs to Computer department or not\n";
    cout << "8. Exit\n";
    cout << "Enter choice = ";
    cin >> ch;
    switch(ch){
        case 1: display(); break;
        case 2: Bub_sort(); break;
        case 3: Ins_sort(); display(); break;
```

```
        case 4: Quick_sort(0, n - 1); display_first_10(); break;
        case 5: Lin_search(); break;
        case 6: Bin_search(); break;
        case 7: Fibo_search(); break;
        case 8: exit(0); break;
        default: cout << "Please enter a valid choice!\n"; ask();
    }
}

int main(){
    enter();
    return 0;
}
```