Source Code :-

```
/*
  Name - Neevan Redkar
  Roll no - 3069
  CRN - 23103069
  Class - SE
  Batch - S3
  Subject - DSA Assignment 1
*/


#include <iostream>
using namespace std;

struct Student{  //declared struct Student
    int rollNum;
    string name;
    float sgpa;
};

void bsort(struct Student[],int);
void inSort(struct Student[],int);
void qSort(struct Student[],int,int);
int partition(struct Student[],int,int);
void gpasearch(struct Student[],int);
int namesearch(struct Student[],int,int);
void print(struct Student[],int);
//declared all required functions

int main(){
    struct Student se_it[15]={{15,"X",8.77},{12,"P",9.54},{13,"Q",6.22},{14,"W",7.55},-
{5,"E",9.23},
    {6,"R",5.40},{10,"T",8.77},{8,"Y",5.68},{7,"U",9.18},{9,"I",7.89},
    {11,"O",8.90},{1,"A",7.25},{3,"S",8.04},{4,"D",7.79},{2,"F",9.33}};
    //GPA sourced from random number generator online
    int length = sizeof(se_it)/sizeof(se_it[0]);
    //int length using sizeof() inbuilt function

    bsort(se_it,length-1); //called bubble sort function to sort by roll numbers

    inSort(se_it,length-1); //sorted array by alphabetical order of name of Students
    cout<<"Sorted alphabetically"<<endl;
    print(se_it,length); //printed array
    qSort(se_it,0,length-1); //Quick sorted array to calculate toppers
    cout<<"10 toppers in classroom are"<<endl;
    for(int i=14;i>4;i--){ //as array is in ascending order,print from other end
        cout<<se_it[i].rollNum<<" "<<se_it[i].name<<" "<<se_it[i].sgpa<<endl;

    }
    cout<<endl;
    gpasearch(se_it,length); //linear search for CGPA
    int result = namesearch(se_it,length-1,0);
    (result==-1)?cout<<"Given name is not in array"<<endl: cout<<"Given name is present
at array having roll number "<<se_it[result].rollNum<<endl;
    //Used ternary operator to see whether name is in array. If condition met, roll no.
of student returned
    return 0;
}

void print(Student arr[],int length){ //function for printing
   for(int i=0;i<length;i++){
        cout<<arr[i].rollNum<<" "<<arr[i].name<<" "<<arr[i].sgpa;
        cout<<"\n";
    }
```

```
    cout<<endl;
}

void bsort(Student  arr[],int length){ //bubble sort function

   for(int i =0;i<length;i++){
        for(int j=0;j<length;j++){
            if(arr[j].rollNum>arr[j+1].rollNum){
                swap(arr[j+1],arr[j]);    //using in-built swap function in iostream
std.
            }

        }
    }
    cout<<"List sorted by roll number is"<<endl;
    print(arr,length+1);
}



void inSort(struct Student arr[],int length)
{
    int j;
    for(int i=1;i<length;i++){
        Student key=arr[i]; //took a seperate key
        j=i-1;
        while(j>=0){
            if(arr[j].name<key.name){
                break; //ignore if greater
            }else{
                arr[j+1].name=arr[j].name;
                arr[j].name=key.name;
                arr[j+1].rollNum=arr[j].rollNum;
                arr[j].rollNum=key.rollNum;
                arr[j+1].sgpa=arr[j].sgpa;
                arr[j].sgpa=key.sgpa;
                j--; //replace with key if smaller
            }
        }

    }


}


int partition(Student arr[],int low,int high){ //pivot & swap
    float pivot = arr[high].sgpa;
    int i = low-1;
    for(int j = low;j<high;j++){
        if(arr[j].sgpa<=pivot){
            i++;
            swap(arr[i],arr[j]);

        }
    }
    swap(arr[i+1],arr[high]);
    return (i+1);
}

void qSort(struct Student arr[],int low,int high){ //recursive quick sort
    if (low < high) {

    int pi = partition(arr, low, high);
    qSort(arr, low, pi - 1);
    qSort(arr, pi + 1, high);
```

```
  }
}

void gpasearch(Student arr[],int length){
    float target;
    int counter=-1; //counter initialized with -1
    cout<<"Enter CGPA of student to be found ";
    cin>>target;//input of target
    for(int i=0;i<length;i++){ //linear search
        if(target==arr[i].sgpa){
            cout<<"Student having CGPA "<<target<<" is "<<arr[i].name<<" and their roll
no is "<<arr[i].rollNum<<endl;
            counter++;//counter increased if condition met
        }
    }
    if(counter<0){  //if element not found, counter negative
        cout<<"No student having CGPA "<<target<<" found"<<endl;
    }

}

int namesearch(Student arr[],int high,int low){
  inSort(arr,high); //called insertion sort function defined above, to sort in ascending
order of name
  string target;
  cout<<"Enter name to be found"<<endl;
  cin>>target; //input of target element
  while (low <= high) { //Binary search by implementing while loop, divides array
        int mid = low + (high - low) / 2;

        if (arr[mid].name == target){
            return mid; //returns if found
        }
        else if(arr[mid].name < target){
             low = mid + 1; //lower part of array
        }else{
            high = mid - 1;} //takes upper part of array
    }


    return -1; //returns -1 if element not found

}

Output :-

pvg@pvg-HP-ProDesk-400-G4-SFF:~/Desktop/SE_IT_3069/Assignments/Assignment1$ ./-
studentDatabase
List sorted by roll number is
1 A 7.25
2 F 9.33
3 S 8.04
4 D 7.79
5 E 9.23
6 R 5.4
7 U 9.18
8 Y 5.68
9 I 7.89
10 T 8.77
11 O 8.9
12 P 9.54
13 Q 6.22
14 W 7.55
15 X 8.77

Sorted alphabetically
```

```
1 A 7.25
4 D 7.79
5 E 9.23
2 F 9.33
9 I 7.89
11 O 8.9
12 P 9.54
13 Q 6.22
6 R 5.4
3 S 8.04
10 T 8.77
7 U 9.18
14 W 7.55
8 Y 5.68
15 X 8.77

10 toppers in classroom are
12 P 9.54
2 F 9.33
5 E 9.23
7 U 9.18
11 O 8.9
15 X 8.77
10 T 8.77
3 S 8.04
9 I 7.89
4 D 7.79

Enter CGPA of student to be found 8.04
Student having CGPA 8.04 is S and their roll no is 3
Enter name to be found
A
Given name is present at array having roll number 1
```