**DSA Lab Assignment 1**

**Searching and Sorting**

Name – Parth Vijay Raut

Roll no – 23338

Batch – F11

Q. Consider a student database of SEIT class. Database contains different fields of every student like Roll No, Name and SGPA.

a. Design a roll call list, arrange list of students according to roll numbers in ascending order (Use Bubble Sort)

b. Arrange list of students according to name. (Use Insertion sort)

c. Arrange list of students to find out first ten toppers from a class. (Use Quick sort)

d. Search students according to SGPA. If more than one student having same SGPA, then print list of all students having same SGPA.

e. Search a particular student according to name using binary search without recursion.

#include <iostream>

using namespace std;

struct Student{

    int rollno;

    string name;

    float sgpa;

};

class division{

    private:

    Student s[10];

    public:

    int n;

    public:

        void input();

        void display();

        void lsearch();

        void bsearch();

        void ins\_sort();

        void bubble\_sort();

        void selection\_sort();

        void quick\_sort(int low,int high);

}SE11;

void division::input(){

    cout<<"Enter value of n"<<endl;

    cin>>n;

    for(int i=0;i<n;i++){

        cin>>s[i].name>>s[i].rollno>>s[i].sgpa;

    }

}

void division::display(){

    for(int i=0;i<n;i++){

        cout<<s[i].name<<" "<<s[i].rollno<<" "<<s[i].sgpa<<endl;

    }

}

void division::lsearch(){

    int k;

    cout<<"Enter roll no to find"<<endl;

    cin>>k;

    int i=0;

    for( i=0;i<n;i++){

        if(s[i].rollno==k) {

            cout<<s[i].name<<endl;

            break;

        }

    }

    if(i==n) cout<<"Not found"<<endl;

}

void division::bsearch(){

    string key;

    cout<<"Enter name to find"<<endl;

    cin>>key;

    int high = n-1;

    int low = 0;

    int mid = n/2;

    while(low<=high){

        int x = key.compare(s[mid].name);

        if(x==0){

            cout<<s[mid].rollno<<endl;

            break;

        }

        else if(x<0){

            high = mid-1;

        }

        else if(x>0){

            low = mid +1;

        }

        mid = (high + low)/2;

    }

    if(low>high) cout<<"not found"<<endl;

}

void division::ins\_sort(){

    for(int i=1;i<n;i++){

        int j=i-1;

        Student temp = s[i];

        while(s[j].name>temp.name and j>=0){

            s[j+1]=s[j];

            j--;

        }

        s[j+1]=temp;

    }

    for(int i=0;i<n;i++){

        cout<<s[i].name<<" ";

        cout<<s[i].rollno<<" ";

        cout<<s[i].sgpa<<" "<<endl;

    }

}

void division::bubble\_sort(){

    for(int i = 0 ; i<n-1;i++){

        for(int j=0;j<n-i-1;j++){

            if(s[j].rollno>s[j+1].rollno){

                Student temp=s[j];

                s[j] = s[j+1];

                s[j+1] = temp;

            }

        }

    }

    for(int i=0;i<n;i++){

        cout<<s[i].name<<" ";

        cout<<s[i].rollno<<" ";

        cout<<s[i].sgpa<<" "<<endl;

    }

}

void division::selection\_sort(){

    for(int i=0;i<n-1;i++){

        for(int j=i+1;j<n;j++){

            if(s[j].name<s[i].name){

                Student temp = s[i];

                s[i] = s[j];

                s[j] = temp;

            }

        }

    }

    for(int i=0;i<n;i++) {

        cout<<s[i].name<<" ";

        cout<<s[i].rollno<<" ";

        cout<<s[i].sgpa<<" "<<endl;

    }

}

void division::quick\_sort(int low,int high){

    if(low<=high){

    int pivot = s[low].sgpa;

    int i = low+1;

    int j = high;

    while(i<j){

        while(s[i].sgpa<pivot and i<=high) i++;

        while(s[j].sgpa>pivot and j>=low) j--;

        if(i<j){

            Student tempt = s[i];

            s[i] = s[j];

            s[j] = tempt;

            i++;

            j--;

        }

    }

    if(s[low].sgpa>=s[j].sgpa) {

        Student temp = s[low];

        s[low] = s[j];

        s[j] = temp;

    }

    quick\_sort(low,j-1);

    quick\_sort(j+1,high);

}

}

int main() {

   while(1){

       cout<<"1. Input"<<endl;

       cout<<"2. Display"<<endl;

       cout<<"3. Lsearch"<<endl;

       cout<<"4. Bsearch"<<endl;

       cout<<"5. Insertion Sort"<<endl;

       cout<<"6. Bubble Sort"<<endl;

       cout<<"7. Selection Sort"<<endl;

       cout<<"8. Quick Sort"<<endl;

       cout<<"9. Exit"<<endl;

       int ch;

       cout<<"Enter command"<<endl;

       cin>>ch;

       switch (ch){

           case 1:

           SE11.input();

           break;

           case 2:

           SE11.display();

           break;

           case 3:

           SE11.lsearch();

           break;

           case 4:

           SE11.bsearch();

           break;

           case 5:

           SE11.ins\_sort();

           break;

           case 6:

           SE11.bubble\_sort();

           break;

           case 7:

           SE11.selection\_sort();

           break;

           case 8:

           SE11.quick\_sort(0,SE11.n-1);

           break;

           case 9:

           exit(0);

       }

       cout<<endl;

   }

    return 0;

}