NAME – AARYAN BAIRAGI

ROLL NUMBER- 27002

CLASS- SE IT B

#include<iostream>

#include<stdlib.h>

#include<iomanip>

#include<stdlib.h>

using namespace std;

int n;

class Student{

        int rollNo;

        string name;

        float sgpa;

        public:

        void getData();

        void ShowData();

        void rollList();

        void findsgpa();

        void FindName();

        void NameList();

        void TopTen();

        //extra function

        int partition(Student ar[],int lb,int ub);

        void QuickSort(Student ar[],int lb,int ub);

}s[25];

    void swap(int & a, int &b){

    int temp;

    temp=a;

    a=b;

    b=temp;

    }

void ask(int mid){

    char ch;

    cout<<endl<<"Studemt Found ";

    cout<<endl<<"do you want to see its data (y/n) ";

        cin>>ch;

    if(ch=='y'){

        s[mid].ShowData();

    }

}

//main function

main()

{

    int m;

    int count=1;

    while(1)

    {

    cout<<endl<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

    cout<<endl<<"\t\t"<<"MAIN MENU";

    cout<<endl<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

    cout<<endl<<"|\t"<<"1. Enter data                |";

    cout<<endl<<"|\t"<<"2. Show Data                 |";

    cout<<endl<<"|\t"<<"3. Search Student by SGPA    |";

    cout<<endl<<"|\t"<<"4. Show Roll List            |";

    cout<<endl<<"|\t"<<"5. Search student by name    |";

    cout<<endl<<"|\t"<<"6. Name List                 | ";

    cout<<endl<<"|\t"<<"7. Topper List               | ";

    cout<<endl<<"|\t"<<"8. Exit                      |";

    cout<<endl<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

    cout<<endl<<"Enter your choice ";

        cin>>m;

    cout<<"............................................."<<endl;

    switch(m)

        {

        case 1:

            cout<<"Enter number of student ";

                cin>>n;

            cout<<"............................................."<<endl;

            for(int i=0;i<n;i++)

            {

                cout<<endl<<"Enter data for students "<<count++<<endl;

                s[i].getData();

                cout<<".............................................";

            }

            break;

        case 2:

            for(int i=0;i<n;i++)

                s[i].ShowData();

            break;

        case 3:

            s[0].findsgpa();

            break;

        case 4:

            s[0].rollList();

            break;

        case 8:

                exit(0);

        case 5:

            s[0].FindName();

            break;

        case 6:

            s[0].NameList();

            break;

        case 7:

            s[0].TopTen();

            break;

        }

    }

}

//funcion defination of getData from class Student

void Student :: getData()

{

    cout<<"Enter Name of Student : ";

    cin>>name;

    cout<<"Enter Roll no of Student : ";

    cin>>rollNo;

    cout<<"Enter Student SGPA :  ";

    cin>>sgpa;

    cout<<endl;

    cout<<endl;

}

//function defination of ShowData

void Student::ShowData(){

    cout<<endl;

    cout<<"NAME :"<<name<<"\tRoll no :"<<rollNo;

    cout<<fixed<<setprecision(2)<<"\tSGPA :"<<sgpa;

    }

//function defination of findsgpa

void Student::findsgpa()

{

    int count =0;

    float toFind;

        count=0;

    cout<<endl<<"Enter sgpa to search ";

        cin>>toFind;

    for(int i=0;i<n;i++){

        if(s[i].sgpa==toFind){

            ask(i);

            count++;

        }

    }

    if(count ==0)

        cout<<toFind<<"sgpa not fount";

}

//function defination of rollList

void Student::rollList()

{

    int rtemp;

    string tname;

    float gtemp;

    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)

            {

                rtemp = s[j].rollNo;

                s[j].rollNo = s[j+1].rollNo;

                s[j+1].rollNo = rtemp;

                tname = s[j].name;

                s[j].name = s[j+1].name;

                s[j+1].name = tname;

                gtemp = s[j].sgpa;

                s[j].sgpa = s[j+1].sgpa;

                s[j+1].sgpa = gtemp;

            }

        }

    }

    cout<<"\n Sorted Roll Call List\n";

    for(int i=0;i<n;i++)

        s[i].ShowData();

    cout<<endl;

}

//Defination of Name List

void Student::NameList(){

    string stemp;

    int rtemp;

    float gtemp;

    for(int i=0;i<n-1;i++)

        for(int j=i+1;j>0;j--)

            if(s[j].name[0]<s[j-1].name[0])

            {

                stemp=s[j].name;

                s[j].name=s[j-1].name;

                s[j-1].name=stemp;

                gtemp = s[j].sgpa;

                s[j].sgpa = s[j-1].sgpa;

                s[j-1].sgpa = gtemp;

                rtemp = s[j].rollNo;

                s[j].rollNo = s[j-1].rollNo;

                s[j-1].rollNo = rtemp;

            }

    for(int i=0;i<n;i++)

        s[i].ShowData();

}

//defination of search name

void Student::FindName(){

    int temp;

    string sname,tname;

    float gtemp;

    cout<<"Enter Name to search ";

        cin>>sname;

    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)

            {

                temp = s[j].rollNo;

                s[j].rollNo = s[j+1].rollNo;

                s[j+1].rollNo = temp;

                tname = s[j].name;

                s[j].name = s[j+1].name;

                s[j+1].name = tname;

                gtemp = s[j].sgpa;

                s[j].sgpa = s[j+1].sgpa;

                s[j+1].sgpa = gtemp;

            }

        }

    }

    int f=0,r=n,mid;

    while (f <= r) {

    mid = (f + r) / 2;

    if (s[mid].name==sname) {

        ask(mid );

        break;

        }

    else if(s[mid].name <sname)

        f = mid + 1;

    else

        r = mid - 1;

    }

    if (f > r)

        cout << "\nNot Found \n";

}

int Student::partition(Student ar[],int lb,int ub)

    {

    string stemp;

    int rtemp;

    float gtemp;

    int pivot=ar[lb].sgpa,start=lb,end=ub;

    while(start<end){

        while(ar[start].sgpa<=pivot)

            start++;

        while(ar[end].sgpa>pivot)

            end--;

        if(start<end){

            rtemp = s[end].rollNo;

            s[end].rollNo = s[start].rollNo;

            s[start].rollNo = rtemp;

            stemp = s[end].name;

            s[end].name = s[start].name;

            s[start].name = stemp;

            gtemp = s[end].sgpa;

            s[end].sgpa = s[start].sgpa;

            s[start].sgpa = gtemp;

        }

    }

    swap(ar[lb].sgpa,ar[end].sgpa);

    return end;

    }

void Student::QuickSort(Student ar[],int lb,int ub){

    if(lb<ub){

        int pos=partition(ar,lb,ub);

        QuickSort(ar,lb,pos-1);

        QuickSort(ar,pos+1,ub);

        }

    }

void Student::TopTen(){

    QuickSort(s,0,(n-1));

    for(int i=n-1;i>=0;i--)

        s[i].ShowData();

}