#include <iostream> using namespace std;

#include<string.h>

int const size=3;

struct student{ int rno; char name[20];

float SGPA;

};

void accept(struct student list[size]); void display(struct student list [80]); void displayTop(struct student list[80]); void bubbleSort(struct student list[size]); void insertSort(struct student list[size]); void quickSort(struct student list[size],int,int); void search(struct student list[size] );

void binarysearch(struct student list[size]);

main()

{ int ch, i;

struct student data[20]; accept (data); cout<<"\n 1:Bubble Sort"; cout<<"\n 2:Insertion Sort"; cout<<"\n 3:Quick Sort"; cout<<"\n 4:Search"; cout<<"\n 5:Binary Search"; cout<<"\n Select your choice:";

cin>>ch;

switch(ch)

{ case 1:

bubbleSort(data); display(data);

break;

case 2: insertSort(data); display(data); break;

case 3:

quickSort(data,0,size-1); displayTop(data);

break;

case 4: search(data);

break;

case 5:

binarysearch(data);

break;

default:

cout<<"Invalid choice....";

}

}

void accept(struct student list[size])

{

int i;

for (i=0;i<size;i++)

{

cout<<"Enter rollno,name & SGPA:";

cin>>list[i].rno>>list[i].name>>list[i].SGPA;

}

}

void display(struct student list[80])

{

int i;

cout<<"\n Roll no \t Name \t SGPA \n";

for(i=0;i<size;i++)

{

cout<<"\n"<<list[i].rno<<"\t"<<list[i].name<<"\t"<<list[i].SGPA;

}

}

void displayTop(struct student list[80])

{

int i;

cout<<"\n\nRollno\tName\tSGPA\n";

for(i=0;i<3;i++)

{

cout<<"\n"<<list[i].rno<<"\t"<<list[i].name<<"\t"<<list[i].SGPA;

}

}

void bubbleSort(struct student list[size])

{

int i,j;

struct student temp; for(i=0;i<size-1;i++)

{

for(j=0;j<(size-1-i);j++)

{

if(list[j].rno>list[j+1].rno)

{

temp=list[j]; list[j]=list[j+1];

list[j+1]=temp;

}

}

}

}

void insertSort(struct student list[size])

{

int k,j;

struct student temp;

for(k=1;k<size;k++)

{

temp=list[k]; j=k-1;

while(strcmp(list[j].name,temp.name)>0&&j>=0)

{

list[j+1]=list[j];

--j;

}

list[j+1]=temp;

}

}

void quickSort(struct student list[size],int first,int last)

{ int pivot,i,j;

struct student temp;

if(first<last)

{

pivot=first; i=first; j=last;

while(i<j)

{

while(list[i].SGPA>=list[pivot].SGPA&&i<last)

i++;

while(list[j].SGPA<list[pivot].SGPA)

j--; if(i<j)

{

temp=list[i];

list[i]=list[j]; list[j]=temp;

}

}

temp=list[pivot]; list[pivot]=list[j]; list[j]=temp; quickSort(list,first,j-1); quickSort(list,j+1,last);

}

}

void search(struct student list[size])

{

float SGPA;

int i;

cout<<"\n Enter SGPA"; cin>>SGPA;

cout<<"\n Rollno \t Name \t SGPA \n";

for(int i=0;i<size;i++)

{

if(SGPA==list[i].SGPA)

cout<<"\n"<<list[i].rno<<"\t"<<list[i].name<<"\t"<<list[i].SGPA;

}

}

void binarysearch(struct student list[size])

{

int k, lower, upper, mid; char search[80];

cout<<"\n Enter name of the students you want to search"; cin>>search;

lower=0; upper=size-1;

mid=(lower+upper)/2;

while(lower<=upper)

{

if(strcmp(list[mid].name,search)<0) lower=mid+1;

else if(strcmp(list[mid].name,search)==0)

{

cout<<"\n"<<list[mid].rno<<"\t"<<list[mid].name<<"\t"<<list[mid].SGPA; break;

} else

upper=mid-1;

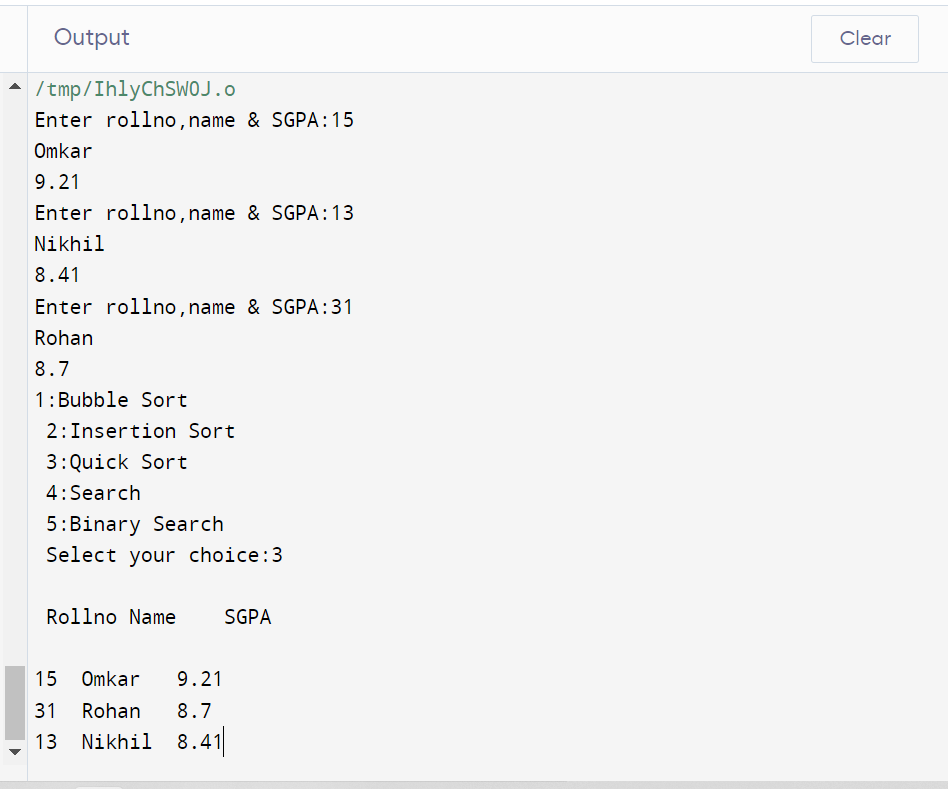
mid=(lower+upper)/2;

}

if(lower>upper) cout<<search<<"not found in the list";

}

**OUTPUT:-**

****