#include <iostream>

#include <string.h>

using namespace std;

int const size= 15;

struct student

{

int roll\_no;

char name[30];

float SGPA;

};

void accept(struct student list[size]);

void display(struct student list[80]);

void bubble\_sort(struct student list[size]);

void insert\_sort(struct student list[size]);

void quick\_sort(struct student list[size],int first,int last);

void search(struct student list[size]);

void binary\_search(struct student list[size]);

// ACCEPT FUNCTION

void accept(struct student list[size])

{

int i;

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

{

cout<<"\nEnter Roll-Number, Name, SGPA:";

cin>>list[i].roll\_no>>list[i].name>>list[i].SGPA;

}

}

// DISPLAY FUNCTION

void display(struct student list[80])

{

int i;

cout<<"\n Roll-Number \t Name \t SGPA \n";

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

{

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

}

cout<<"\n";

}

// BUBBLE SORT FUNCTION

void bubble\_sort(struct student list[size])

{

int i,j,c=0;

struct student key;

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

{

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

{

if(list[j].roll\_no>list[j+1].roll\_no)

{

key=list[j];

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

list[j+1]=key;

}

c++;

}

}

cout<<"\n";

cout<<"\nNumber of passes is: "<<c;

}

// INSERTION SORT FUNCTION

void insert\_sort(struct student list[size])

{

int k,j,c=0;

struct student key;

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

{

key=list[k];

j=k-1;

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

{

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

j--;

c++;

}

list[j+1]=key;

}

cout<<"\nNumber of passes is: "<<c;

}

// QUICK SORT FUNCTION

void quick\_sort(struct student list[size],int first,int last)

{

int pivot,i,j;

if(first < last)

{

pivot = first;

i = first;

j = last;

while(i < j)

{

while(list[i].roll\_no <= list[pivot].roll\_no && i < last)

{

i++;

}

while(list[j].roll\_no > list[pivot].roll\_no)

{

j--;

}

}

swap(list[pivot].roll\_no, list[j].roll\_no);

quick\_sort(list,first,j-1);

quick\_sort(list,j+1,last);

}

}

// SEARCH FUNCTION

void search(struct student list[size])

{

float SGPA;

int i;

cout<<"\n Enter SGPA:";

cin>>SGPA;

cout<<"\n Roll- Number \t Name \t SGPA \n";

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

{

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

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

}

}

// BINARY SEARCH FUNCTION

void binary\_search(struct student list[size])

{

int k, lower,upper,mid;

char search[80];

cout<<"\n Enter Name Of Student 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].roll\_no<<"\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";

}

// MAIN FUNCTION

int main()

{

int ch,i;

struct student data[20];

accept(data);

do

{

cout<<"\n";

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 6) Exit \n";

cout<<"\n Select Your Choice: ";

cin>>ch;

switch(ch)

{

case 1:

bubble\_sort(data);

display(data);

break;

case 2:

insert\_sort(data);

display(data);

break;

case 3:

quick\_sort(data,0,size-1);

display(data);

break;

case 4:

search(data);

display(data);

break;

case 5:

binary\_search(data);

display(data);

break;

case 6:

cout<<"\nYou Have Successfully Exitted!!!.";

break;

default:

cout<<"\nPlease Enter Valid Choice.\n";

}

}

while(ch!=6);

return 0;

}

Select Your Choice: 6

You Have Successfully Exitted!!!.