ASSIGNMENT NO.1

#include<iostream>

using namespace std;

struct student

{

int roll\_no;

string name;

float sgpa;

};

void bubblesort(student arr[],int n)

{

for(int i=0;i<4;i++)

{

for(int j=0;j<4;j++)

{

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

{

student a=arr[j];

arr[j]=arr[j+1];

arr[j+1]=a;

}

}

}

}

void insertionsort(student arr[],int n)

{

for(int i=1;i<n;i++)

{

student key=arr[i];

int j=i-1;

while(j>=0 && arr[j].name>key.name)

{

arr[j+1]=arr[j];

j=j-1;

}

arr[j+1]=key;

}

}

int partition(student arr[],int low,int high)

{

float pivot=arr[high].sgpa;

int i=low-1;

for(int j=low;j<high;j++)

{

if(arr[j].sgpa>pivot)

{

i++;

student temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

student temp=arr[i+1];

arr[i+1]=arr[high];

arr[high]=temp;

return i+1;

}

void quicksort(student arr[],int low,int high)

{

if(low<high)

{

int pi=partition(arr,low,high);

quicksort(arr,low,pi-1);

quicksort(arr,pi+1,high);

}

}

void searchSgpa(student s[],int n,float search)

{

bool found=false;

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

{

if(s[i].sgpa==search)

{

if(!found)

{

cout<<"student with SGPA::"<<search<<endl;

found = true;

}

cout<<"name:"<<s[i].name<<endl;

}

}

if(!found)

{

cout<<"no students found with SGPA"<<search<<endl;

}

}

int binarySearch(student arr[],int size,string key)

{

int start=0;

int end=size-1;

int mid=start+(end-start)/2;

while(start<=end)

{

if(arr[mid].name==key)

{

return mid;

}

if(key>arr[mid].name)

{

start=mid+1;

}

else

{

end=mid-1;

}

mid=start+(end-start)/2;

}

return-1;

}

int main()

{

student s[5];

for(int i=0;i<5;i++)

{

cout<<"Enter the details of student no."<<i+1<<":-"<<endl;

cout<<endl;

cout<<"Enter the name of the student:-";

cin>>s[i].name;

cout<<"Enter the roll no. of the student:-";

cin>>s[i].roll\_no;

cout<<"Enter the SGPA of the student:-";

cin>>s[i].sgpa;

cout<<"name:"<<s[i].name<<endl;

cout<<"roll no.:"<<s[i].roll\_no<<endl;

cout<<"sgpa:"<<s[i].sgpa<<endl;

cout<<endl;

}

//bubble sort

bubblesort(s,5);

cout<<"sorted roll numbers are::"<<endl;

cout<<endl;

for(int j=0;j<5;j++)

{

cout<<s[j].roll\_no<<endl;

}

cout<<endl;

//insertion sort

insertionsort(s,5);

cout<<"sorted names of students are::"<<endl;

cout<<endl;

for(int i=0;i<5;i++)

{

cout<<s[i].name<<endl;

}

cout<<endl;

//quick sort

quicksort(s,0,4);

cout<<"Top three students are::"<<endl;

cout<<endl;

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

{

cout<<"Name::"<<s[i].name<<"Roll no::"<<s[i].roll\_no<<"SGPA::"<<s[i].sgpa<<endl;

}

cout<<endl;

//search by sgpa/linear search

float search;

cout<<"Enter SGPA to search::";

cin>>search;

searchSgpa(s,5,search);

cout<<endl;

//binary search

string key;

cout<<"Enter a name to search::"<<endl;

cin>>key;

cout<<endl;

insertionsort(s,5);

int index;

index=binarySearch(s,5,key);

if(index!=-1)

{

cout<<"student found::"<<endl;

cout<<"name::"<<s[index].name<<endl;

cout<<"roll no::"<<s[index].roll\_no<<endl;

cout<<"sgpa::"<<s[index].sgpa;

}

else

{

cout<<"student not found"<<endl;

}

return 0;

}

OUTPUT



