/\*Consider a student database of SEIT class (at least 15 records). Database contains different fields of every student like Roll No, Name and SGPA.(array of structure)

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 alphabetically. (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. (all the student records having the presence of search key should be displayed)

\*/

#include<iostream>

#include<string.h>

using namespace std;

typedef struct student

{

int roll\_num;

char name [20];

float marks;

}stud;

void create(stud s[20],int n);

void display(stud s[20],int n);

void bubble\_sort(stud s[20],int n);

void insertionSort(stud s[20],int n);

void quick\_sort(stud s[20],int,int);

int partition(stud s[20],int,int);

void search(stud s[20],int n,int key);

int bsearch(stud s[20], char x[20],int low,int high);

int main()

{

stud s[20];

int ch,n,key,result;

char x[20];

do

{

cout<<"\n 1) Create Student Database ";

cout<<"\n 2) Display Student Records ";

cout<<"\n 3) Bubble Sort ";

cout<<"\n 4) Insertion Sort ";

cout<<"\n 5) Quick Sort ";

cout<<"\n 6) Linear search ";

cout<<"\n 7) Binary search ";

cout<<"\n 8) Exit ";

cout<<"\n Enetr Your Choice:=";

cin>>ch;

switch(ch)

{

case 1:

cout<<"\n Enter The Number Of Records:=";

cin>>n;

create(s,n);

break;

case 2:

display(s,n);

break;

case 3:

bubble\_sort(s,n);

break;

case 4:

insertionSort(s,n);

break;

case 5:

quick\_sort(s,0,n-1);

cout<<"\n"<< "\t"<< "Roll No"<< "\t"<<" Name" <<"\t"<< "Marks";

for(int i=n-1; i>=n-10; i--)

{

cout<<"\n";

cout<<"\t "<< s[i].roll\_num<<"\t "<<s[i].name<<"\t "<<s[i].marks;

}

break;

case 6:

cout<<"\n Enter the marks which u want to search:=";

cin>>key;

search(s,n,key);

break;

case 7:

cout<<"\n Enter the name of student which u want to search:=";

cin>>x;

insertionSort(s,n);

result=bsearch(s,x,0,(n-1));

if(result==-1)

{

cout<<" \n Student name you want to search for is not present ! \n";

}

else

{

cout<<" \n The student is present :\t" << s[result].name;

}

break;

case 8:return 0;

default:cout<<"\n Invalid choice !! Please enter your choice again."<<endl;

}

}while(ch!=8);

}

void create(stud s[20],int n)

{

int i;

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

{

cout<<"\n Enter the roll number:=";

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

cout<<"\n Enter the Name:=";

cin>>s[i].name;

cout<<"\n Enter the marks:=";

cin>>s[i].marks;

}

}

void display(stud s[20],int n)

{

int i;

cout<<"\n"<< "\t"<< "Roll No"<< "\t"<<" Name" <<"\t"<< "Marks";

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

{

cout<<"\n";

cout<<"\t "<< s[i].roll\_num<<"\t "<<s[i].name<<"\t "<<s[i].marks;

}

}

//bubble sort to sort in ascending order on roll number

void bubble\_sort(stud s[20],int n)

{

int i,j;

stud temp;

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

{

for(j=0;j<n-i;j++)

{

if(s[j].roll\_num>s[j+1].roll\_num)

{

temp=s[j];

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

s[j+1]=temp;

}

}

}

}

// insertion sort to sort on names in ascending order

void insertionSort(stud s[20], int n)

{

int i, j;

stud key;

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

{

key= s[i];

j = i - 1;

/\* Move elements of arr[0..i-1], that are

greater than key, to one position ahead

of their current position \*/

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

{

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

j = j - 1;

}

s[j + 1]= key;

}

}

//Quick sort to sort on marks

void quick\_sort(stud s[20], int l,int u)

{

int j;

if(l<u)

{

j=partition(s,l,u);

quick\_sort(s,l,j-1);

quick\_sort(s,j+1,u);

}

}

int partition(stud s[20], int l,int u)

{

int i,j;

stud temp, v;

v=s[l];

i=l;

j=u+1;

do

{

do

i++;

while(s[i].marks<v.marks&&i<=u);

do

j--;

while(v.marks<s[j].marks);

if(i<j)

{

temp=s[i];

s[i]=s[j];

s[j]=temp;

}

}while(i<j);

s[l]=s[j];

s[j]=v;

return(j);

}

// linear search for marks if more than one student having same marks print all of them

void search(stud s[20],int n,int key)

{

int i;

cout<<"\n"<< "\t"<< "Roll No"<< "\t"<<" Name" <<"\t"<< "Marks";

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

{

if(key==s[i].marks)

{

cout<<"\n\t "<< s[i].roll\_num<<"\t "<<s[i].name<<"\t "<<s[i].marks;

}

}

}

int bsearch(stud s[20], char x[20],int low,int high)

{

int mid;

while(low<=high)

{

mid=(low+high)/2;

if(strcmp(x,s[mid].name)==0)

{

return mid;

}

else if(strcmp(x,s[mid].name)<0)

{

high=mid-1;

}

else

{

low=mid+1;

}

}

return -1;

}

***OUTPUT***

1) Create Student Database

2) Display Student Records

3) Bubble Sort

4) Insertion Sort

5) Quick Sort

6) Linear search

7) Binary search

8) Exit

Enetr Your Choice:=1

Enter The Number Of Records:=4

Enter the roll number:=3

Enter the Name:=ABC

Enter the marks:=98

Enter the roll number:=4

Enter the Name:=PQR

Enter the marks:=97

Enter the roll number:=2

Enter the Name:=PQR

Enter the marks:=95

Enter the roll number:=1

Enter the Name:=XYZ

Enter the marks:=89

1) Create Student Database

2) Display Student Records

3) Bubble Sort

4) Insertion Sort

5) Quick Sort

6) Linear search

7) Binary search

8) Exit

Enetr Your Choice:=2

Roll No Name Marks

3 ABC 98

4 PQR 97

2 PQR 95

1 XYZ 89

1) Create Student Database

2) Display Student Records

3) Bubble Sort

4) Insertion Sort

5) Quick Sort

6) Linear search

7) Binary search

8) Exit

Enetr Your Choice:=3

1) Create Student Database

2) Display Student Records

3) Bubble Sort

4) Insertion Sort

5) Quick Sort

6) Linear search

7) Binary search

8) Exit

Enetr Your Choice:=2

Roll No Name Marks

1 XYZ 89

2 PQR 95

3 ABC 98

4 PQR 97

1) Create Student Database

2) Display Student Records

3) Bubble Sort

4) Insertion Sort

5) Quick Sort

6) Linear search

7) Binary search

8) Exit

Enetr Your Choice:=4

1) Create Student Database

2) Display Student Records

3) Bubble Sort

4) Insertion Sort

5) Quick Sort

6) Linear search

7) Binary search

8) Exit

Enetr Your Choice:=2

Roll No Name Marks

3 ABC 98

2 PQR 95

4 PQR 97

1 XYZ 89

1) Create Student Database

2) Display Student Records

3) Bubble Sort

4) Insertion Sort

5) Quick Sort

6) Linear search

7) Binary search

8) Exit

Enetr Your Choice:=5

Roll No Name Marks

3 ABC 98

4 PQR 97

2 PQR 95

1 XYZ 89

1) Create Student Database

2) Display Student Records

3) Bubble Sort

4) Insertion Sort

5) Quick Sort

6) Linear search

7) Binary search

8) Exit

Enetr Your Choice:=2

Roll No Name Marks

1 XYZ 89

2 PQR 95

4 PQR 97

3 ABC 98

1) Create Student Database

2) Display Student Records

3) Bubble Sort

4) Insertion Sort

5) Quick Sort

6) Linear search

7) Binary search

8) Exit

Enetr Your Choice:=6

Enter the marks which u want to search:=89

Roll No Name Marks

1 XYZ 89

1) Create Student Database

2) Display Student Records

3) Bubble Sort

4) Insertion Sort

5) Quick Sort

6) Linear search

7) Binary search

8) Exit

Enetr Your Choice:=7

Enter the name of student which u want to search:=ABC

The student is present : ABC

1) Create Student Database

2) Display Student Records

3) Bubble Sort

4) Insertion Sort

5) Quick Sort

6) Linear search

7) Binary search

8) Exit

Enetr Your Choice:=8