Name: Pratik Rameshwar Shinde
Roll No.: 2274
Class: SE-IT
Div: B Batch : B-3
Subject: DSA LAB

Practical I : Searching and Sorting

Aim: Consider a student database of SE-IT class.
Databmse contains different fields of every student like Roll No, Name and SGPA.

a. Design a roll call list, arrange list of students according to roll numbers in ascending order (Use Bubble sort)

b. b. Arrange list of students according to name. (Use Insertion sort)

c. c. Arrange list of students to find out first ten toppers from a class. (Use Quick sort)

d. d. Search students according to SGPA. If more than one student having same SGPA, then print list of all students having same SGPA

e. e. Search a particular student according to name using binary search without recursion.(Students having same name should be displayed)

******************************PROGRAM******************************

```
#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
ubble_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
last, int first);

int main()

stud s[20]; int
ch,key,n,result;
char x[20]; do
{
cout<<"\n  l) Create Student Database " ,
cout<<"\n  2)       Display Student Database " ;
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  Enter Your Choice: ";
cin>>ch;
switch(ch)

case 1:
cout<<"\n l
cin>>n;
create(s,n);Enter The Number Of Records: " ,
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"<<"
for(int i=n-1; i>=n-10; i--)                    Name"           "Marks";

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


break;


case 6:
      "\nEnter the marks which you want to search: ";
cin>>key; search(s,n,key); break;


case 7:
      "\nEnter the name Of student which you want to search: ';


  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. in";
      else

      cout<<"in The student is present: <<s[result]




case 8:return O;
   default:cout<<"\nInvalid choice !! Please enter your choice again."<<endl;
}
}while(ch!=8);
}

void create(stud             n)

int i;

for(i=0;i<n;i++)
{
cout<<"\n EnterEnter the roll number: ".
cin>>s[i].roll_num;
cout<<"\n Enter  Enter the Name: '.
cin>>s[i].name;
cout<<"\n Enter  Enter the marks: '.
cin>>s[i].marks;
}
}

void display(stud             n)

   i;
cout<<"\n"<< "\t"<<"ROII No"<< "\t"<<"Name"              "Marks";

int
```

```
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 )ubble_sort(stud s[20],int n)

int i,j;
stud temp;
for(i=1;i<n;i++)
{
for(j=();j<n-i;j++)

if(s[j].roll_num>s[j+l].roll_num)
{
temp=s[j];
s[j]=s[j+1];
s[j+l ]=temp;
}
}
}
}
```

// insertion sort to sort on names in ascending order

```
void insertionSort(stud s[20], int n)         //for arranging data alphabetically

   int i, j; stud
   key;
```

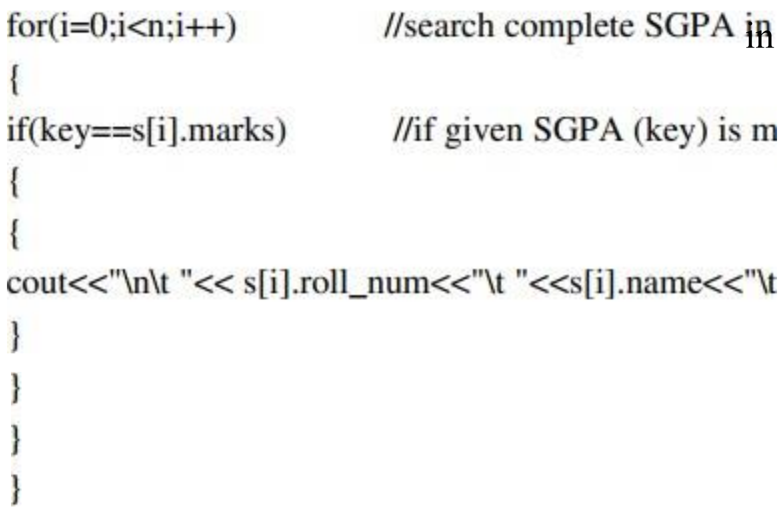
```
for (1 — l; i < n;

   key= s[i]; j
   = i - l;

   /* Move elements Of         that are greater
   than key, to one position ahead
   Of their Current position */

   while (j >= 0 && strcmp(s[j].name,
   {
      s[j + 1]= s[j];                          key .name)>0)

      j =j - l;

   s[j+      key;
```

[Quick sort to sort on marks (SGPA)

```
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+l,u);



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



int
```

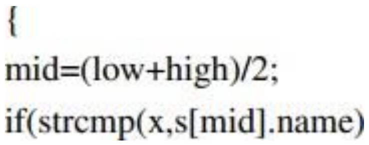
```
i,j; stud temp,
   v;
   i=l;

  do

     do

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

       do

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

       if(i<j)

          temp=s[i];
         s[i]=s[j];
         s[j]=temp;
       }
   }while(i<j);

   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)

    i;
cout<<"\n"<< "\t"<<"ROII No"<< "\t"<<" Name"                    "Marks";

int
```



```
for(i=0;i<n;i++)            //search complete SGPA in array    //if given SGPA (key) is
{                                                                  matched with list array
if(key==s[i].marks)         //if given SGPA (key) is m                            Of SGPA
{
{
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)                    //compare list Of mid character with searching
character

return mid;                        //case2: if searching character is equal to mid chatacter
in list

else if(strcmp(x,s[mid].name)<())
high=mid-l ;                          //case3: if searching character lower than mid
in list                               character

else

                                     //casel: if searching character greater than mid
low=mid+l ;                          character
   list
return - l ;          "searching character not found in the list i.e low>high}
******************************OUTPUT******************************

[admin@fedora       g++ hfbl .cpp [admin@fedora
          ./a.out

1) Create Student Database
2) Display Student
Database
3)       Bubble Sort
4)       Insertion Sort
5)       Quick Sort
6)       Linear Search 7) Binary Search
8) Exit
Enter Your Choice: 1

Enter The Number Of Records: 15

Enter the roll number: 1

Enter the Name: Venky

Enter the marks: 40

Enter the roll number: 2

Enter the Name: Yash

Enter the marks: 60

Enter the roll number: 3
```

int

```
Enter the Name: Sanket

Enter the marks: 56

Enter the roll number: 4

Enter the Name: Kuldeep

Enter the marks: 45

Enter the roll number: 5

Enter the Name: Jaydeep

Enter the marks: 43

Enter the roll number: 6

Enter the Name: Raj

Enter the marks: 78

Enter the roll number: 7

Enter the Name: Aniket

Enter the marks: 68

Enter the roll number: 8

Enter the Name: Ashmit

Enter the marks: 98

Enter the roll number: 9
Enter the Name: Puru
```

Database

Database

```
Enter the marks: 35

Enter the roll number: 10

Enter the Name: Devesh

Enter the marks: 87

Enter the roll number: 1 1

Enter the Name: Yug

Enter the marks: 67

Enter the roll number: 12

Enter the Name: Samarth

Enter the marks: 51

Enter the roll number: 13

Enter the Name: Kalpesh

Enter the marks: 85

Enter the roll number: 14

Enter the Name: Wasik

Enter the marks: 45

Enter the roll number: 15

Enter the Name: Saurabh Enter the
marks: 90
```

```
1) Create Student
2) Display Student
3)  Bubble Sort
4)  Insertion Sort
5)  Quick Sort
6)  Linear Search
7)  Binary Search
8)  Exit
Enter Your Choice: 2


      Roll No        Name          Marks
      1         Venky         40
      2          Yash 60
      3         Sanket        56
      4         Kuldeep       45
      5         Jaydeep       43
      6         Raj 78
      7         Aniket        68
      8         Ashmit        98
      9         Puru 35
       10       Devesh        87
       11       Yug 67
      12        Samarth       51
      13        Kalpesh       85
      14        Wasik         45
      15        Saurabh       90
1)  Create Student Database
2)  Display Student Database
3)  Bubble Sort
4)  Insertion Sort
5)  Quick Sort
6)  Linear Search
7)  Binary Search
8)  Exit
```

```
                        Database
                         Database
Enter Your Choice: 3

1) Create Student
2) Display Student
3)  Bubble Sort
4)  Insertion Sort
5)  Quick Sort
6)  Linear Search
7)  Binary Search
8)  Exit
Enter Your Choice: 2

     Roll No        Name          Marks
     1        Venky        40
     2         Yash 60
     3        Sanket        56
     4        Kuldeep       45
     5        Jaydeep       43
     6        Raj 78
     7        Aniket        68
     8        Ashmit        98
     9        Puru 35
     10       Devesh        87
     11       Yug 67
     12       Samarth       51
     13       Kalpesh       85
     14       Wasik         45
     15       Saurabh       90
1) Create Student Database
2) Display Student
Database
3)  Bubble Sort
4)  Insertion Sort
5)  Quick Sort
```

```
6)  Linear Search
7)  Binary Search
8)  Exit
Enter Your Choice: 4

1) Create Student
2) Display Student
3)  Bubble Sort
4)  Insertion Sort
5)  Quick Sort
6)  Linear Search
7)  Binary Search
8)  Exit
Enter Your Choice: 2

        Roll No         Name            Marks
         7          Aniket          68
         8          Ashmit          98
         10         Devesh          87
         5          Jaydeep         43
         13         Kalpesh         85
         4          Kuldeep         45
         9          Puru 35
         6          Raj 78
         12         Samarth         51
         3          Sanket          56
         15         Saurabh         90
         1          Venky           40
         14         Wasik           45
         2           Yash 60
         11          Yug 67
1) Create Student Database
2) Display Student
Database
3)  Bubble Sort
4)  Insertion Sort
```

```
                    Database
                     Database
5)  Quick Sort
6)  Linear Search
7)  Binary Search
8)  Exit
Enter Your Choice: 5

        Roll No        Name         Marks
        8        Ashmit       98
        15       Saurabh      90
        10       Devesh       87
        13       Kalpesh      85
        6        Raj 78
        7        Aniket       68
        11       Yug 67
        2        Yash 60
        3        Sanket       56
        12       Samarth      51
1) Create Student Database
2) Display Student Database
3)  Bubble Sort
4)  Insertion Sort
5)  Quick Sort
6)  Linear Search
7)  Binary Search
8)  Exit
Enter Your Choice: 2

        Roll No        Name         Marks
        9        Puru 35
        1        Venky        40
        5        Jaydeep      43
        14       Wasik        45
        4        Kuldeep      45
        12       Samarth      51
        3        Sanket       56
```

2	Yash 60	
11	Yug 67	
7	Aniket	68
6	Raj 78	
13	Kalpesh	85
10	Devesh	87
15	Saurabh	90
8	Ashmit	98

```
1) Create Student Database 2)
Display Student Database
3)  Bubble Sort
4)  Insertion Sort
5)  Quick Sort
6)  Linear Search
7)  Binary Search
8)  Exit
Enter Your Choice: 6

Enter the marks which you want to search: 90

         Roll No         Name          Marks
          15      Saurabh       90
1)  Create Student Database
2)  Display Student Database
3)  Bubble Sort
4)  Insertion Sort
5)  Quick Sort
6)  Linear Search
7)  Binary Search
8)  Exit
Enter Your Choice: 7

Enter the name Of student which you want to search:
Puru

The student is present: Puru
1) Create Student Database
2) Display Student
Database
3)  Bubble Sort
4)  Insertion Sort
5)  Quick Sort
6)  Linear Search
7)  Binary Search
```

```
8)  Exit
Enter Your Choice: 7

Enter the name Of student which you want to search: xtz

Student name you want to search for is not present.


1) Create Student Database 2)
Display Student Database
3)  Bubble Sort
4)  Insertion Sort
5)  Quick Sort
6)  Linear Search
7)  Binary Search
8)  Exit
Enter Your Choice: 8
[admin@fedora       AC
[admin@fedora~]$
```