Module 2: Searching & Hashing Techniques

1) Linear Search

```
#include<iostream.h>
#include<conio.h>
void main()
{
int n,a[20], i ,k,index;
clrscr();
cout<<"Enter no of elements:";
cin>>n;
cout<<"\nEnter array elements:";
for(i=0;i<n;i++)
{
cin>>a[i];
}
cout<<"\nEnter the element to be searched:";

cin>>k;
for(i=0;i<n;i++)
{
if(a[i]==k)
{
index=i;
break;
} }
if(a[i]==k)
{
cout<<"The element is form at index"<<index;
}
else      {
cout<<"The element not found ";
}
getch();
}

/* Output1 -
Enter no of elements:5

Enter array elements:1 2 3 4 5

Enter the element to be searched:5
The element is form at index4
```

Output2 -
Enter no of elements:5

Enter array elements:1 2 3 4 5

Enter the element to be searched:6
The element not found
*/

2) Binary Search

```
#include<iostream.h>
#include<conio.h>
void main()
{
clrscr();
int n,a[25],k,low,high,i;
float mid;
cout<<"Enter elements:";
cin>>n;
cout<<"\nEnter an array element:";
for(i=0;i<n;i++)
{
cin>>a[i];
}
low=0;
high=n-1;
mid=(low+high)/2;
cout<<"Enter element to be search:";
cin>>k;
while(low<=high)
{
if(a[mid]<k)
low=mid+1;
else
if(a[mid]==k)
{
cout<<"\nElements found at index"<<mid;
break;
}
else
high=mid-1;

mid=(low+high)/2;
```

```
}
if(low>high)
{
cout<<"\nElement not found";
}

getch();
}
/* Output1 -
Enter elements:5

Enter an array element:1 2 3 4 5
Enter element to be search:4

Elements found at index3

Output2 -
Enter elements:5

Enter an array element:1 2 3 4 5
Enter element to be search 7
Element not found

*/
```

HASHING TECHNIQUES

3) Linear Probe

```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
#define max 10
class Hash
{
  private:
   int a[max];
  public:
   Hash();
   int create(int);
   void linear_prob(int,int),display();
   };

   Hash::Hash()
   {
    int i;
    for(i=0;i<max;i++)
    {
      a[i]=-1;
      }
      }
   int Hash::create(int num)
   {
    int key;
    key=num%10;
    return key;
    }
   void Hash::linear_prob(int key,int num)
   {
    int flag,i,count=0;
    flag=0;
    if(a[key]==-1)
    {
      a[key]=num;
      }
    else
    {
      i=0;
      while(i<max)
```

```
    {
        if(a[i]!=-1)
          count++;
          i++;
          }
    if(count==max)
    {
        cout<<"\n hash table is full"<<num<<"cannot be inserted";
        display();
        getch();
        exit(1);
        }
    for(i=key+1;i<max;i++)
        if(a[i]==-1)
        {
          a[i]=num;
          flag=1;
          break;
          }
    for(i=0;i<key && flag==00;i++)
         if(a[i]==-1)
         {
           a[i]=num;
           flag=1;
           break;
           }
           }
           }
 void Hash::display()
 {
   int i;
   cout<<"\n the hash table is"<<endl;
   for(i=0;i<max;i++)
   {
    cout<<"\n"<<i<<" "<<a[i];
    }   }
void main()
{
   int num,key;
   char ans;
   Hash h;
   clrscr();
   cout<<"\n Collision Handling By Linera Probbing";
   do
   {
    cout<<"\n Enter the number:";
```

```
        cin>>num;
        key=h.create(num);
        h.linear_prob(key,num);
        cout<<"\nDo you wish to continue?(yes/no)";
            ans=getch();
            }
            while(ans=='y');
            h.display();
            getch();
            }

            /* Output -
             Collision Handling By Linera Probbing
 Enter the number:72

Do you wish to continue?(yes/no)
 Enter the number:27

Do you wish to continue?(yes/no)
 Enter the number:36

Do you wish to continue?(yes/no)
 Enter the number:24

Do you wish to continue?(yes/no)
 the hash table is

0 -1
1 -1
2 72
3 -1
4 24
5 -1
6 36
7 27
8 -1
9 -1

   Enter the number:
43

Do you wish to continue?(yes/no)
 Enter the number:81

Do you wish to continue?(yes/no)
 Enter the number:92
```

Do you wish to continue?(yes/no)
Enter the number:101

Do you wish to continue?(yes/no)
the hash table is

0 -1
1 81
2 92
3 43
4 101
5 -1
6 -1
7 -1
8 -1
9 -1
*/