Assignment no:-1

Input:

```
#include<iostream>
#include<string.h>
using namespace std;

 struct node
{
          int value;
         node* next;
}*HashTable[10];

class hashing
{
public:
hashing()
{
for(int i=0 ; i<10 ; i++)
{
                        HashTable[i]=NULL;
             }
 }

int HashFunction(int value)
{
  return (value%10);
 }
 node* create_node(int x)
{
              node* temp=new node;
             temp->next=NULL;
             temp->value=x;
             return temp;
 }

 void display()
{
              for(int i=0 ; i< 10; i++)
{
                        node * temp=new node;
                        temp=HashTable[i];
                        cout<<"a["<<i<<"] : ";
                        while(temp !=NULL)
{
                                  cout<<" ->"<<temp->value;
                                  temp=temp->next;
                         }
                        cout<<"\n";
```

```
}
 }

int searchElement(int value)
{
            bool flag = false;
          int hash_val = HashFunction(value);
          node* entry = HashTable[hash_val];
          cout<<"\nElement found at : ";
          while (entry != NULL)
            {
                    if (entry->value==value)
                     {
                              cout<<hash_val<<" : "<<entry->value<<endl;
                              flag = true;
                     }
                    entry = entry->next;
          }
          if (!flag)
          return -1;
 }

 void deleteElement(int value)
{
            int hash_val = HashFunction(value);
          node* entry = HashTable[hash_val];
          if (entry == NULL )
          {
            cout<<"No Element found ";
                      return;
          }

          if(entry->value==value)
{
           HashTable[hash_val]=entry->next;
           return;
          }
          while ((entry->next)->value != value)
           {
             entry = entry->next;
          }
          entry->next=(entry->next)->next;
 }

 void insertElement(int value)
{
            int hash_val = HashFunction(value);


          node* temp=new node;
```

```
        node* head=new node;
        head = create_node(value);
        temp=HashTable[hash_val];
        if (temp == NULL)
                {
              HashTable[hash_val] =head;
               }
        else
{
          while (temp->next != NULL)
         {
                 temp = temp->next;
                 }
                temp->next =head;
        }
 }
};
int main()
{
         int ch;
          int data,search,del;
         hashing h;
         do
{
cout<<"\nTelephone : \n1.Insert \n2.Display \n3.Search \n4.Delete \n5.Exit \n";
cout<<"\n Enter your choice:";
                    cin>>ch;
                    switch(ch)
{
                              case 1:cout<<"\nEnter phone no. to be inserted : ";
                                        cin>>data;
h.insertElement(data);
                                         break;
                              case 2:h.display();
                                        break;
                              case 3:cout<<"\nEnter the no to be searched : ";
                                        cin>>search;

                                         if (h.searchElement(search) == -1)
                                        {
                                                  cout<<"No element found at key ";
                                                   continue;
                                        }
                                        break;
                              case 4:cout<<"\nEnter the phno. to be deleted : ";
                                         cin>>del;
                                         h.deleteElement(del);
                                          cout<<"Phno. Deleted"<<endl;
                                        break;
                                        }
```

```
                    }while(ch!=5);
                    return 0;
}
```

Output:

Telephone :

1.Insert

2.Display

3.Search

4.Delete

5.Exit

Enter your choice:1

Enter phone no. to be inserted : 15

Enter your choice:1

Enter phone no. to be inserted : 66

Enter your choice:1

Enter phone no. to be inserted : 89

Enter your choice:1

Enter phone no. to be inserted : 52

Enter your choice:1

Enter phone no. to be inserted : 75

Telephone :

1.Insert

2.Display

3.Search

4.Delete

5.Exit

Enter your choice:2

a[0] :

a[1] :

a[2] : ->52

a[3] :

a[4] :

a[5] : ->15 ->75

a[6] : ->66

a[7] :

a[8] :

a[9] : ->89

Telephone :

1.Insert

2.Display

3.Search

4.Delete

5.Exit

Enter your choice:3

Enter the no to be searched : 66

Element found at : 6 : 66

Telephone :

1.Insert

2.Display

3.Search

4.Delete

5.Exit

Enter your choice:4

Enter the phno. to be deleted : 75

Phno. Deleted

Telephone :

1.Insert

2.Display

3.Search

4.Delete

5.Exit

Enter your choice:2

a[0] :

a[1] :

a[2] : ->52

a[3] :

a[4] :

a[5] : ->15

a[6] : ->66

a[7] :

a[8] :

a[9] : ->89

Telephone :

1.Insert

2.Display

3.Search

4.Delete

5.Exit

Enter your choice:5

Process exited after 102.8 seconds with return value 0

Press any key to continue . . .