**Consider the telephone book database of N clients. Make use of a hash table implementation to quickly look up the client‘s telephone number.**

#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\* prev = NULL;

//node\* entry = HashTable[hash\_val];

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";

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;

}