```
/*
A1-Consider telephone book database of N clients. Make use of a hash table implementation
to quickly look up client‘s telephone number. Make useof two collision handling techniques
and compare them using number of comparisons required to find a set
of telephone numbers
*/

#include <iostream>
using namespace std;

// Store details : Node-> Key Name Telephone
class node{
   private:
      string name;
      string telephone;
      int key;
   public:
      node(){
         key=0;
      }
      friend class hashing;   // To access the private members of class node
};

// Hashng Fuction that generates different key value
// Sum of ascii value of each character in string
int ascii_generator(string s){
   int sum=0;
   for (int i = 0; s[i] != '\0'; i++)
      sum = sum + s[i];
   return sum%100;
}

// Class -> Hashing
class hashing{
   private:
      node data[100]; // Size of directory -> 100
      string n;
      string tele;
      int k, index;
      int size=100;
   public:
      hashing(){
         k=0;
      }

      // Function to create record
      void create_record(string n,string tele){
         k=ascii_generator(n);   //using ascii value of string as key
         index=k%size;
         for (int j=0;j<size;j++){
            if(data[index].key==0){
               data[index].key=k;
               data[index].name=n;
               data[index].telephone=tele;
               break;
            }
            else
```

```
            index=(index+1)%size;
        }
    }

    // Function to search for record based on name input
    void search_record(string name){
        int index1,k,flag=0;
        k=ascii_generator(name);
        index1=k%size;

        for(int a=0;a<size;a++){
            if(data[index1].key==k){
                flag=1;
                cout<<"\nRecord found\n";
                cout<<"Name :: "<<data[index1].name<<endl;
                cout<<"Telephone :: "<<data[index1].telephone<<endl;
                break;
            }
            else
                index1=(index1+1)%size;
        }
        if(flag==0)
            cout<<"Record not found";
    }

    // Function to delete existing record
    void delete_record(string name){
        int index1,key,flag=0;
        key=ascii_generator(name);
        index1=key%size;

        for(int a=0;a<size;a++){
            if(data[index].key==key){
                flag=1;
                data[index1].key=0;
                data[index1].name=" ";
                data[index1].telephone=" ";
                cout<<"\nRecord Deleted successfully"<<endl;
                break;
            }
            else
                index1=(index1+1)%size;
        }
        if(flag==0)
            cout<<"\nRecord not found";
    }

    // Function to update existing record
    void update_record(string name){
        int index1,key,flag=0;
        key=ascii_generator(name);
        index1=key%size;

        for(int a=0;a<size;a++){
            if(data[index1].key==key){
                flag=1;
                break;
```

```
            }
            else
               index1=(index1+1)%size;
         }

         if(flag==1){
            cout<<"Enter the new telephone number :: ";
            cin>>tele;
            data[index1].telephone=tele;
            cout<<"\nRecord Updated successfully";
         }
      }

      // Function to display the directory
      void display_record(){
         cout<<"\t  Name  \t\t Telephone";
         for (int a = 0; a < size; a++) {
            if(data[a].key!=0){
               cout<<"\n\t"<<data[a].name<<" \t\t\t "<<data[a].telephone;
            }
         }
      }
};

// Main Function
int main(){
   hashing s;
   string name;
   string telephone;
   int choice,x;
   bool loop=1;
   // Menu driven code
   while(loop){
      cout<<"\n-------------------------"<<endl
         <<" Telephone book Database "<<endl
         <<"-------------------------"<<endl
         <<"1. Create Record"<<endl
         <<"2. Display Record"<<endl
         <<"3. Search Record"<<endl
         <<"4. Update Record"<<endl
         <<"5. Delete Record"<<endl
         <<"6. Exit"<<endl
         <<"Enter choice :: ";
      cin>>choice;
      switch (choice)
      {
      case 1:
         cout<<"\nEnter name :: ";
         cin>>name;
         cout<<"Enter Telephone number :: ";
         cin>>telephone;
         s.create_record(name,telephone);
         break;

      case 2:
         s.display_record();
         break;
```

```
        case 3:
            cout<<"\nEnter the name :: ";
            cin>>name;
            s.search_record(name);
            break;

        case 4:
            cout<<"\nEnter the name :: ";
            cin>>name;
            s.update_record(name);
            break;

        case 5:
            cout<<"\nEnter name to Delete :: ";
            cin>>name;
            s.delete_record(name);
            break;

        case 6:
            loop=0;
            break;

        default:
            cout<<"\nYou Entered something wrong!";
            break;
        }
    }
    return 0;
}
```