**Practical GroupA\_1**

**Aim:**

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

**Code:**

class Node:

def \_\_init\_\_(self):

self.name = ""

self.telephone = ""

self.key = 0

class Hashing:

def \_\_init\_\_(self):

self.data = [Node() for \_ in range(100)]

self.size = 100

def ascii\_generator(self, s):

total = 0

for char in s:

total += ord(char)

return total % 100

def create\_record(self, name, telephone):

k = self.ascii\_generator(name)

index = k % self.size

for \_ in range(self.size):

if self.data[index].key == 0:

self.data[index].key = k

self.data[index].name = name

self.data[index].telephone = telephone

break

else:

index = (index + 1) % self.size

def search\_record(self, name):

k = self.ascii\_generator(name)

index = k % self.size

for \_ in range(self.size):

if self.data[index].key == k:

print("\nRecord found")

print("Name ::", self.data[index].name)

print("Telephone ::", self.data[index].telephone)

break

else:

index = (index + 1) % self.size

else:

print("Record not found")

def delete\_record(self, name):

key = self.ascii\_generator(name)

index = key % self.size

for \_ in range(self.size):

if self.data[index].key == key:

self.data[index].key = 0

self.data[index].name = ""

self.data[index].telephone = ""

print("\nRecord deleted successfully")

break

else:

index = (index + 1) % self.size

else:

print("Record not found")

def update\_record(self, name):

key = self.ascii\_generator(name)

index = key % self.size

for \_ in range(self.size):

if self.data[index].key == key:

print("Enter the new telephone number :: ")

self.data[index].telephone = input()

print("\nRecord updated successfully")

break

else:

index = (index + 1) % self.size

else:

print("Record not found")

def display\_record(self):

print("\tName\t\tTelephone")

for a in range(self.size):

if self.data[a].key != 0:

print("\t" + self.data[a].name + "\t\t" + self.data[a].telephone)

if \_\_name\_\_ == "\_\_main\_\_":

s = Hashing()

loop = True

while loop:

print("\n-------------------------")

print(" Telephone book Database ")

print("-------------------------")

print("1. Create Record")

print("2. Display Record")

print("3. Search Record")

print("4. Update Record")

print("5. Delete Record")

print("6. Exit")

print("Enter choice :: ")

choice = int(input())

if choice == 1:

print("Enter name :: ")

name = input()

print("Enter Telephone number :: ")

telephone = input()

s.create\_record(name, telephone)

elif choice == 2:

s.display\_record()

elif choice == 3:

print("Enter the name :: ")

name = input()

s.search\_record(name)

elif choice == 4:

print("Enter the name :: ")

name = input()

s.update\_record(name)

elif choice == 5:

print("Enter name to Delete :: ")

name = input()

s.delete\_record(name)

elif choice == 6:

loop = False

else:

print("\nYou entered something wrong!")