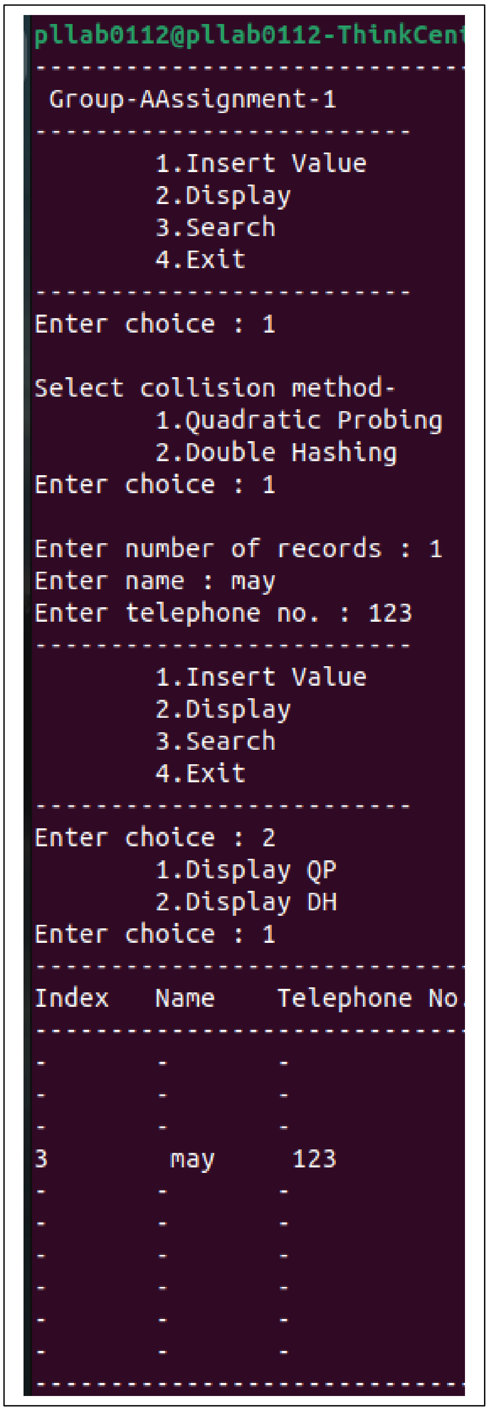
Assignment 1

Mayur Zope SE Comp A 75

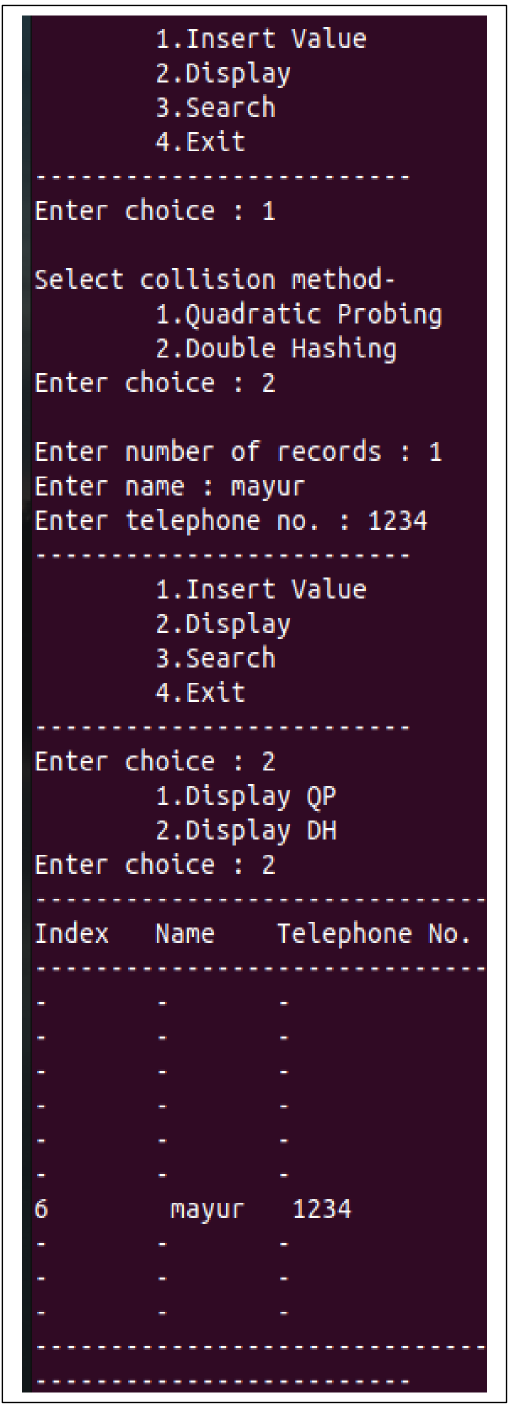
Consider telephone book database of N clients. Make use of a hash table implementation to quickly look up client's telephone number. Make use of two collision handling techniques and compare them using number of comparisons required to find a set of telephone numbers

```
class TelephoneBook:
    def __init__(self, name, tel_no):
        self.name = name
        self.tel_no = tel_no


#*************************************************************************************
def Insertion_QuadProbing():
    hashtable = [None for i in range(10)]
    num_records = int(input("\nEnter number of records : "))
    j = 1
    for i in range(num_records):
        n = input("Enter name : ")
        t = int(input("Enter telephone no. : "))
        hashValue = t % 10  # hash function
        if hashtable[hashValue] is None:
            hashtable[hashValue] = TelephoneBook(n, t)  # creating obj of class and inserting into hashtable
        elif hashtable[hashValue] is not None:
            hashValue = (hashValue + (j * j)) % 10
            hashtable[hashValue] = TelephoneBook(n, t)
            j += 1
    return hashtable


#*************************************************************************************
def Insertion_DoubleHashing():
    hashtable = [None for i in range(10)]
    num_records = int(input("\nEnter number of records : "))
    j = 2
    for i in range(num_records):
        n = input("Enter name : ")
        t = int(input("Enter telephone no. : "))
        hashvalue = t % 9 + 7 - (t % 7)  # finding hashvalue using 2 hash functions 1) key%9
        # 2) 7-(key%7)
        if hashtable[hashvalue] is None:  # Check if the slot is empty
            hashtable[hashvalue] = TelephoneBook(n, t)
        elif hashtable[hashvalue] is not None:
            hashvalue = t % 9 + j * (7 - (t % 7))
            j += 1
    return hashtable

#****************************************************************************************************
def Display_QP(hash1):
    print("------------------------------")
    print("Index\tName\tTelephone No.")
    print("------------------------------")
    for obj in hash1:
        if(obj is None):
            print("-\t-\t-")
        if (obj is not None):
            print(hash1.index(obj), "\t", obj.name, "\t", obj.tel_no)
    print("------------------------------")

#****************************************************************************************************
def Display_DH(hash2):
    print("------------------------------")
    print("Index\tName\tTelephone No.")
    print("------------------------------")
    for obj in hash2:
        if(obj is None):
            print("-\t-\t-")
        if (obj is not None):
            print(hash2.index(obj), "\t", obj.name, "\t", obj.tel_no)
    print("-------------------------------")

#**********************************************************************************
def Search(hash1, hash2):
    n = input("Enter name to search: ")
    f1 = 0
    f2 = 0
    for obj in hash1:
        if(obj is None):
            continue
        if obj.name == n:
            print("\nFound in Hashtable-1 !")
            print("-------------------------------")
            print("Index\tName\tTelephone No.")
            print("-------------------------------")
            print(hash1.index(obj), "\t", obj.name, "\t", obj.tel_no)
            print("-------------------------------")
            f1 = 1
    for obj in hash2:
        if(obj is None):
            continue
        if obj.name == n:
            print("\nFound in Hashtable-2 !")
            print("-------------------------------")
            print("Index\tName\tTelephone No.")
            print("-------------------------------")
            print(hash2.index(obj), "\t", obj.name, "\t", obj.tel_no)
            print("-------------------------------")
            f2 = 1
    if f1 == 0 and f2 == 0:
        print("\nNot found !!!\n")

#*************************************************************************************
def main():
    # initialising hashtables to "None"
    hash1 = [None for i in range(10)]
    hash2 = [None for i in range(10)]
    print("-------------------------------")
    print(" Group-AAssignment-1")
    while True:
        print("------------------------")
        print("\t1.Insert Value")
        print("\t2.Display")
        print("\t3.Search")
        print("\t4.Exit")
        print("------------------------")
        ch = int(input("Enter choice : "))
        if ch == 1:
            print("\nSelect collision method-")
            print("\t1.Quadratic Probing")
            print("\t2.Double Hashing")
            c = int(input("Enter choice : "))
            if c == 1:
                hash1 = Insertion_QuadProbing()
            elif c == 2:
                hash2 = Insertion_DoubleHashing()
        elif ch == 2:
            print("\t1.Display QP")
            print("\t2.Display DH")
            c1 = int(input("Enter choice : "))
            if c1 == 1:
                Display_QP(hash1)  # To display hashtable which uses quadratic probing collision method
            else:
                Display_DH(hash2)  # To display hashtable which uses double hashing collision method
        elif ch == 3:
            Search(hash1, hash2)
        elif ch == 4:
            quit()
        else:
            print("! Enter valid choice.")


# Start the program
main()
```

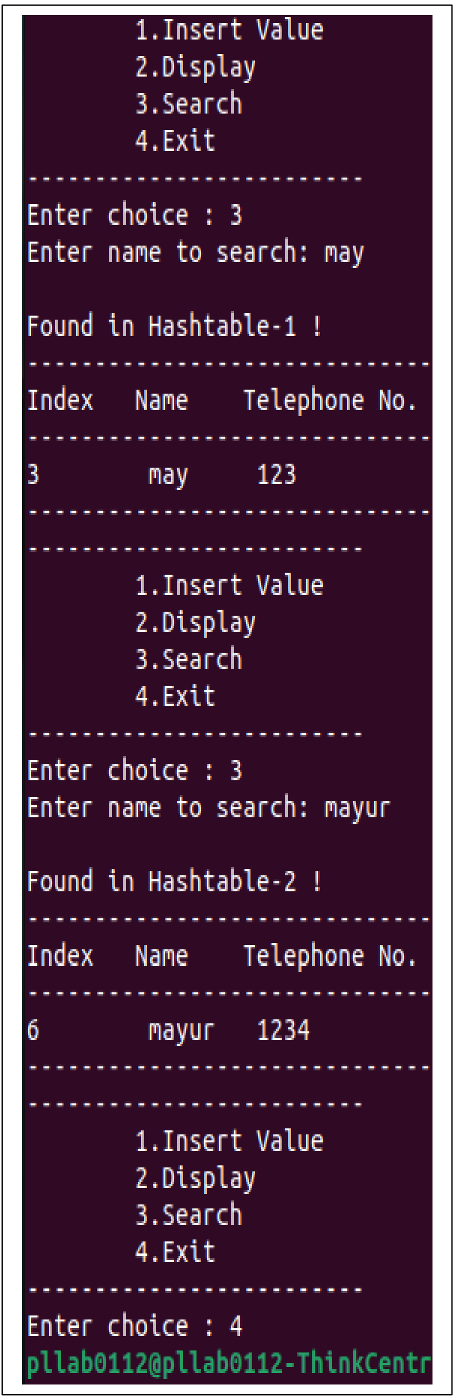
//Output



```
pllab0112@pllab0112-ThinkCent
-----------------------------
 Group-AAssignment-1
-------------------------
        1.Insert Value
        2.Display
        3.Search
        4.Exit
-------------------------
Enter choice : 1

Select collision method-
        1.Quadratic Probing
        2.Double Hashing
Enter choice : 1

Enter number of records : 1
Enter name : may
Enter telephone no. : 123
-------------------------
        1.Insert Value
        2.Display
        3.Search
        4.Exit
-------------------------
Enter choice : 2
        1.Display QP
        2.Display DH
Enter choice : 1
-----------------------------
Index   Name    Telephone No.
-----------------------------
-       -       -
-       -       -
-       -       -
3        may     123
-       -       -
-       -       -
-       -       -
-       -       -
-       -       -
-       -       -
-----------------------------
```



```
        1.Insert Value
        2.Display
        3.Search
        4.Exit
-------------------------
Enter choice : 1

Select collision method-
        1.Quadratic Probing
        2.Double Hashing
Enter choice : 2

Enter number of records : 1
Enter name : mayur
Enter telephone no. : 1234
-------------------------
        1.Insert Value
        2.Display
        3.Search
        4.Exit
-------------------------
Enter choice : 2
        1.Display QP
        2.Display DH
Enter choice : 2
------------------------------
Index   Name    Telephone No.
------------------------------
-       -       -
-       -       -
-       -       -
-       -       -
-       -       -
-       -       -
6        mayur   1234
-       -       -
-       -       -
-       -       -
------------------------------
-------------------------
```



```
        1.Insert Value
        2.Display
        3.Search
        4.Exit
-------------------------
Enter choice : 3
Enter name to search: may

Found in Hashtable-1 !
------------------------------
Index   Name    Telephone No.
------------------------------
3        may     123
------------------------------
-------------------------
        1.Insert Value
        2.Display
        3.Search
        4.Exit
-------------------------
Enter choice : 3
Enter name to search: mayur

Found in Hashtable-2 !
------------------------------
Index   Name    Telephone No.
------------------------------
6        mayur   1234
------------------------------
-------------------------
        1.Insert Value
        2.Display
        3.Search
        4.Exit
-------------------------
Enter choice : 4
pllab0112@pllab0112-ThinkCentr
```