**Code**

size = 10

class LinearProbing:

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

self.hash = [0] \* size

self.flag = [0] \* size

def insert(self, x):

loc = x % size

for i in range(size):

if self.flag[loc] == 0:

self.hash[loc] = x

self.flag[loc] = 1

break

else:

loc = (loc + 1) % size

def create(self):

n = int(input("Enter the number of phone numbers to be inserted:\n"))

print("Enter the Phone numbers to be inserted in Hash table ->")

for i in range(n):

x = int(input())

self.insert(x)

def find(self, x):

loc = x % size

for i in range(size):

if self.flag[loc] == 1 and self.hash[loc] == x:

return loc

else:

loc = (loc + 1) % size

return -1

def search(self):

x = int(input("Enter the phone number to be searched:\n"))

loc = self.find(x)

if loc == -1:

print("Phone number is not found")

else:

print(f"Phone number {x} is found at {loc}th location")

def print(self):

print("Hash Table is ->")

for i in range(size):

print(f"({i}) -> ", end="")

if self.flag[i] == 1:

print(self.hash[i])

else:

print("----")

class Node:

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

self.phone = phone

self.next = None

class Chaining:

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

self.hash = [None] \* size

def create(self):

n = int(input("Enter the number of phone numbers to be inserted:\n"))

print("Enter the Phone numbers to be inserted in Hash table ->")

for i in range(n):

x = int(input())

self.insert(x)

def insert(self, key):

loc = key % size

p = Node(key)

if self.hash[loc] == None:

self.hash[loc] = p

else:

q = self.hash[loc]

while q.next != None:

q = q.next

q.next = p

def display(self):

print("Hash Table is ->")

for i in range(size):

q = self.hash[i]

print(f"({i}) -> ", end="")

while q:

print(q.phone, end="")

if q.next:

print(" -> ", end="")

q = q.next

print()

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

print("----------------- Linear Probing -----------------")

lp = LinearProbing()

lp.create()

lp.print()

lp.search()

print("----------------- Separate Chaining -----------------")

h = Chaining()

h.create()

h.display()