class Record:

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

self.\_name = None

self.\_number = None

def get\_name(self):

return self.\_name

def get\_number(self):

return self.\_number

def set\_name(self, name):

self.\_name = name

def set\_number(self, number):

self.\_number = number

def \_\_str\_\_(self):

record = "Name: " + str(self.get\_name()) + "\t" + "\tNumber: " + str(self.get\_number())

return record

class doubleHashTable:

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

self.size = int(input("Enter the Size of the hash table : "))

self.table = [None] \* self.size

self.elementCount = 0

self.comparisons = 0

def isFull(self):

return self.elementCount == self.size

def h1(self, element):

return element % self.size

def h2(self, element):

return 5 - (element % 5)

def doubleHashing(self, record):

posFound = False

limit = self.size

i = 1

while i <= limit:

newPosition = (self.h1(record.get\_number()) + i \* self.h2(record.get\_number())) % self.size

if self.table[newPosition] == None:

posFound = True

break

else:

i += 1

return posFound, newPosition

def insert(self, record):

if self.isFull():

print("Hash Table Full")

return False

posFound = False

position = self.h1(record.get\_number())

if self.table[position] == None:

self.table[position] = record

print("Phone number of " + record.get\_name() + " is at position " + str(position))

self.elementCount += 1

return True

else:

print("Collision has occurred for " + record.get\_name() + "'s phone number at position " + str(

position) + " finding new Position.")

while not posFound:

posFound, position = self.doubleHashing(record)

if posFound:

self.table[position] = record

self.elementCount += 1

print("Phone number of " + record.get\_name() + " is at position " + str(position))

return posFound

def search(self, record):

found = False

position = self.h1(record.get\_number())

self.comparisons += 1

if self.table[position] != None:

if self.table[position].get\_name() == record.get\_name():

print("Phone number found at position {} and total comparisons are 1".format(position))

return position

else:

limit = self.size

i = 1

newPosition = position

while i <= limit:

position = (self.h1(record.get\_number()) + i \* self.h2(record.get\_number())) % self.size

self.comparisons += 1

if self.table[position] != None:

if self.table[position].get\_name() == record.get\_name():

found = True

break

elif self.table[position].get\_name() == None:

found = False

break

else:

i += 1

if found:

print("Phone number found at position {} and total comparisons are {}".format(position, i + 1))

else:

print("Record not Found")

return found

def display(self):

print("\n")

for i in range(self.size):

print("Hash Value: " + str(i) + "\t\t" + str(self.table[i]))

print("The number of phonebook records in the Table are : " + str(self.elementCount))

class hashTable:

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

self.size = int(input("Enter the Size of the hash table : "))

self.table = [None] \* self.size

self.elementCount = 0

self.comparisons = 0

def isFull(self):

return self.elementCount == self.size

def hashFunction(self, element):

return element % self.size

def insert(self, record):

if self.isFull():

print("Hash Table Full")

return False

position = self.hashFunction(record.get\_number())

if self.table[position] == None:

self.table[position] = record

print("Phone number of " + record.get\_name() + " is at position " + str(position))

self.elementCount += 1

return True

else:

print("Collision has occurred for " + record.get\_name() + "'s phone number at position " + str(

position) + " finding new Position.")

while self.table[position] != None:

position += 1

if position >= self.size:

position = 0

self.table[position] = record

print("Phone number of " + record.get\_name() + " is at position " + str(position))

self.elementCount += 1

return True

def search(self, record):

found = False

position = self.hashFunction(record.get\_number())

self.comparisons += 1

if self.table[position] != None:

if self.table[position].get\_name() == record.get\_name() and self.table[position].get\_number() == record.get\_number():

found = True

print("Phone number found at position {} and total comparisons are 1".format(position))

return position

else:

position += 1

if position >= self.size - 1:

position = 0

while self.table[position] != None or self.comparisons <= self.size:

if self.table[position].get\_name() == record.get\_name() and self.table[position].get\_number() == record.get\_number():

found = True

print("Phone number found at position {} and total comparisons are {}".format(position, self.comparisons))

return position

position += 1

if position >= self.size - 1:

position = 0

self.comparisons += 1

if not found:

print("Record not found")

return False

def display(self):

print("\n")

for i in range(self.size):

print("Hash Value: " + str(i) + "\t\t" + str(self.table[i]))

print("The number of phonebook records in the Table are : " + str(self.elementCount))

def input\_record():

record = Record()

name = input("Enter Name:")

number = int(input("Enter Number:"))

record.set\_name(name)

record.set\_number(number)

return record

choice1 = 0

while choice1 != 3:

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("1. Linear Probing \*")

print("2. Double Hashing \*")

print("3. Exit \*")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

choice1 = int(input("Enter Choice: "))

if choice1 > 3:

print("Please Enter Valid Choice")

if choice1 == 1:

h1 = hashTable()

choice2 = 0

while choice2 != 4:

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("1. Insert \*")

print("2. Search \*")

print("3. Display \*")

print("4. Back \*")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

choice2 = int(input("Enter Choice"))

if choice2 > 4:

print("Please Enter Valid Choice")

if choice2 == 1:

record = input\_record()

h1.insert(record)

elif choice2 == 2:

record = input\_record()

h1.search(record)

elif choice2 == 3:

h1.display()

elif choice1 == 2:

h2 = doubleHashTable()

choice2 = 0

while choice2 != 4:

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("1. Insert \*")

print("2. Search \*")

print("3. Display \*")

print("4. Back \*")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

choice2 = int(input("Enter Choice"))

if choice2 > 4:

print("Please Enter Valid Choice")

if choice2 == 1:

record = input\_record()

h2.insert(record)

elif choice2 == 2:

record = input\_record()

h2.search(record)

elif choice2 == 3:

h2.display()