class hasht:

arry = []

for i in range(10):

a = []

arry.append(a)

def insert(self, no):

if (len(self.arry[no % 10]) == 0):

self.arry[no % 10].append(no)

else:

print("collision occur chose which collision resulution technique you want to use \n1] open chaning \n2] linear probing")

v = int(input())

if(v == 1):

self.arry[no % 10].append(no)

if(v == 2):

N = no % 10

T = N + 10

N = N + 1

while(len(self.arry[N % 10])) != 0 and T > N:

N = N + 1

if (T == N):

print("Hash table is full!!!")

else:

self.arry[N % 10].append(no)

def printhash(self):

for i in range(len(self.arry)):

print(i, ") ", end="")

if (len(self.arry[i]) == 0):

print(" ----- ")

else:

for j in self.arry[i]:

print(j, " ", end="")

print(" ")

def delm(self):

print("Enter the no you want to delet")

dn = int(input())

indx = dn % 10

u = indx

find = 0

for z in range(len(self.arry[indx])):

if self.arry[indx][z] == dn:

self.arry[indx].pop(z)

return

for k in range(10):

if len(self.arry[k]) == 1:

if self.arry[k][0] == dn:

self.arry[k].pop(0)

return

print("No not found!!")

def fd(self):

print("Enter the no you want to find")

dn = int(input())

indx = dn % 10

u = indx

find = 0

for z in range(len(self.arry[indx])):

if self.arry[indx][z] == dn:

print("No found!!")

return

for k in range(10):

if len(self.arry[k]) == 1:

if self.arry[k][0] == dn:

print("No found!!")

return

print("No not found!!")

p1 = hasht()

c = -1

while c != 5:

print("Enter 1 to add the no \nEnter 2 to display hashing table \nEnter 3 to delet \nEnter 4 to find \nEnter 5 to exit")

c = int(input())

if c == 1:

print("Enter the no you want to insert")

num = int(input())

p1.insert(num)

if c == 2:

p1.printhash()

if c == 3:

p1.delm()

if c == 4:

p1.fd()