

INPUT:

```
max=int(input("Enter hash table size: "))
ht = [-1]*max
ht1 = [-1]*max
print(ht)
keys=[]
```

```
def LinearProbe(k):
    pos=k%max
    if(ht[pos]==-1):
        ht[pos]=k
    else:
        while(ht[pos]!=-1):
            pos=(pos+1)%max
        ht[pos]=k
```

```
def QuadraticProbe(k):
    pos=k%max
    if(ht1[pos]==-1):
        ht1[pos]=k
    else:
        i=1
        while True:
            newpos=(pos+(i*i))%max
            if(ht1[newpos]==-1):
                ht1[newpos]=k
                break
            i=i+1
```

```
def insert():
    print("\tPerforming Linear Probing: ")
    print("\nEnter the keys below: ")
    ans=1
    cnt=1
    while(ans==1):
        key=int(input("Enter key: "))
        keys.append(key)
        LinearProbe(key)
        print(ht)
        cnt=cnt+1
        if(cnt>max):
            print("Hash table is full!")
            print(ht)
            break
        ans=int(input("Enter more?(1/0) "))

    print("\tPerforming Quadratic Probing on same key set: ")
    print("keys: ", keys)
    for i in keys:
        QuadraticProbe(i)
    print(ht1)
```

```

def Lsearch(k):
    pos=k%max
    global lcmp
    lcmp=0
    lcmp+=1
    if(ht[pos]==k):
        return pos
    else:
        loc=(pos+1)%max
        for i in range(max-1):
            lcmp+=1
            if(ht[loc]==k):
                return loc
            loc=(loc+1)%max
        return -1

def Qsearch(k):
    pos=k%max
    global qcmp
    qcmp=0
    qcmp+=1
    if(ht1[pos]==k):
        return pos
    else:
        stepcount=1
        for i in range(max-1):
            newpos=(pos+(stepcount*stepcount))%max
            qcmp+=1
            if(ht1[newpos]==k):
                return newpos
            stepcount+=1
        return -1

def menu():
    while True:
        #os.system('clear')
        print("*****Menu*****")
        print("1.Insert")
        print("2.Display")
        print("3.Search")
        print("4.Exit")

        n=int(input("Enter your choice: "))
        if(n==1):
            global ht, ht1, keys
            ht=[-1]*max
            ht1=[-1]*max
            keys=[]
            insert()
        elif(n==2):

```

```

        print("\nUsing Linear probing: ",ht)
        print("\nUsing Quadratic probing: ",ht1)
    elif(n==3):
        print("\nUsing Linear probing: ",ht)
        print("\nUsing Quadratic probing: ",ht1)
        k=int(input("\nEnter key to search: "))
        print("\n\tLinear search: ")
        location=Lsearch(k)
        if(location== -1):
            print("Element not found")
        else:
            print("Element found at index: ", location, "with ", lcmp, "number of comparison(s).")
        if(location!= -1):
            location=Qsearch(k)
            print("\n\tQuadratic search below:")
            print("Element found at index: ", location, "with", qcmp, "number of comparison(s)")
    elif(n==4):
        print("\n\t\tProgram Ends.")
        break
    else:
        print("\t\tInvalid input!")

```

menu()

OUTPUT:

```

Enter hast table size: 10
[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
*****Menu*****
1.Insert
2.Display
3.Search
4.Exit
Enter your choice: 1
    Performing Linear Probing:

Enter the keys below:
Enter key: 11
[-1, 11, -1, -1, -1, -1, -1, -1, -1, -1]
Enter more?(1/0) 1
Enter key: 22
[-1, 11, 22, -1, -1, -1, -1, -1, -1, -1]
Enter more?(1/0) 1
Enter key: 55
[-1, 11, 22, -1, -1, 55, -1, -1, -1, -1]
Enter more?(1/0) 1
Enter key: 21
[-1, 11, 22, 21, -1, 55, -1, -1, -1, -1]
Enter more?(1/0) 1
Enter key: 31
[-1, 11, 22, 21, 31, 55, -1, -1, -1, -1]
Enter more?(1/0) 1
Enter key: 89
[-1, 11, 22, 21, 31, 55, -1, -1, -1, 89]
Enter more?(1/0) 1
Enter key: 99

```

[99, 11, 22, 21, 31, 55, -1, -1, -1, 89]

Enter more?(1/0) 1

Enter key: 44

[99, 11, 22, 21, 31, 55, 44, -1, -1, 89]

Enter more?(1/0) 0

Performing Quadratic Probing on same key set:

keys: [11, 22, 55, 21, 31, 89, 99, 44]

[-1, 11, -1, -1, -1, -1, -1, -1, -1, -1]

[-1, 11, 22, -1, -1, -1, -1, -1, -1, -1]

[-1, 11, 22, -1, -1, 55, -1, -1, -1, -1]

[21, 11, 22, -1, -1, 55, -1, -1, -1, -1]

[21, 11, 22, -1, -1, 55, -1, 31, -1, -1]

[21, 11, 22, -1, -1, 55, -1, 31, -1, 89]

[21, 11, 22, 99, -1, 55, -1, 31, -1, 89]

[21, 11, 22, 99, 44, 55, -1, 31, -1, 89]

*****Menu*****

1.Insert

2.Display

3.Search

4.Exit

Enter your choice: 3

Using Linear probing: [99, 11, 22, 21, 31, 55, 44, -1, -1, 89]

Using Quadratic probing: [21, 11, 22, 99, 44, 55, -1, 31, -1, 89]

Enter key to search: 99

Linear search:

Element found at index: 0 with 2 number of comparison(s).

Quadratic search below:

Element found at index: 3 with 3 number of comparison(s)

*****Menu*****

1.Insert

2.Display

3.Search

4.Exit

Enter your choice: 4

Program Ends.