Circular Linked list

class element:

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

self.dt=v

self.next=None

class cll:

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

self.head=None

self.count=0

def add(self,val):

newnode=element(val)

newnode.next=self.head

if self.head is None:

self.head=newnode

self.count=self.count+1

else:

i=1

tmp=self.head

while i<self.count:

tmp=tmp.next

i=i+1

tmp.next=newnode

self.count=self.count+1

val=int(input('Enter the element to be removed'))

tmp=self.head

i=1

if tmp.dt==val:

nn=tmp.next

while tmp.next!=self.head:

tmp=tmp.next

tmp.next=nn

self.head=nn

self.count=self.count-1

else:

while i<self.count:

if val==tmp.next.dt:

print("number found")

tmp.next=tmp.next.next

self.count=self.count-1

break

tmp=tmp.next

i=i+1

def display(self):

tmp=self.head

i=1

while i<=self.count:

print(tmp.dt)

i=i+1

tmp=tmp.next

def search(self):

val=int(input('enter the number to be searched'))

tmp=self.head

i=1

f=0

while i<self.count:

if val==tmp.dt:

print("number found")

f=0

break

else:

f=1

i=i+1

tmp=tmp.next

if f==1:

print('number not found')

#main

cir=cll()

cir.add(10)

cir.add(20)

cir.add(30)

cir.add(40)

cir.add(50)

cir.display()

cir.search()

cir.remove()

cir.display()