作业2

单链表类定义

import math as math

class LNode:

def \_\_init\_\_(self,elem,next\_=None):

self.elem=elem

self.next=next\_

class LinkedListUnderflow(ValueError):

pass

class LList:

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

self.\_head = None

self.\_tail = None

def is\_empty(self):

return self.\_head is None

def prepend(self,elem):

self.\_head=LNode(elem,self.\_head)

def append(self,elem,add=None):

if self.\_head is None:

self.\_head=LNode(elem)

self.\_tail=self.\_head

return

p=self.\_head

while p.next is not None:

p=p.next

if add is not None:

p.next=LNode(elem,add)

else:

p.next=LNode(elem)

self.\_tail=p.next

#if add is not None:

# p.next=LNode(elem)

#else:

def shift(self):

if self.\_head is None:

raise LinkedListUnderflow("in shift")

p=self.\_head

#e=self.\_head.elem

self.\_head=self.\_head.next

return p

def pop(self):

if self.\_head is None:

raise LinkedListUnderflow("in pop")

p=self.\_head

while p.next.next is not None:

p=p.next

q=p.next

q.next=None

p.next = None

return q.elem

def printall(self):

p=self.\_head

A=[]

while p is not None:

A.append(p.elem)

p=p.next

print A

第一题：

代码

def ration(N,M):

x=N/M

y=N%M

CList=LList()

CList.append(x)

B=[y]

while True:

x=y\*10/M

y=(y\*10)%M

B.append(y)

if y==B[0]:

#CList.append(x) 打印结果时采用该行

CList.append(x,CList.\_head.next)

return CList

else:

CList.append(x)

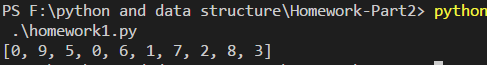
N=77

M=81

klist=ration(N,M)

klist.printall()

结果：



第二题/第三题

代码：

def IsCircle(CList):

f=s=CList.\_head

i=0

while f is not None and f.next is not None:

if i==0:

f=f.next.next

elif i==1:

f=f.next

s=s.next

if f is s:

if i==0:

i+=1

f=CList.\_head

print "This list has a circle"

else:

return f.elem #入口结点值

return "This list does not have a circle"

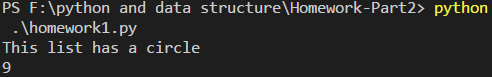
N=77

M=81

klist=ration(N,M)

print IsCircle(klist) #判断时输出入口

结果：



第四题：

代码

def Change(List):

p=List.\_head

while p is not None and p.next is not None:

q=LNode(List.pop())

temp=p.next

p.next=q

q.next=temp

p=temp

List.printall()

lst=LList()

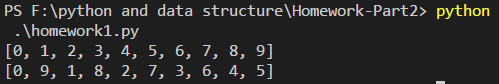
for i in range(10):

lst.append(i)

lst.printall()

Change(lst)

结果：



第五题

代码：

def main(List):

    B=0

    for i in List:

        if B==0:

            A=i

            B=1

        elif i == A:

            B+=1

        elif i != A:

            B-=1

    if B>1:

        return A

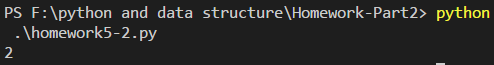
    else:

        return -1

List=[1,2,3,1,2,3,2,3,2,1,2,1,1,2,2,2]

print main(List)

结果：



第六题：

代码：

def find(List):

    C,D=0,0

    for i in range(len(List)):

        if C==0:

            A=List[i]

            C=1

        elif D==0 and A != List[i]:

            B=List[i]

            D=1

        elif A==List[i]:

            C+=1

        elif B==List[i]:

            D+=1

        elif A!=List[i] and B!=List[i]:

            C-=1

            D-=1

    if C>1 and D>1:

        return A,B

    elif C>1 and D<=1:

        return A

    elif D>1 and C<=1:

        return B

    else:

        return -1

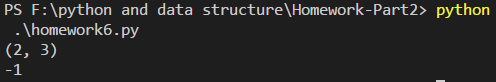
L1=[1,2,3,2,1,2,3,3,3,2]

L2=[1,2,3,1,2,3,1,2,3]

print find(L1)

print find(L2)

结果：



第七题

代码

def findcentre(A):

max=0

B=[]

for i in range(len(A)):

if A[i]>max:

max=A[i]

if A[i+1]<max:

i+=1

continue

else:

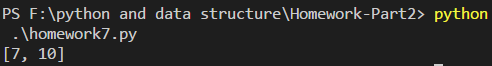
B.append(A[i])

return B

A=[3,1,6,5,4,7,9,8,10,14,12]

print findcentre(A)

结果：



第八题

代码：

def dis(x,y,z):

return abs(x-y)+abs(y-z)+abs(x-z)

def mindis(A,B,C):

i,j,k=0,0,0

min=dis(A[0],B[0],C[0])

L=[A[0],B[0],C[0]]

while i<len(A) and j<len(B) and k<len(C):

dist=dis(A[i],B[j],C[k])

if dist<min:

min=dist

L=[A[i],B[j],C[k]]

if A[i]<B[j]:

if A[i]<C[k]:

i+=1

else:

k+=1

else:

if B[j]<C[k]:

j+=1

else:

k+=1

print "The minimal distance is",min

return L

A=[0,1,2]

B=[1,2,3]

C=[2,3,4]

print mindis(A,B,C)

结果：

