**Answer1**

class node:

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

self.data=data

self.next=None

def displayList():

global head

if(head.next==None):

print("List is Empty")

else:

output=""

temp=head

while(temp!=None):

output=output+str(temp.data)+"\t"

temp=temp.next

print("Values= "+output)

def addAtBegining(val):

global head

temp=node(val)

temp.next=head

head=temp

print("Value "+str(val)+" added at begining")

def delFromEnd():

if(head.next==None):

print("List is Empty")

else:

temp=head

while((temp.next).next!=None):

temp=temp.next

print("Removed Value: "+str((temp.next).data))

temp.next=None

head=node(1)

addAtBegining(2)

addAtBegining(3)

addAtBegining(4)

displayList()

delFromEnd()

delFromEnd()

**Answer2**

arr=[5,10,15,20,25,30,35,40,45,50,55,60,65,70,75]

def find(x):

global arr

beg=0

end=len(arr)-1

while(arr[beg]<arr[end]):

mid=(beg+end)/2

if(x==arr[mid]):

return (mid+1)

else:

if(x>arr[mid]):

beg=mid+1

else:

end=mid

return -1

num=25

print(str(num)+" is present at pos: "+str(find(num)))

num=50

print(str(num)+" is present at pos: "+str(find(num)))

num=33

print(str(num)+" is present at pos: "+str(find(num)))

**Answer3**

count=0

class node:

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

self.data=data

self.next=None

def displayList():

global head

if(head.next==None):

print("List is Empty")

else:

output=""

temp=head

while(temp!=None):

output=output+str(temp.data)+"\t"

temp=temp.next

print("Values= "+output)

def addAtBegining(val):

global head

global count

temp=node(val)

temp.next=head

head=temp

print("Value "+str(val)+" added at begining")

count=count+1

def findMiddle():

print("count= "+str(count))

if(count==0):

print("List Empty")

elif(count==1 or count==2 or count==3):

print("Middle Value: "+str(head.data))

else:

mid=count/2

temp=head

i=0

while(i<mid):

print("i= "+str(i))

temp=temp.next

print("temp.val= "+str(temp.data))

i=i+1

print("Middle Value: "+str(temp.data))

head=node(1)

count=count+1

addAtBegining(2)

addAtBegining(3)

addAtBegining(4)

addAtBegining(5)

displayList()

findMiddle()