Anusha Jhampri

class node:

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

self.data=data

self.next=None

class LinkedList:

    def \_\_init\_\_(self):

        self.start = None

def insert(self,n\_node,n\_data):

# if no node is present

if self.start is None:

            n\_node.next = self.start

n\_node = node(n\_data)

            self.start = n\_node

# if only one node is present

elif self.start.data >= n\_node.data:

            n\_node.next = self.start

            self.start = n\_node

n\_node = node(n\_data)

# for more than one node present

# s is the current node

else:

s = self.head

            while(s.next is not None and s.next.data < n\_node.data):

                s = s.next

            n\_node.next = s.next

            s.next = n\_node

n\_node = node(n\_data)

def print(self):

        p = self.start

        while(p!=None):

            print p.data,

            p= p.next

l\_list = LinkedList()

n\_node = node(1)

l\_list.insert(n\_node)

n\_node = node(4)

l\_list.insert(n\_node)

n\_node = node(3)

l\_list.insert(n\_node)

n\_node = node(2)

l\_list.insert(n\_node)

n\_node = node(6)

l\_list.insert(n\_node)

n\_node = node(5)

l\_list.insert(n\_node)

print "Sorted Linked List:"

l\_list.print()