class ListNode:  
def \_\_init\_\_(self, val=0, next=None):  
self.val = val  
[self.next](http://self.next/) = next

def create\_new\_linked\_list(list1, list2):  
dummy = ListNode(0)  
current = dummy  
  
while list1 is not None and list2 is not None:  
if list1.val > list2.val:  
[current.next](http://current.next/) = ListNode(list1.val)  
list1 = [list1.next](http://list1.next/)  
else:  
[current.next](http://current.next/) = ListNode(list2.val)  
list2 = [list2.next](http://list2.next/)  
current = [current.next](http://current.next/)  
  
return [dummy.next](http://dummy.next/)

# Create the first linked list (list1)  
list1 = ListNode(5)  
[list1.next](http://list1.next/) = ListNode(2)  
[list1.next.next](http://list1.next.next/) = ListNode(3)  
[list1.next.next.next](http://list1.next.next.next/) = ListNode(8)  
  
# Create the second linked list (list2)  
list2 = ListNode(1)  
[list2.next](http://list2.next/) = ListNode(7)  
[list2.next.next](http://list2.next.next/) = ListNode(4)  
[list2.next.next.next](http://list2.next.next.next/) = ListNode(5)

new\_list = create\_new\_linked\_list(list1, list2)  
  
# Print the new linked list  
current = new\_list  
while current is not None:  
print(current.val, "->", end=" ")  
current = [current.next](http://current.next/)