**Given two linked lists of size N and M, which are sorted**

**in non-decreasing order. The task is to merge them in such a way that the resulting list is in non-increasing**

**order.**

**Input:**

**N = 2, M = 2**

**list1 = 1->3**

**list2 = 2->4**

**Output:**

**4->3->2->1**

**Explanation:**

**After merging the two lists in non-increasing**

**order, we have new lists as 4->3->2->1.**

**Approach:** convert the two lists into a single list and reverse it using a function called **Collections.reverseOrder()**

**Code:**

class GfG

{

Node mergeResult(Node node1, Node node2)

{

ArrayList<Integer> list = new ArrayList<Integer>();

Node current = node1;

while(current != null)

{

list.add(current.data);

current = current.next;

}

current = node2;

while(current!=null)

{

list.add(current.data);

current = current.next;

}

Collections.sort(list, Collections.reverseOrder());

Node head = null;

Node tail = null;

for(int i=0;i<list.size();i++)

{

Node newnode = new Node(list.get(i));

if(head == null)

{

head = newnode;

tail = newnode;

}

else{

tail.next = newnode;

tail = newnode;

}

}

return head;

}

}