# CS 400 HW 3: Linked Lists

**Note**: for CS 400’s homeworks, you will need to

Submit your solutions of BOTH questions to blackboard

## Question 1: merge two sorted linked lists

Develop a function takes two NodePtrs and returns another one. The two given NodePtrs are pointing at two linked list heads, where both linked lists are sorted in ascending order. The return NodePtr should points at a linked list formed by merging two given lists.

For example:

If list1 is 1->3->5->7 and list2 is 2->4->6->8, then the merge one should be 1->2->3->4->5->6->7->8.

Note that you do not need to use LinkedList class and its member functions for this lab. All operations should be done at Node level (such as creating a node and linking a node.)

The function should use the property that both given lists are sorted and have O(m+n) time complexity (m is the length of the first given list and n is the length of the second.)

You may use the definition of Node from lecture.

Note: use the .cpp files I gave to you to finish the work, and submit it through the blackboard.

## Question 2: re-arrange a linked list

Given a linked list with numbers, re-arrange the list so all even numbers will be after odd numbers. Note that relative order among odd numbers should be preserved, so should the relative order among even numbers.

For example, given a list 1 -> 2 -> 5 -> 8 -> 3 -> 6 -> 7 -> 4, the list should be re-organized to:

1 -> 5 -> 3 -> 7 -> 2 -> 8 ->6 -> 4. Note that the relative orders (1 is before 5 is before 3 is before 7; and 2 is before 8 is before 6 is before 4) are preserved.

Note: use the .cpp files I gave to you to finish the work, and submit it through the blackboard.