**[Flattening a Linked List](https://practice.geeksforgeeks.org/problems/da62a798bca208c7a678c133569c3dc7f5b73500/1)**

Given a Linked List of size N, where every node represents a sub-linked-list and contains two pointers:  
(i) a**next**pointer to the next node,  
(ii) a**bottom** pointer to a linked list where this node is head.  
Each of the sub-linked-list is in sorted order.  
Flatten the Link List such that all the nodes appear in a single level while maintaining the sorted order.   
**Note:** The flattened list will be printed using the bottom pointer instead of next pointer.

**Example 1:**

**Input:**

5 -> 10 -> 19 -> 28

| | | |

7 20 22 35

| | |

8 50 40

| |

30 45

**Output:**  5-> 7-> 8- > 10 -> 19-> 20->

22-> 28-> 30-> 35-> 40-> 45-> 50.

**Explanation**:

The resultant linked lists has every

node in a single level.

(**Note:** | represents the bottom pointer.)

**Example 2:**

**Input:**

5 -> 10 -> 19 -> 28

| |

7 22

| |

8 50

|

30

**Output:** 5->7->8->10->19->22->28->30->50

**Explanation:**

The resultant linked lists has every

node in a single level.

(**Note:** | represents the bottom pointer.)

**Your Task:**  
You do not need to read input or print anything. Complete the function **flatten()** that takes the **head**of the linked list as input parameterand returns the head of flattened link list.

**Expected Time Complexity:** O(N\*M)  
**Expected Auxiliary Space:** O(1)

**Constraints:**  
0 <= N <= 50  
1 <=**Mi**<= 20  
1 <= Element of linked list <= 103