**Binary Tree to CDLL:-**

Given a Binary Tree of**N** edges. The task is to convert this to a Circular Doubly Linked List(**CDLL**) in-place. The left and right pointers in nodes are to be used as previous and next pointers respectively in converted CDLL. The order of nodes in CDLL must be same as Inorder of the given Binary Tree. The first node of Inorder traversal (left most node in BT) must be head node of the CDLL.

**Example 1:**

**Input:**

      1

   /   \

  3     2

**Output:**

3 1 2

2 1 3

**Explanation:** After converting tree to CDLL

when CDLL is is traversed from head to

tail and then tail to head, elements

are displayed as in the output.

**Example 2:**

**Input:**

     10

  /    \

 20    30

/  \

40  60

**Output:**

40 20 60 10 30

30 10 60 20 40

**Explanation:**After converting tree to CDLL,

when CDLL is is traversed from head to

tail and then tail to head, elements

are displayed as in the output.

**Your Task:**  
You don't have to take input. Complete the function **bTreeToCList()**that takes root as a parameter and **returns**the **head of the list**. The printing is done by the driver code.

**Constraints:**  
1 <= N <= 103  
1 <= Data of a node <= 104

**Expected time complexity:**O(N)

**Expected auxiliary space:**O(h) , where h = height of tree