# Inserting a Node Into a Sorted Doubly Linked List



Given a reference to the head of a doubly-linked list and an integer, data, create a new DoublyLinkedListNode object having data value data and insert it into a sorted linked list.

Complete the DoublyLinkedListNode SortedInsert(DoublyLinkedListNode head, int data) method in the editor below. It has two parameters:

- 1. *head*: A reference to the head of a doubly-linked list of *Node* objects.
- 2. *data*: An integer denoting the value of the *data* field for the *Node* you must insert into the list.

The method must insert a new *Node* into the sorted (in ascending order) doubly-linked list whose data value is *data* without breaking any of the list's double links or causing it to become unsorted.

**Note:** Recall that an empty list (i.e., where head = null) and a list with one element *are* sorted lists.

### **Input Format**

The first line contains an integer t, the number of test cases.

Each of the test case is in the following format:

- The first line contains an integer n, the number of elements in the linked list.
- Each of the next *n* lines contains an integer, the *data* for each node of the linked list.
- The last line contains an integer data which needs to be inserted into the sorted doubly-linked list.

# **Constraints**

- $1 \le t \le 10$
- $1 \le n \le 1000$
- $1 \le list[i] \le 1000$

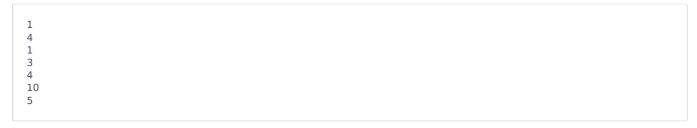
#### **Output Format**

**Do not print anything to stdout.** Your method must return a reference to the head of the same list that was passed to it as a parameter.

The ouput is handled by the code in the editor and is as follows:

For each test case, print the elements of the sorted doubly-linked list separated by spaces on a new line.

#### Sample Input



## **Sample Output**

# **Explanation**

The initial doubly linked list is:  $1\leftrightarrow 3\leftrightarrow 4\leftrightarrow 10 \to NULL$  .

The doubly linked list after insertion is:  $1\leftrightarrow 3\leftrightarrow 4\leftrightarrow 5\leftrightarrow 10\to NULL$