**Insert in a Sorted List: -**

**Easy** Accuracy: **31.37%** Submissions: **103K+** Points: **2**

Given a linked list sorted in ascending order and an integer called **data**, insert data in the linked list such that the list remains sorted.

**Example 1:**

**Input:**

LinkedList: 25->36->47->58->69->80

data: 19

**Output:**19 25 36 47 58 69 80  
**Explanation:**  
After inserting 19 the sorted linked list will look like the one in the output.

**Example 2:**

**Input:**

LinkedList: 50->100

data: 75

**Output:**50 75 100  
**Explanation:**  
After inserting 75 the sorted linked list will look like the one in the output.

**Your Task:**  
The task is to complete the function **sortedInsert**() which should insert the element in sorted Linked List and return the head of the linked list

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

**Constraints:**  
1 <= N <= 104  
-99999 <= A[i] <= 99999, for each valid i

**Code: -**

//{ Driver Code Starts

//

#include <bits/stdc++.h>

using namespace std;

struct Node

{

int data;

struct Node\* next;

Node(int x){

data = x;

next = NULL;

}

};

void printList(Node\* node)

{

while (node != NULL) {

cout << node->data <<" ";

node = node->next;

}

cout<<"\n";

}

// } Driver Code Ends

/\*

structure of the node of the list is as

struct Node

{

int data;

struct Node\* next;

Node(int x){

data = x;

next = NULL;

}

};

\*/

class Solution{

public:

// Should return head of the modified linked list

Node \*sortedInsert(struct Node\* head, int data) {

// Code here

Node \*node = new Node(data);

if(data <= head->data){

node->next = head;

return node;

}

Node \*ptr = head, \*prev = NULL;

while(ptr){

if(prev and prev->data <= data and data <= ptr->data){

prev->next = node;

node->next = ptr;

break;

}

else if(!ptr->next and ptr->data < data){

ptr->next = node;

break;

}

prev = ptr;

ptr = ptr->next;

}

return head;

}

};

//{ Driver Code Starts.

int main()

{

int t;

cin>>t;

while(t--)

{

int n;

cin>>n;

int data;

cin>>data;

struct Node \*head = new Node(data);

struct Node \*tail = head;

for (int i = 0; i < n-1; ++i)

{

cin>>data;

tail->next = new Node(data);

tail = tail->next;

}

int k;

cin>>k;

Solution obj;

head = obj.sortedInsert(head,k);

printList(head);

}

return 0;

}

// } Driver Code Ends

**T.C: - O(N)**

**S.C: - O(1)**