**Flatten A Linked List**

/\*

\* Definition for linked list.

\* class Node {

\* public:

\* int data;

\* Node \*next;

\* Node \*child;

\* Node() : data(0), next(nullptr), child(nullptr){};

\* Node(int x) : data(x), next(nullptr), child(nullptr) {}

\* Node(int x, Node \*next, Node \*child) : data(x), next(next), child(child) {}

\* };

\*/

Node\* merge(Node\* left, Node\* right)

{

if(left == NULL)

return right;

if(right == NULL)

return left;

Node\* ans = new Node(-1);

Node\* temp = ans;

while(left != NULL && right != NULL)

{

if(left->data < right->data)

{

temp->child = left;

temp = left;

left = left->child;

}

else

{

temp->child = right;

temp = right;

right = right->child;

}

}

if(left != NULL)

{

temp->child = left;

temp = left;

left = left->child;

}

if(right != NULL)

{

temp->child = right;

temp = right;

right = right->child;

}

return ans->child;

}

Node\* flattenLinkedList(Node\* head)

{

if(head == NULL)

return head;

Node\* down = head;

Node\* right = head->next;

down->next = NULL;

right = flattenLinkedList(right);

Node\* ans = merge(down, right);

return ans;

}