Merge Two Sorted Lists

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
 *     ListNode(int x) : val(x), next(nullptr) {}
 *     ListNode(int x, ListNode *next) : val(x), next(next) {}
 * };
 */
class Solution {
public:
    ListNode* mergeTwoLists(ListNode* list1, ListNode* list2,ListNode*
head = nullptr, int calls = 0, ListNode* cur = nullptr) {

        if (list1 == nullptr && list2 == nullptr){
          return head;
        }

        int check;

        if (list1 == nullptr) check = 2;
        else if (list2 == nullptr) check = 1;
        else if (list1->val <= list2->val) check = 1;
        else if (list1->val > list2->val) check = 2;

        if (check == 1){
          if (calls == 0){
            head = list1;
            list1 = list1->next;
            head->next = nullptr;
            cur = head;
            calls +=1;
          }else{
```

```
            cur->next = list1;
            list1 = list1->next;
            cur = cur->next;
            cur->next = nullptr;
          }
          return mergeTwoLists(list1,list2,head,calls,cur);
        }else if (check == 2){
          if (calls == 0){
            head = list2;
            list2 = list2->next;
            head->next = nullptr;
            cur = head;
            calls +=1;
          }else{
            cur->next = list2;
            list2 = list2->next;
            cur = cur->next;
            cur->next = nullptr;
          }
          return mergeTwoLists(list1,list2,head,calls,cur);
        }

        return head;
    }
};
```