HACKER RANK: Merge-two-sorted-linked-list

```
// Complete the mergeLists function below.

/*
 * For your reference:
 *
 * SinglyLinkedListNode {
 *     int data;
 *     SinglyLinkedListNode* next;
 };
SinglyLinkedListNode* mergeLists(SinglyLinkedListNode* head1, SinglyLinkedListNode* head2) {
    SinglyLinkedListNode *head3 = NULL, *t1 = head1, *t2 = head2, *t3 = NULL;

    while (t1 != NULL && t2 != NULL) {
        SinglyLinkedListNode* newNode = (SinglyLinkedListNode*)malloc(sizeof(SinglyLinkedListNode));

        if (t1->data < t2->data) {
            newNode->data = t1->data;
            t1 = t1->next;
        } else {
            newNode->data = t2->data;
            t2 = t2->next;
        }

        newNode->next = NULL;

        if (head3 == NULL) {
            head3 = newNode;
            t3 = head3;
        } else {
```

```
            t3->next = newNode;
            t3 = newNode;
        }
    }

    // If one of the lists is not fully processed, append the remaining elements to the merged list.
    if (t1 != NULL) {
        if (head3 == NULL) {
            head3 = t1;
        } else {
            t3->next = t1;
        }
    }

    if (t2 != NULL) {
        if (head3 == NULL) {
            head3 = t2;
        } else {
            t3->next = t2;
        }
    }

    return head3;
}
```

OUTPUT:

Input (stdin)
Download

- 1
- 3
- 1
- 2
- 3
- 2
- 3
- 4

Expected Output: **1 2 3 3 4**