

```

SinglyLinkedListNode* mergeLists(SinglyLinkedListNode* head1, SinglyLinkedListNode* head2) {

    SinglyLinkedListNode *head3 = NULL, *t1 = head1, *t2 = head2, *t3 = NULL;

    while (t1 != NULL && t2 != NULL) {

        SinglyLinkedListNode* newNode = 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;

}
```

Line: 122 Col: 2

Upload Code as File




☐ Test against custom input

Run Code

Submit Code

## Congratulations

You solved this challenge. Would you like to challenge your friends?



Next Challenge

Test case 0

Test case 1

Test case 2

Test case 3

Test case 4

Test case 5

Test case 6

Input (stdin)

Download

1	1
2	3
3	1
4	2
5	3
6	2
7	3
8	4

Expected Output

Download

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