```c
#include <stdio.h>
#include <stdlib.h>
struct Node {
    int data;
    struct Node* next;
};
struct Node *createNode(int data) {
    struct Node *newNode = malloc(sizeof(struct Node));
    if (newNode == NULL) {
        printf("Memory allocation error.\n");
        exit(EXIT_FAILURE);
    }
    newNode->data = data;
    newNode->next = NULL;
    return newNode;
}
struct Node* merge_list(struct Node* head1, struct Node* head2) {
    struct Node *head3 = NULL;
    struct Node *current1 = head1;
    struct Node *current2 = head2;

    while (current1 != NULL && current2 != NULL) {
        struct Node* newNode = createNode(0);

        if (current1->data < current2->data) {
            newNode->data = current1->data;
            current1 = current1->next;
        } else if (current1->data > current2->data) {
            newNode->data = current2->data;
            current2 = current2->next;
        } else if (current1->data == current2->data) {
            newNode->data = current1->data;
            current1 = current1->next;
            current2 = current2->next;
        }

        if (head3 == NULL) {
            head3 = newNode;
        } else {
            struct Node* current3 = head3;
            while (current3->next != NULL) {
                current3 = current3->next;
            }
            current3->next = newNode;
        }
    }
    while (current1 != NULL) {
        struct Node* newNode = createNode(current1->data);
        current1 = current1->next;

        if (head3 == NULL) {
            head3 = newNode;
        } else {
            struct Node* current3 = head3;
            while (current3->next != NULL) {
                current3 = current3->next;
            }
            current3->next = newNode;
        }
    }
    while (current2 != NULL) {
        struct Node* newNode = createNode(current2->data);
        current2 = current2->next;
        if (head3 == NULL) {
            head3 = newNode;
        } else {
```

```c
            struct Node *current3 = head3;
            while (current3->next != NULL) {
                current3 = current3->next;
            }
            current3->next = newNode;
        }
    }

    return head3;
}
void printList(struct Node* head) {
    while (head != NULL) {
        printf("%d -> ", head->data);
        head = head->next;
    }
    printf("NULL\n");
}
void freeList(struct Node* head) {
    struct Node* temp;
    while (head != NULL) {
        temp = head;
        head = head->next;
        free(temp);
    }
}

int main() {
    struct Node* list1 = createNode(1);
    list1->next = createNode(3);
    list1->next->next = createNode(5);
    struct Node* list2 = createNode(2);
    list2->next = createNode(4);
    list2->next->next = createNode(6);
    printf("Original List 1: ");
    printList(list1);
    printf("Original List 2: ");
    printList(list2);
    struct Node* mergedList = merge_list(list1, list2);
    printf("Merged List: ");
    printList(mergedList);
    freeList(list1);
    freeList(list2);
    freeList(mergedList);

    return 0;

```