```
#include<stdio.h>
#include <stdlib.h>
struct Node {
 int data;
 struct Node* next;
};
struct Node* createNode(int data) {
 struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
 newNode->data = data;
 newNode->next = NULL;
 return newNode;
}
void insertEnd(struct Node** head, int data) {
 struct Node* newNode = createNode(data);
 if (*head == NULL) {
 *head = newNode;
 return;
 }
 struct Node* temp = *head;
 while (temp->next != NULL) {
 temp = temp->next;
 }
 temp->next = newNode;
}
void display(struct Node* head) {
 while (head) {
 printf("%d -> ", head->data);
 head = head->next;
 }
 printf("NULL\n");
}
```

```
void sortList(struct Node* head) {
 struct Node *i, *j;
 int temp;
 for (i = head; i != NULL; i = i->next) {
 for (j = i->next; j != NULL; j = j->next) {
 if (i->data > j->data) {
 temp = i->data;
 i->data = j->data;
 j->data = temp;
 }
 }
 }
}
void reverseList(struct Node** head) {
 struct Node *prev = NULL, *current = *head, *next = NULL;
 while (current != NULL) {
 next = current->next;
 current->next = prev;
 prev = current;
 current = next;
 }
 *head = prev;
}
void concatenate(struct Node** head1, struct Node* head2) {
 if (*head1 == NULL) {
 *head1 = head2;
 return;
 }
 struct Node* temp = *head1;
 while (temp->next != NULL)
 temp = temp->next;
```

```
 temp->next = head2;
}
int main() {
 struct Node* list1 = NULL;
 struct Node* list2 = NULL;
 insertEnd(&list1, 40);
 insertEnd(&list1, 10);
 insertEnd(&list1, 50);
 insertEnd(&list1, 70);
 printf("List 1: ");
 display(list1);
 insertEnd(&list2, 7);
 insertEnd(&list2, 3);
 insertEnd(&list2, 9);
 printf("List 2: ");
 display(list2);
 sortList(list1);
 printf("\nList 1 after sorting: ");
 display(list1);
 reverseList(&list1);
 printf("\nList 1 after reversal: ");
 display(list1);
 concatenate(&list1, list2);
 printf("\nAfter concatenation (List1 + List2): ");
display(list1);
 return 0;
}
```

```
List 1: 40 -> 10 -> 50 -> 70 -> NULL
List 2: 7 -> 3 -> 9 -> NULL

List 1 after sorting: 10 -> 40 -> 50 -> 70 -> NULL

List 1 after reversal: 70 -> 50 -> 40 -> 10 -> NULL

After concatenation (List1 + List2): 70 -> 50 -> 40 -> 10 -> 7 -> 3 -> 9 -> NULL

Process returned 0 (0x0)   execution time : 0.010 s
Press any key to continue.
```