```
#include <stdio.h>
#include <stdlib.h>

struct Node {
   int data;
   struct Node* next;
};

void insertNode(struct Node** head, int data) {
   struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
   newNode->data = data;
   newNode->next = NULL;

   if (*head == NULL) {
      *head = newNode;
      return;
   }

   struct Node* temp = *head;
   while (temp->next != NULL) {
      temp = temp->next;
   }

   temp->next = newNode;
}

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

struct Node* copyList(struct Node* head) {
   struct Node* newHead = NULL;
   struct Node* tail = NULL;

   while (head != NULL) {
      insertNode(&newHead, head->data);
      head = head->next;
   }

   return newHead;
}

void sortList(struct Node** head) {
   int swapped, temp;
   struct Node* ptr1;
```

```
  struct Node* lptr = NULL;

  if (*head == NULL)
    return;

  do {
    swapped = 0;
    ptr1 = *head;

    while (ptr1->next != lptr) {
      if (ptr1->data > ptr1->next->data) {
        temp = ptr1->data;
        ptr1->data = ptr1->next->data;
        ptr1->next->data = temp;
        swapped = 1;
      }
      ptr1 = ptr1->next;
    }
    lptr = ptr1;
  } while (swapped);
}

void reverseList(struct Node** head) {
  struct Node *prev, *current, *next;
  prev = NULL;
  current = *head;

  while (current != NULL) {
    next = current->next;
    current->next = prev;
    prev = current;
    current = next;
  }

  *head = prev;
}

struct Node* concatenateLists(struct Node* first, struct Node* second) {
  struct Node* result = copyList(first);

  struct Node* temp = result;
  while (temp->next != NULL) {
    temp = temp->next;
  }

  temp->next = copyList(second);

  return result;
}
```

```
int main() {
    struct Node* list1 = NULL;
    struct Node* list2 = NULL;
    int n, data;

    printf("Enter the number of elements for List 1: ");
    scanf("%d", &n);

    printf("Enter elements for List 1:\n");
    for (int i = 0; i < n; ++i) {
        scanf("%d", &data);
        insertNode(&list1, data);
    }

    printf("Enter the number of elements for List 2: ");
    scanf("%d", &n);

    printf("Enter elements for List 2:\n");
    for (int i = 0; i < n; ++i) {
        scanf("%d", &data);
        insertNode(&list2, data);
    }

    struct Node* sortedList = copyList(list1);
    sortList(&sortedList);
    printf("Sorted List 1: ");
    printList(sortedList);

    struct Node* reversedList = copyList(list1);
    reverseList(&reversedList);
    printf("Reversed List 1: ");
    printList(reversedList);

    struct Node* concatenatedList = concatenateLists(list1, list2);
    printf("Concatenated List: ");
    printList(concatenatedList);

    return 0;
}
```

```
Enter the number of elements for List 1: 4
Enter elements for List 1:
6
3
6
3
Enter the number of elements for List 2: 6
Enter elements for List 2:
2
783
6
2
7
9
Sorted List 1: 3 -> 3 -> 6 -> 6 -> NULL
Reversed List 1: 3 -> 6 -> 3 -> 6 -> NULL
Concatenated List: 6 -> 3 -> 6 -> 3 -> 2 -> 783 -> 6 -> 2 -> 7 -> 9 -> NULL
```