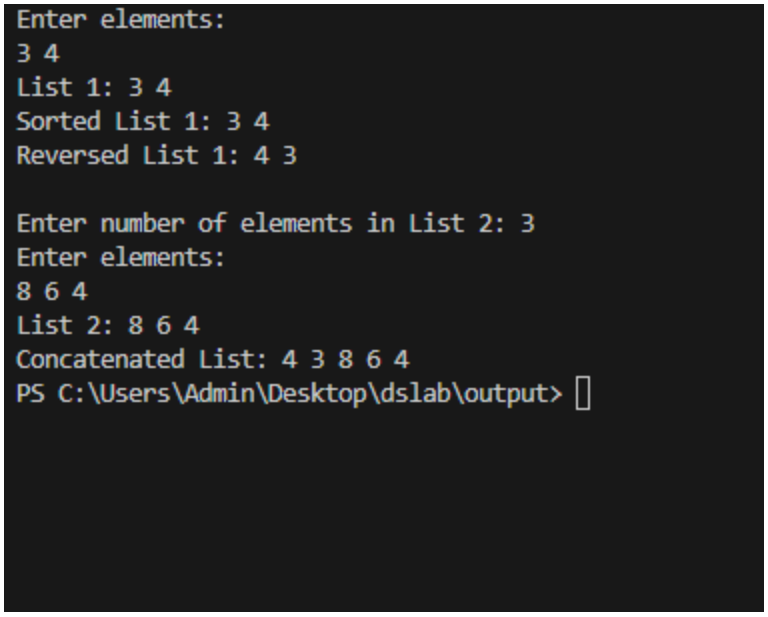
```c
#include <stdio.h>
#include <stdlib.h>

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

struct node* createNode(int x) {
    struct node *newnode = (struct node*)malloc(sizeof(struct node));
    newnode->data = x;
    newnode->next = NULL;
    return newnode;
}

void insertEnd(struct node **head, int x) {
    struct node *newnode = createNode(x);

    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) {
    if (head == NULL) {
        printf("List is empty\n");
        return;
    }
    while (head != NULL) {
        printf("%d ", head->data);
        head = head->next;
    }
    printf("\n");
```

```c
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, *curr = *head, *next = NULL;

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

    *head = prev;
}
struct node* concatenate(struct node *head1, struct node *head2) {
    if (head1 == NULL) return head2;
    if (head2 == NULL) return head1;

    struct node *temp = head1;
    while (temp->next != NULL)
        temp = temp->next;

    temp->next = head2;

    return head1;
}
```

```c
int main() {
    struct node *list1 = NULL, *list2 = NULL, *finalList;
    int n, x;
    printf("Enter number of elements in List 1: ");
    scanf("%d", &n);

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

    printf("List 1: ");
    display(list1);


    sortList(&list1);
    printf("Sorted List 1: ");
    display(list1);

    reverseList(&list1);
    printf("Reversed List 1: ");
    display(list1);
    printf("\nEnter number of elements in List 2: ");
    scanf("%d", &n);

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

    printf("List 2: ");
    display(list2);
    finalList = concatenate(list1, list2);
    printf("Concatenated List: ");
    display(finalList);

    return 0;
}
```


Enter elements:
3 4
List 1: 3 4
Sorted List 1: 3 4
Reversed List 1: 4 3
Enter number of elements in List 2: 3
Enter elements:
8 6 4
List 2: 8 6 4
Concatenated List: 4 3 8 6 4
PS C:\Users\Admin\Desktop\dslab\output>