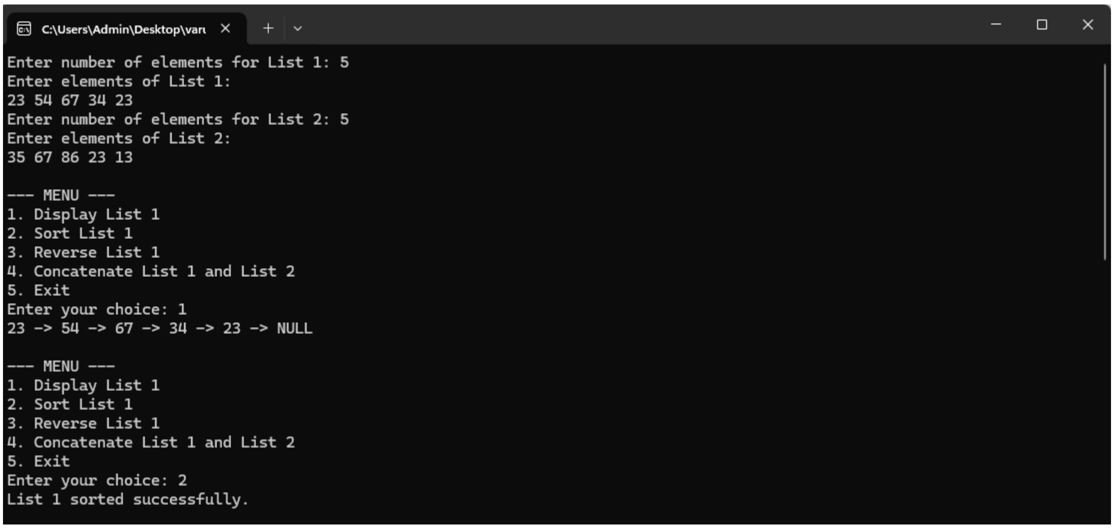
6. a) WAP to Implement Single Link List with following operations: Sort the linked list, Reverse the linked list, Concatenation of two linked lists.

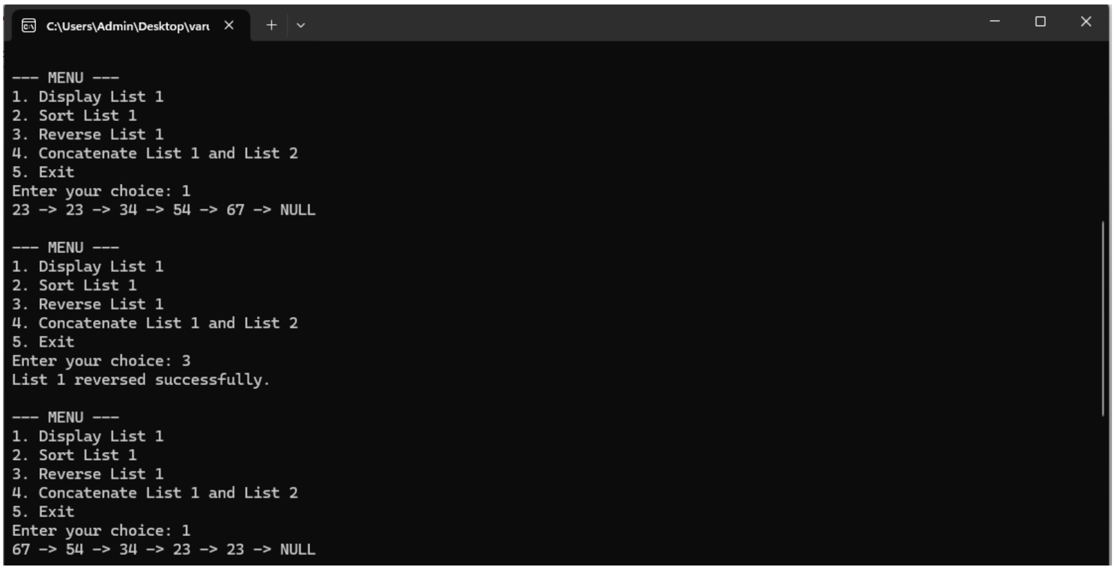
```
#include <stdio.h>
#include <stdlib.h>

/* Node structure */
struct node {
   int data;
   struct node *next;
};

/* Create new node */
struct node* createNode(int data) {
   struct node *newNode = (struct node*)malloc(sizeof(struct node));
   newNode->data = data;
   newNode->next = NULL;
   return newNode;
}

/* Insert node at end */
struct node* insertEnd(struct node *head, int data) {
   struct node *newNode = createNode(data);
   struct node *temp;

   if (head == NULL)
      return newNode;

   temp = head;
   while (temp->next != NULL)
      temp = temp->next;

   temp->next = newNode;
   return head;
}

/* Display linked list */
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("NULL\n");
}

/* Sort linked list (Bubble Sort) */
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;
            }
        }
    }
}

/* Reverse linked list */
struct node* 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;
    }
    return prev;
}

/* Concatenate two lists */
struct node* concatenate(struct node *head1, struct node *head2) {
    struct node *temp;

    if (head1 == NULL)
        return head2;

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

    temp->next = head2;
    return head1;
}

/* Main Function */
```

```
int main() {
    struct node *list1 = NULL, *list2 = NULL;
    int n, val, choice;

    /* Input List 1 */
    printf("Enter number of elements for List 1: ");
    scanf("%d", &n);
    printf("Enter elements of List 1:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &val);
        list1 = insertEnd(list1, val);
    }

    /* Input List 2 */
    printf("Enter number of elements for List 2: ");
    scanf("%d", &n);
    printf("Enter elements of List 2:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &val);
        list2 = insertEnd(list2, val);
    }

    /* Menu */
    do {
        printf("\n--- MENU ---");
        printf("\n1. Display List 1");
        printf("\n2. Sort List 1");
        printf("\n3. Reverse List 1");
        printf("\n4. Concatenate List 1 and List 2");
        printf("\n5. Exit");
        printf("\nEnter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                display(list1);
                break;
            case 2:
                sortList(list1);
                printf("List 1 sorted successfully.\n");
                break;
            case 3:
                list1 = reverseList(list1);
                printf("List 1 reversed successfully.\n");
                break;
            case 4:
                list1 = concatenate(list1, list2);
                printf("Lists concatenated successfully.\n");
                break;
            case 5:
                printf("Exiting program...\n");
                exit(0);
            default:
                printf("Invalid choice!\n");
        }
    } while (1);

    return 0;
}
```

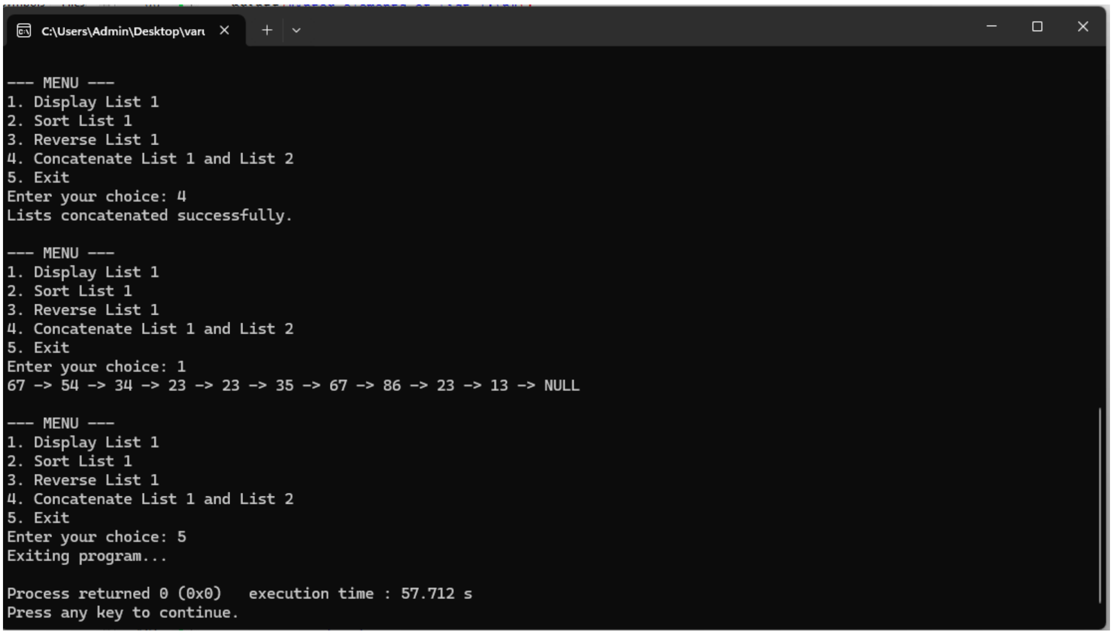
Output:



```
Enter number of elements for List 1: 5
Enter elements of List 1:
23 54 67 34 23
Enter number of elements for List 2: 5
Enter elements of List 2:
35 67 86 23 13

--- MENU ---
1. Display List 1
2. Sort List 1
3. Reverse List 1
4. Concatenate List 1 and List 2
5. Exit
Enter your choice: 1
23 -> 54 -> 67 -> 34 -> 23 -> NULL

--- MENU ---
1. Display List 1
2. Sort List 1
3. Reverse List 1
4. Concatenate List 1 and List 2
5. Exit
Enter your choice: 2
List 1 sorted successfully.
```



```
--- MENU ---
1. Display List 1
2. Sort List 1
3. Reverse List 1
4. Concatenate List 1 and List 2
5. Exit
Enter your choice: 1
23 -> 23 -> 34 -> 54 -> 67 -> NULL

--- MENU ---
1. Display List 1
2. Sort List 1
3. Reverse List 1
4. Concatenate List 1 and List 2
5. Exit
Enter your choice: 3
List 1 reversed successfully.

--- MENU ---
1. Display List 1
2. Sort List 1
3. Reverse List 1
4. Concatenate List 1 and List 2
5. Exit
Enter your choice: 1
67 -> 54 -> 34 -> 23 -> 23 -> NULL
```



```
--- MENU ---
1. Display List 1
2. Sort List 1
3. Reverse List 1
4. Concatenate List 1 and List 2
5. Exit
Enter your choice: 4
Lists concatenated successfully.

--- MENU ---
1. Display List 1
2. Sort List 1
3. Reverse List 1
4. Concatenate List 1 and List 2
5. Exit
Enter your choice: 1
67 -> 54 -> 34 -> 23 -> 23 -> 35 -> 67 -> 86 -> 23 -> 13 -> NULL

--- MENU ---
1. Display List 1
2. Sort List 1
3. Reverse List 1
4. Concatenate List 1 and List 2
5. Exit
Enter your choice: 5
Exiting program...

Process returned 0 (0x0)   execution time : 57.712 s
Press any key to continue.
```