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

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

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

// Function to create a 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 at end
struct Node* insertEnd(struct Node* start, int item) {
    struct Node* p = createNode(item);
    if (start == NULL) {
        start = p;
    } else {
        struct Node* temp = start;
        while (temp->next != NULL)
            temp = temp->next;
        temp->next = p;
    }
    return start;
}

// Display list
void display(struct Node* head) {
    if (head == NULL) {
        printf("List is empty!\n");
        return;
    }
    struct Node* temp = head;
    while (temp != NULL) {
        printf("%d -> ", temp->data);
        temp = temp->next;
    }
    printf("NULL\n");
}
```

```c
// Sort list (Bubble Sort)
void sortList(struct Node* head) {
    if (head == NULL) return;
    struct Node* i, *j;
    int temp;
    for (i = head; i->next != 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;
            }
        }
    }
    printf("List sorted successfully!\n");
}

// Reverse list
struct Node* reverseList(struct Node* head) {
    struct Node* prev = NULL;
    struct Node* curr = head;
    struct Node* next = NULL;
    while (curr != NULL) {
        next = curr->next;
        curr->next = prev;
        prev = curr;
        curr = next;
    }
    printf("List reversed successfully!\n");
    return prev;
}

// Concatenate two lists
struct Node* concatenate(struct Node* list1, struct Node* list2) {
    if (list1 == NULL) return list2;
    struct Node* temp = list1;
    while (temp->next != NULL)
        temp = temp->next;
    temp->next = list2;
    printf("Lists concatenated successfully!\n");
    return list1;
}

// Main function
int main() {
    struct Node* list1 = NULL;
    struct Node* list2 = NULL;
```

```c
    int choice, item, n, n1, n2;

    while (1) {
        printf("\n--- MENU ---\n");
        printf("1. Insert into List1\n");
        printf("2. Insert into List2\n");
        printf("3. Display List1\n");
        printf("4. Display List2\n");
        printf("5. Sort List1\n");
        printf("6. Reverse List1\n");
        printf("7. Concatenate List1 and List2\n");
        printf("8. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter number of nodes for List1: ");
                scanf("%d", &n1);
                printf("Enter values: ");
                for (int i = 0; i < n1; i++) {
                    scanf("%d", &item);
                    list1 = insertEnd(list1, item);
                }
                break;
            case 2:
                printf("Enter number of nodes for List2: ");
                scanf("%d", &n2);
                printf("Enter values: ");
                for (int i = 0; i < n2; i++) {
                    scanf("%d", &item);
                    list2 = insertEnd(list2, item);
                }
                break;
            case 3:
                printf("List1: ");
                display(list1);
                break;
            case 4:
                printf("List2: ");
                display(list2);
                break;
            case 5:
                sortList(list1);
                break;
            case 6:
                list1 = reverseList(list1);
                break;
```
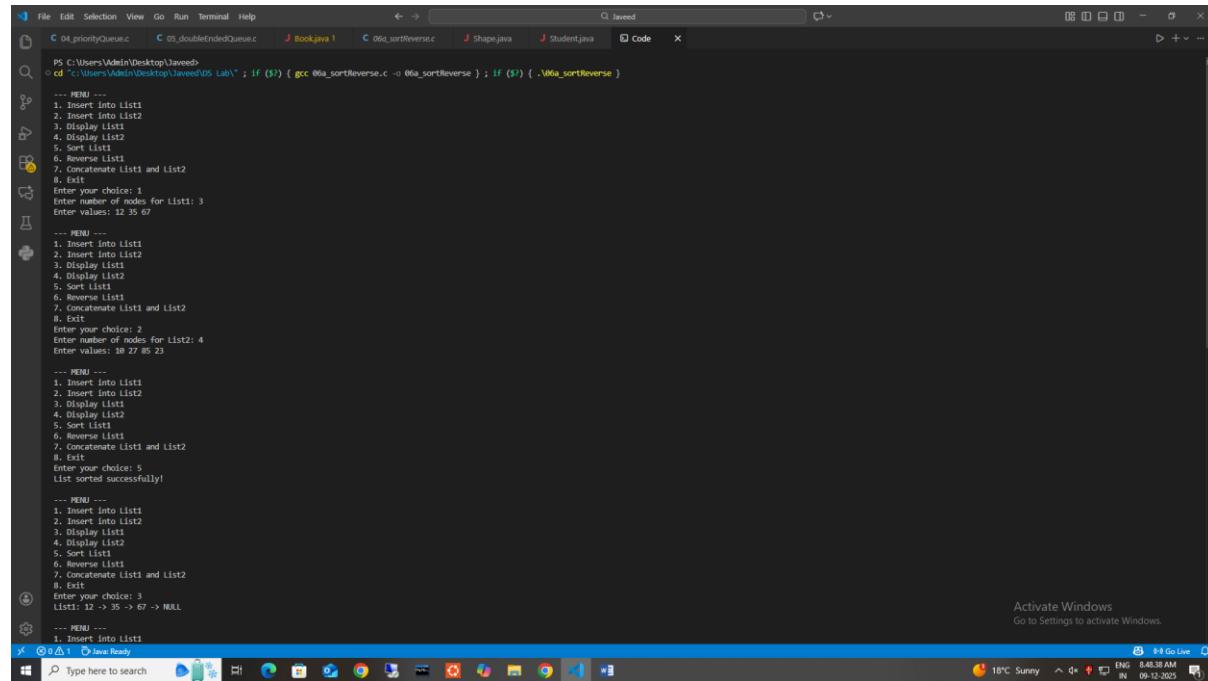
```c
        case 7:
            list1 = concatenate(list1, list2);
            break;
        case 8:
            printf("Exiting program...\n");
            exit(0);
        default:
            printf("Invalid choice! Try again.\n");
        }
    }
    return 0;
}
```

OUTPUT:

```
cd "c:\Users\Admin\Desktop\Javeed\DS Lab\" ; if ($?) { gcc 06a_sortReverse.c -o 06a_sortReverse } ; if ($?) { .\06a_sortReverse }
1. Insert into List1
2. Insert into List2
3. Display List1
4. Display List2
5. Sort List1
6. Reverse List1
7. Concatenate List1 and List2
8. Exit
Enter your choice: 6
List reversed successfully!

--- MENU ---
1. Insert into List1
2. Insert into List2
3. Display List1
4. Display List2
5. Sort List1
6. Reverse List1
7. Concatenate List1 and List2
8. Exit
Enter your choice: 3
List1: 67 -> 35 -> 12 -> NULL

--- MENU ---
1. Insert into List1
2. Insert into List2
3. Display List1
4. Display List2
5. Sort List1
6. Reverse List1
7. Concatenate List1 and List2
8. Exit
Enter your choice: 7
Lists concatenated successfully!

--- MENU ---
1. Insert into List1
2. Insert into List2
3. Display List1
4. Display List2
5. Sort List1
6. Reverse List1
7. Concatenate List1 and List2
8. Exit
Enter your choice: 3
List1: 67 -> 35 -> 12 -> 10 -> 27 -> 85 -> 23 -> NULL

--- MENU ---
1. Insert into List1
2. Insert into List2
3. Display List1
4. Display List2
5. Sort List1
6. Reverse List1
```