LAB PROGRAM 6A

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>
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 push(struct Node** top, int data) {
    struct Node* newNode = createNode(data);
    newNode->next = *top;
    *top = newNode;
}

int pop(struct Node** top) {
    if (*top == NULL) {
        printf("Stack Underflow\n");
        return -1;
    }
    int data = (*top)->data;
```

```
    struct Node* temp = *top;
    *top = (*top)->next;
    free(temp);
    return data;
}

void enqueue(struct Node** front, struct Node** rear, int data) {
    struct Node* newNode = createNode(data);
    if (*rear == NULL) {
        *front = *rear = newNode;
        return;
    }
    (*rear)->next = newNode;
    *rear = newNode;
}

int dequeue(struct Node** front) {
    if (*front == NULL) {
        printf("Queue Underflow\n");
        return -1;
    }
    int data = (*front)->data;
    struct Node* temp = *front;
    *front = (*front)->next;
    free(temp);
    return data;
}

void display(struct Node* head) {
    while (head != NULL) {
        printf("%d -> ", head->data);
```

```
        head = head->next;
    }
    printf("NULL\n");
}

int main() {
    struct Node* stack = NULL;
    struct Node *front = NULL, *rear = NULL;

    int choice, data;

    while (1) {
        printf("\nMenu:\n");
        printf("1. Push (Stack)\n");
        printf("2. Pop (Stack)\n");
        printf("3. Display Stack\n");
        printf("4. Enqueue (Queue)\n");
        printf("5. Dequeue (Queue)\n");
        printf("6. Display Queue\n");
        printf("7. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter value to push: ");
                scanf("%d", &data);
                push(&stack, data);
                break;

            case 2:
```

```
            printf("Popped: %d\n", pop(&stack));
            break;

        case 3:
            printf("Stack: ");
            display(stack);
            break;

        case 4:
            printf("Enter value to enqueue: ");
            scanf("%d", &data);
            enqueue(&front, &rear, data);
            break;

        case 5:
            printf("Dequeued: %d\n", dequeue(&front));
            break;

        case 6:
            printf("Queue: ");
            display(front);
            break;
        case 7:
            exit(0);

        default:
            printf("Invalid choice!\n");
      }
   }
   return 0;
}
```