Program 7(operations)

```
void reverse (struct node **head){
    struct node *q, *r, *s;
    q = *head;
    r = NULL;
    if (*head == NULL){
        printf("Empty LIST\n");
        return;
    }
    while (q != NULL){
        s = r;
        r = q;
        q = q -> next;
        r -> next = s;
    }
    *head = r;
}

struct node * merge(struct node *a, struct node *b)
    if (a == NULL){
        return b;
    }
    if (b == NULL){
        return a;
    }
```

```
struct node *c = NULL;
if (a->data < b->data) {
        c = a;
        c->next = merger(a->next, b);
}
else {
        c = b;                    b->next
        c->next = merger(a->next, c);
}
return c;
}
struct node* MidPoint(struct node *head) {
        if (head == NULL || head->next == NULL)
        {
            return head;
        }
        struct node *fast = head->next;
        struct node *slow = head;

        while(fast != NULL && fast->next != NULL)
        {
            fast = fast->next->next;
            slow = slow->next;
        }
        return slow;
}
```

```
struct node * MergeSort(struct node* head)
{
    if(head == NULL || head->next == NULL)
        return head;
}

struct node* mid = MidPoint(head);
struct node* a = head;
struct node* b = mid->next;

mid->next = NULL;

a = MergeSort(a);
b = MergeSort(b);
struct node* c = merge(a,b);
return c;
}
```