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

struct Node{
    int data;
    struct Node *prev, *next;
};
struct Node *createNode(int val){
        struct Node *nn;
        nn = (struct Node *) malloc(sizeof(struct Node));
        nn->data = val;
        nn->next = nn->prev = NULL;
        return nn;
}
void display(struct Node *head){
    while(head!=NULL){
        printf("%d\t", head->data);
        head=head->next;
    }
    printf("\n\n");
}

//insert at tail
struct Node *insert_tail(struct Node *head, int val){

}

//insert at head
struct Node *insert_head(struct Node *head, int val)
{

}

//insert at specific position
struct Node *insert_pos(struct Node *head, int pos, int x)
{

}

// delete node at head
struct Node *delete_head(struct Node *head)
{

}
// delete node at tail
struct Node *delete_tail(struct Node *head)
{

}

// search Element
int search(struct Node * head, int val){

}

//count no of nodes
int count(struct Node *head){

}

// Insert After
struct Node *insert_after(struct Node *head, int data, int val){

}

// Insert Before
struct Node *insert_before(struct Node *head, int data, int val){

}
```

```
// delete element
struct Node *delete_element(struct Node *head, int ele){

}
//insert element after given direct address to node.
void insert_after_addr(struct Node* curr_node, int val) {

}
struct Node *revList( struct Node *head)
{   /// BEGIN SOLUTION

}

int main(){
    struct Node *head = NULL;
    head = insert_tail(head, 1);
    head = insert_tail(head, 2);
    head = insert_tail(head, 3);
    head = insert_head(head, 4);
    display(head);
    head = insert_after(head, 2, 7);
    display(head);
    head = insert_before(head, 2, 9);
    display(head);
    head = insert_pos(head, 11, 6);
    display(head);
    head = delete_head(head);
    display(head);
    head = delete_tail(head);
    display(head);
    head = delete_element(head, 7);
    display(head);
    insert_after_addr(head->next, 22);
    display(head);
    head = revList(head);
    display(head);
    return 0;
}
```