Linked List

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

struct node {
     int data;
     struct node *next;
};


void insert(struct node *header,int data){
     struct node *temp,*ptr;
     temp = (struct node*) malloc(sizeof(struct node));

     temp->next = header->next;
     header->next = temp;
     temp->data = data;
}

void add(struct node* head, int data){
    struct node *temp =(struct node*) malloc(sizeof(struct node));
    temp->data  = data;
    temp->next = NULL;

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

void reverseList(struct node *h) {
     struct node *ptr, *after, *prev, *last;
     prev=h->next;
     last=prev;
     ptr=prev->next;
     while(ptr->next!=NULL)
     {
          after=ptr->next;
          ptr->next=prev;
          prev=ptr;
          ptr=after;
     }
     ptr->next=prev;
     h->next=ptr;
     last->next=NULL;
}
```

```
void delete(struct node *header, int key)
{

     struct node *ptr = header->next,*temp;
     temp = header->next;

     if(temp->data == key) {
          header->next = header->next->next;
          return;
     }
     while(ptr!=NULL){
          if(ptr->data == key){
               temp->next=ptr->next;
               free(ptr);
          }
          else{
               temp = ptr;
               ptr = ptr->next;

          }
     }
}

void display(struct node *header)
{
     struct node  *ptr = header->next;
     while(ptr!=NULL){
          printf("%d ",ptr->data);
          ptr = ptr->next;
     }
     printf("\n");
}

int search(struct node *header, int key)
{
     struct node  *ptr = header->next;
     while(ptr!=NULL){
          if(ptr->data == key){
               return ptr->data;
          }
          ptr = ptr->next;
     }
     return -1;
}
```

```
int length(struct node *header)
{
     struct node  *ptr = header->next;
     int count=0;
     while(ptr!=NULL){
          count++;
          ptr = ptr->next;
     }
     return count;
}

void sortL(struct node *head){
     int i=0, j=0;
     struct node *temp1=head->next, *temp = head->next;
     while(temp1 != NULL){
          while(temp != NULL){
               if(temp1->data < temp->data){
                    int T = temp1->data;
                    temp1->data = temp->data;
                    temp->data = T;
               }
               temp = temp->next;
          }
          temp1 = temp1->next;
          temp = head->next;
     }
}
int main(void) {
     struct node *header;
     header=(struct node *) malloc(sizeof(struct node ));
     header->next=NULL; //In Main after Allocating memory
     add(header,-23);
     add(header,10);
     add(header,20);
     add(header,40);
     add(header,50);
     add(header,30);
     display(header);
     delete(header,-23);
     delete(header,30);
     display(header);
     sortL(header);
     display(header);
     reverseList(header);
     display(header);
     return 0;
}
```

Output

-23 10 20 40 50 30 // after insert.
10 20 40 50 // ater delete.
10 20 40 50 // after sort.
50 40 20 10 // after reverseList.