

```
1: #include <stdio.h>
2: #include <stdlib.h>
3:
4: struct node {
5:     int key;
6:     struct node *link;
7: };
8:
9: void displayList(struct node * start){
10:     struct node *p;
11:     p = start;
12:     printf("\n");
13:     if (checkEmpty(p)){
14:         printf("List is Empty.");
15:     }
16:     else{
17:         while (p != NULL){
18:             printf("%d  ", p->key);
19:             p = p->link;
20:         }
21:     }
22: }
23:
24: int checkEmpty(struct node * start){
25:     struct node *p;
26:     p = start;
```

```
27:     if (p == NULL){
28:         return 1;
29:     }
30:     else {
31:         return 0;
32:     }
33: }
34:
35: int lengthOfList(struct node * start){
36:     int count = 0;
37:     struct node *p;
38:     p = start;
39:     if (checkEmpty(p)){
40:         return 0;
41:     }
42:     while (p != NULL){
43:         count++;
44:         p = p->link;
45:     }
46:     return count;
47: }
48:
49: void displayLength(int length){
50:     printf("Length of LinkedList is : %d", leng
51: }
52:
```

```
53: void main(){
54:     struct node *head = NULL;
55:     struct node *second;
56:     struct node *third;
57:
58:     head = (struct node *)malloc (sizeof(struct node));
59:     second = (struct node *)malloc (sizeof(struct node));
60:     third = (struct node *)malloc (sizeof(struct node));
61:
62:     head->key = 7;
63:     head->link = second;
64:
65:     second->key = 11;
66:     second->link = third;
67:
68:     third->key = 87;
69:     third->link = NULL;
70:
71:     displayLength(lengthOfList(head));
72:     displayList(head);
73: }
```