

```

1  #include <stdio.h>
2  #include <malloc.h>
3
4  struct node
5  {
6      int iData;
7      struct node *pNext;
8  };
9
10 void InsertLast(struct node **, int);
11 void Display(struct node *);
12 void DeleteAll(struct node **);
13
14 int main(void)
15 {
16     struct node *pFirst = NULL;
17     struct node *pSecond = NULL;
18
19     InsertLast(&pFirst, 10);
20     InsertLast(&pFirst, 20);
21     InsertLast(&pFirst, 30);
22
23     InsertLast(&pSecond, 40);
24     InsertLast(&pSecond, 50);
25     InsertLast(&pSecond, 60);
26
27     printf("First list is :\n");
28     Display(pFirst);                // 10 20 30
29     printf("\n\nSecond list is :\n");
30     Display(pSecond);              // 40 50 60
31     printf("\n\n");
32
33     DeleteAll(&pFirst);
34     DeleteAll(&pSecond);
35
36     return 0;
37 }
38
39 void InsertLast(struct node **ppHead, int iNo)
40 {
41     struct node *pTemp = NULL;
42     struct node *pNewNode = NULL;
43
44     pNewNode = (struct node *) malloc (sizeof(struct node));
45     if(NULL == pNewNode)
46     {
47         printf("Memory allocation FAILED\n");
48         return;
49     }
50
51     pNewNode->iData = iNo;
52     pNewNode->pNext = NULL;
53
54     if(NULL == *ppHead)
55     {
56         *ppHead = pNewNode;
57         return;
58     }
59
60     pTemp = *ppHead;
61     while(pTemp->pNext != NULL)
62         pTemp = pTemp->pNext;
63
64     pTemp->pNext = pNewNode;
65 }
66
67 void Display(struct node *pHead)

```

```
68 {
69     if(NULL == pHead)
70     {
71         printf("List is empty\n");
72         return;
73     }
74
75     while(pHead != NULL)
76     {
77         printf("|%d|->", pHead->iData);
78         pHead = pHead->pNext;
79     }
80 }
81
82 void DeleteAll(struct node **ppHead)
83 {
84     struct node *pTemp = *ppHead;
85
86     while(*ppHead != NULL)
87     {
88         pTemp = *ppHead;
89         *ppHead = pTemp->pNext;
90         free(pTemp);
91     }
92 }
93
```