

```
1 //
2 // Created by hfwei on 2022/12/16.
3 //
4
5 #include <stdio.h>
6 #include <stdlib.h>
7 #include "ll.h"
8
9 void Init(LinkedList *list) {
10     list->head = NULL;
11     list->tail = NULL;
12 }
13
14 bool IsEmpty(const LinkedList *list) {
15     return list->head == NULL;
16 }
17
18 bool IsSingleton(const LinkedList *list) {
19     return !IsEmpty(list) && list->head == list->tail;
20 }
21
22 int HeadVal(const LinkedList *list) {
23     if (IsEmpty(list)) {
24         return -1;
25     }
26
27     return list->head->val;
28 }
29
30 void Append(LinkedList *list, int val) {
31     Node *node = malloc(sizeof *node);
32     if (node == NULL) {
33         printf("Error: malloc failed in Append()\n");
34         return;
35     }
36     node->val = val;
37
38     if (IsEmpty(list)) { // empty list
39         list->head = node;
40     } else { // non-empty list
41         list->tail->next = node;
42     }
43
44     list->tail = node;
45     list->tail->next = list->head;
46 }
47
48 void Delete(LinkedList *list, Node *prev) {
49     if (IsEmpty(list) || IsSingleton(list)) {
50         list->head = NULL;
51         list->tail = NULL;
52     }
53 }
```

```
54 Node *cur = prev->next;
55 Node *next = cur->next;
56 prev->next = next;
57
58 // cur != list->head || cur != list->tail
59 if (cur == list->head) {
60     list->head = next;
61 }
62
63 if (cur == list->tail) {
64     list->tail = prev;
65 }
66
67 free(cur);
68 }
69
70 void Print(const LinkedList *list) {
71     if (IsEmpty(list)) {
72         return;
73     }
74
75     // iter: iterator
76     Node *iter = list->head;
77
78     do {
79         printf("%d ", iter->val);
80         iter = iter->next;
81     } while (iter != list->head);
82 }
83
84 void Free(LinkedList *list) {
85     while (!IsEmpty(list)) {
86         Delete(list, list->head);
87     }
88 }
```