```c
//
// Created by hfwei on 2022/12/16.
//

#include <stdio.h>
#include <stdlib.h>
#include "ll.h"

void Init(LinkedList *list) {
  list->head = NULL;
  list->tail = NULL;
}

bool IsEmpty(const LinkedList *list) {
  return list->head == NULL;
}

bool IsSingleton(const LinkedList *list) {
  return !IsEmpty(list) && list->head == list->tail;
}

int HeadVal(const LinkedList *list) {
  if (IsEmpty(list)) {
    return -1;
  }

  return list->head->val;
}

void Append(LinkedList *list, int val) {
  Node *node = malloc(sizeof *node);
  if (node == NULL) {
    printf("Error: malloc failed in Append()\n");
    return;
  }
  node->val = val;

  if (IsEmpty(list)) { // empty list
    list->head = node;
  } else {  // non-empty list
    list->tail->next = node;
  }

  list->tail = node;
  list->tail->next = list->head;
}

void Delete(LinkedList *list, Node *prev) {
  if (IsEmpty(list) || IsSingleton(list)) {
    list->head = NULL;
    list->tail = NULL;
  }

```

```c
  Node *cur = prev->next;
  Node *next = cur->next;
  prev->next = next;

  // cur != list->head || cur != list->tail
  if (cur == list->head) {
    list->head = next;
  }

  if (cur == list->tail) {
    list->tail = prev;
  }

  free(cur);
}

void Print(const LinkedList *list) {
  if (IsEmpty(list)) {
    return;
  }

  // iter: iterator
  Node *iter = list->head;

  do {
    printf("%d ", iter->val);
    iter = iter->next;
  } while (iter != list->head);
}

void Free(LinkedList *list) {
  while (!IsEmpty(list)) {
    Delete(list, list->head);
  }
}
```