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

#ifndef CPL_CODING_0_2022_CPL_12_LINKEDLIST_LL_LL_H_
#define CPL_CODING_0_2022_CPL_12_LINKEDLIST_LL_LL_H_

#include <stdbool.h>

typedef struct node {
  int val;
  struct node *next;
} Node;

// Invariants:
// (1) head always points to the first node
// (2) tail always points to the last node
typedef struct ll {
  Node *head;
  Node *tail;
} LinkedList;

void Init(LinkedList *list);

bool IsEmpty(const LinkedList *list);
bool IsSingleton(const LinkedList *list);

/**
 * @brief Get the val of the head of the linked list
 * @param list
 * @return
 */
int HeadVal(const LinkedList *list);
Node *Search(const LinkedList *list, int val);

/**
 * @brief Append a node with val to the tail of the linked list
 * @param list
 * @param val
 */
void Append(LinkedList *list, int val);
/**
 * @brief Delete the node next to prev from the linked list
 * @param list
 * @param prev the previous node of the node to delete
 */
void Delete(LinkedList *list, Node *prev);
/**
 * @brief Insert a new node with val next to prev in the linked list
 * @param list
 * @param prev
 * @param val
 */
```

```
54 void Insert(LinkedList *list, Node *prev, int val);
55
56 void Print(const LinkedList *list);
57
58 void Free(LinkedList *list);
59 #endif //CPL_CODING_0_2022_CPL_12_LINKEDLIST_LL_LL_H_
```