```c
#ifndef linked_list_included
#define linked_list_included

#include "list-node.h"

/****
 *
 * Type LinkedList defines an singly-liked list of integer-valued nodes.
 *
 */

typedef struct LinkedListStruct {

    /** Pointer to the head of the list */
    ListNode* head;

    /** Current number of elements in the list */
    int length;

} LinkedList;


/*-*
 * Functions.
 */

/**
 * Allocate a new empty list, with null head pointer and length = 0.
 */
LinkedList* newLinkedList();

/**
 * Insert the given node before the given index position i in the given list,
 * for 0 <= i <= list->length.  Do nothing if i < 0 or i > list->length.  If
 * node was inserted, increment list->length by 1.
 *
 * Note that i = 0 means the node becomes the head of the list; i =
 * list->length means the node goes at the end.  Any other legal value of i
 * means the node goes between the i-1th and ith nodes in the input list.
 */
void insert(LinkedList* list, ListNode* node, int i);

/**
 * Return the ith node in the given list.  Return null if the list is empty or
 * i < 0 or i >= list->length.
 */
ListNode* getIthNode(LinkedList* list, int i);

/**
 * Print to stdout the elements of the given list, comma separated, in list
 * order, with a newline at the end.
 */
void printList(LinkedList* list);

#endif
```