

```
1 #ifndef linked_list_included
2 #define linked_list_included
3
4 #include "list-node.h"
5
6 /**
7  *
8  * Type LinkedList defines an singly-liked list of integer-valued nodes.
9  *
10 */
11
12 typedef struct LinkedListStruct {
13
14     /** Pointer to the head of the list */
15     ListNode* head;
16
17     /** Current number of elements in the list */
18     int length;
19 } LinkedList;
20
21
22
23 /*-
24  * Functions.
25  */
26
27 /**
28  * Allocate a new empty list, with null head pointer and length = 0.
29  */
30 LinkedList* newLinkedList();
31
32 /**
33  * Insert the given node before the given index position i in the given list,
34  * for 0 <= i <= list->length. Do nothing if i < 0 or i > list->length. If
35  * node was inserted, increment list->length by 1.
36  *
37  * Note that i = 0 means the node becomes the head of the list; i =
38  * list->length means the node goes at the end. Any other legal value of i
39  * means the node goes between the i-1th and ith nodes in the input list.
40  */
41 void insert(LinkedList* list, ListNode* node, int i);
42
43 /**
44  * Return the ith node in the given list. Return null if the list is empty or
45  * i < 0 or i >= list->length.
46  */
47 ListNode* getIthNode(LinkedList* list, int i);
48
49 /**
50  * Print to stdout the elements of the given list, comma separated, in list
51  * order, with a newline at the end.
52  */
53 void printList(LinkedList* list);
54
55 #endif
```