```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "sqlist.h"

sqlink list_create() {
	//malloc
	sqlink L;

	L =(sqlink)malloc(sizeof(sqlist));
	if (L == NULL) {
		printf("list malloc failed\n");
		return L;
	}

	//initialize
	memset(L, 0, sizeof(sqlist));
	L->last = -1;

	//return
	return L;
}

/*
 * @ret   0-success   -1-failed
 * */
int list_clear(sqlink L) {
	if (L == NULL)
		return -1;

	memset(L, 0, sizeof(sqlist));
	L->last = -1;

	return 0;
}

int list_delete(sqlink L){
	if (L == NULL)
		return -1;
	free(L);
	L = NULL;
	return 0;
}

/*
 * list_empty: Is list empty?
 * para L: list
 * @ret  1--empty   0--not empty
 * */
int list_empty(sqlink L) {
	if (L->last == -1)
		return 1;
	else
		return 0;
}

int list_length(sqlink L) {
	if (L == NULL)
		return -1;
```

```
      return (L->last+1);
}

int list_locate(sqlink L, data_t value) {

      return 0;
}

int list_insert(sqlink L, data_t value, int pos) {
      int i;

      //full
      if (L->last == N-1) {
            printf("list is full\n");
            return -1;
      }

      //check para    0<=pos<=Last+1   [0, last+1]
      if (pos < 0 || pos > L->last+1) {
            printf("Pos is invalid\n");
            return -1;
      }

      //move
      for (i = L->last; i >= pos; i--) {
            L->data[i+1] = L->data[i];
      }

      //update value last
      L->data[pos] = value;
      L->last++;

      return 0;
}

int list_show(sqlink L) {
      int i;

      if (L == NULL)
            return -1;
      if (L->last == -1)
            printf("list is empty\n");

      for (i = 0; i <= L->last; i++) {
            printf("%d ", L->data[i]);
      }
      puts("");

      return 0;
}
```