```c
/* DList.h */
#ifndef _DLIST_H
#define _DLIST_H

#define LIST_FORWARD    0

//返回值：0--相同，>0--dataAddr<keyAddr，<0--dataAddr>keyAddr
typedef int  ListCmp(const void *keyAddr, const void *dataAddr);
typedef void ListFree(void *);
typedef void ListTraverseOp(void *, void *);

typedef struct list_node
{
    struct list_node *prev;
    struct list_node *next;
    char key[0];
}LISTNODE;

typedef struct
{
    int size;
    int keySize;
    LISTNODE head;
    ListCmp *cmpFn;
    ListFree *freeFn;
}LIST;

//链表的初始化
void ListNew(LIST *l, int keySize, ListCmp *cmpFn, ListFree *freeFn);
//获取链表的节点数量
int ListSize(LIST *l);
//链表判空
int ListEmpty(LIST *l);
//链表的销毁
void ListDispose(LIST *l);
//根据关键码查找所在节点中的数据地址
void *ListSearch(LIST *l, const void *e);
//链表关键码的插入，mode：0--头插，!0--尾插
//返回值：0--成功，!0--失败
int ListInsert(LIST *l, const void *e, int mode);
//链表删除关键码所在节点，返回值：0--成功，!0--失败
int ListRemove(LIST *l, void *e);
//链表删除关键码所在节点（无需深度删除关键码），返回值：0--成功，!0--失败
int ListRemoveU(LIST *l, void *e);
//链表的遍历
void ListTraverse(LIST *l, ListTraverseOp *traverseOpFn, void *outData);
#endif
```