

```

1  /* OpenHash.h */
2  #ifndef _OPENHASH_H
3  #define _OPENHASH_H
4  #include "SList.h"
5
6  //返回值: 0--相同, >0--dataAddr<keyAddr, <0--dataAddr>keyAddr
7  typedef int HashCmp(const void *keyAddr, const void *dataAddr);
8  typedef void HashFree(void *);
9  //将关键码转换成相应的桶地址
10 typedef int HashFunc(const void *);
11
12 typedef struct
13 {
14     LIST *listSet;
15     int capacity;
16     int size;
17     int keySize;
18     int valSize;
19     HashFunc *hashFn;
20     HashCmp *cmpFn;
21     HashFree *freeFn;
22 }HASH;
23
24 //散列表初始化
25 void HashNew(HASH *h, int capacity, int keySize, int valSize, HashFunc *hashFn,
26 HashCmp *cmpFn, HashFree *freeFn);
27 //获取散列表数据数量
28 int HashSize(HASH *h);
29 //获取散列表容量
30 int HashCapacity(HASH *h);
31 //散列表销毁
32 void HashDispose(HASH *h);
33 //散列表读取
34 void *HashGet(HASH *h, const void *e);
35 //散列表插入
36 int HashPut(HASH *h, const void *e, const void *val);
37 //散列表删除
38 int HashRemove(HASH *h, void *e);
39 #endif

```