

Lab 8

Hashing

Build an **Hash Table** data structure that can accommodate a maximum of 100 hash node elements.

Provide the following routines, functions or methods:

A structure or object definition for the hash node element.

-Your Hash Table should be initially of size zero.

-The Hash Table should employ a double LinkList (which calls on a common LinkedStack).

A program that encapsulates routines to maintain the hashing structure:

-determine a suitable hashing key,

-build a set of N hashkey pointers,

-maintain linked lists off of each hashkey pointer,

-attach/detach information nodes to the proper list based on hashing key,

-query the linklist contents of any hashkey (traverse),

-find a particular node element based upon the primary key,

-query the current size of each hashkey list and the entire hash structure.

A user interface routine that requests the service of the hashing program:

-prompts and accepts each integer entry.

*-routine to **display all** the entries in the hash table for a given key.*

*-routine that will **find** a particular element in the hash table given its value.*

*-routine that will **delete** a particular element from the hash table.*

Test each of the above methods used to manipulate the data structure. Your program should employ a menu that repeatedly prompts the user to perform the above functions.

Demonstrate that it works for the structure's "boundary conditions".

Input 100 numbers between 0 and 1000 randomly to the structure.

(Instructor will provide the input)

Show the elements of each linked list associated with a particular key.

Show the number of elements of each linked list associated with a particular key.

Demonstrate adding/finding/deleting a particular element.

The outputs should look something like the following:

Information element X stored in hashtable at hashkey K.

Hash Key of K has M members in its list.

Element X, node N, is chained to prior node O, in list anchored by hashkey K.

Element X found at hashkey K.

Element X is removed from list of hashkey K.