Dillon Li (1642853)

Assignment 4

Unit 5: Hashes

Submitting: 9/19/16

Brief Description of Assignment

Implementing hashtables in C++. The implementation must work for sizes of 17 and 7 when 11 values are placed into the hash (collision handling must be implemented).

Brief Description of Logic

The HashEntry class is implemented based on the already implemented HashMap class. I changed the constructor for HashMap so that it accepts size as an input. HashEntry has 3 variables, a constructor, and getters for two of the variables (the third variable is a next pointer for linked list).

The HashMap class has the table of HashEntry pointers, and I added a size variable as well since I added one to the constructor.

The HashMap class has most of the functions implemented/used. The constructor and get function are already implemented in the given code. The insert function uses similar hashing logic as the get function in order to insert into the hash table, using value % size as the hash (I made value and key the same, since a particular key was not specified). If the bin being inserted already has a value in it, the value is instead inserted as the next node of that particular bin.

Insert(int value):

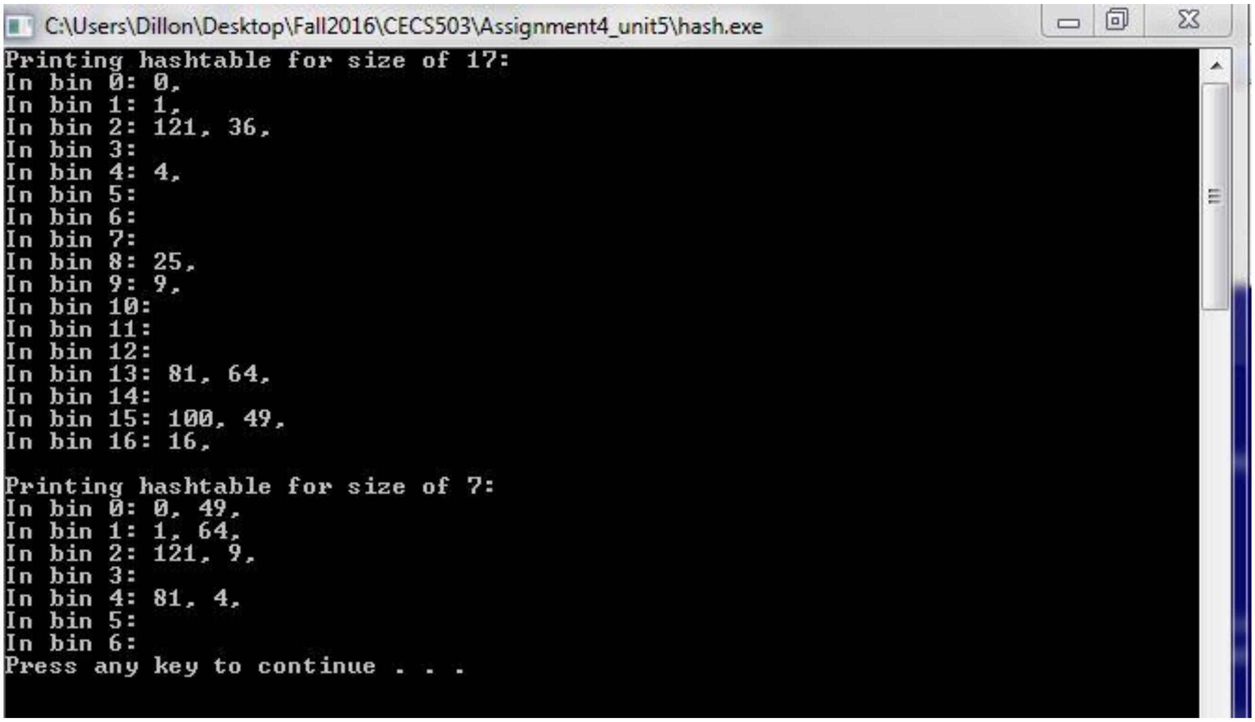
Input: value – the value to be inserted into the hashtable.

Output: none technically, but the table itself is modified.

Void Printhash():

This function is just a nice way to print out all the respective hash bins and their contents. This is what I used to verify that the functions were implemented correctly.

Picture of output:


C:\Users\Dillon\Desktop\Fall2016\CECS503\Assignment4_unit5\hash.exe

```
Printing hashtable for size of 17:
In bin 0: 0,
In bin 1: 1,
In bin 2: 121, 36,
In bin 3:
In bin 4: 4,
In bin 5:
In bin 6:
In bin 7:
In bin 8: 25,
In bin 9: 9,
In bin 10:
In bin 11:
In bin 12:
In bin 13: 81, 64,
In bin 14:
In bin 15: 100, 49,
In bin 16: 16,

Printing hashtable for size of 7:
In bin 0: 0, 49,
In bin 1: 1, 64,
In bin 2: 121, 9,
In bin 3:
In bin 4: 81, 4,
In bin 5:
In bin 6:
Press any key to continue . . .
```