CS 2420 Program 4 – 20 points  
Fall 2019

Déjà vu – Hash Table

**Part 1: Becoming familiar with the code**

HashTable code has been given to you. No testing program has been provided. To become familiar with how the code works, try reading in a small input file and make sure you can create a hash table of those entries.

**Testing:** Make sure the following works:

a. Insert values

b. Delete values

c. Find values

d. Printing the contents of the hash table.

e. Control the size of the hash table.

What happens if you attempt to delete an item that isn’t there?

What if you add more things than can fit into the hash table?

**Modification:** You have been given the hash table code from your text for doing quadratic probing. Modify the code so it takes the key and the associated object as two parameters.

Please retain the generic structure of the hash table. Be careful not to add anything to the hash table that only makes sense for this specific problem.

Part 1 is for your benefit. The code you turn in does not need to show the results of this experimentation.

**Part 2 Using the code to solve a bigger problem**

This assignment is designed to give you experience with hash tables.

Sentiment Analysis: the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product, etc., is positive, negative, or neutral.

The data that the algorithm is going to “learn” from is a set of 8,528 movie reviews in which the sentiment of each review has been manually rated on a scale from 0 to 4. The sentiment labels are:

0 - negative

1 - somewhat negative

2 - neutral

3 - somewhat positive

4 - positive

The assignment is to use the provided data to develop an algorithm that will allow a user to input a new review and will automatically score the sentiment of the review.

The program will require that you

* Read in a review, converting to lower case.
* Assign each word in the review the score attributed to the review
* Enter a WordInfo object (consisting of the word, total score, and number of occurrences) into a hash table. If word already exists in the hash table, update the score and number of occurrences.

The program should prompt the user to input a movie review, and automatically score the review based on the average score of the meaningful words in the review. If a word occurs (on average) in more than 10% of all reviews, ignore that word as being “meaningless”. A composite score between 0 and 1.75 is considered negative. A score between 1.75 and 2.25 is considered neutral. A score above 2.25 is considered positive.

Example output:

REVIEW: A weak script that ends with a quick and boring finale

The review has an average value of 1.46

Negative

REVIEW: Awesome Loved every minute of it

The review has an average value of 2.47

Positive

REVIEW: I hated this Poke my eyes out

The review has an average value of 2.13

Neutral

REVIEW: Loved loved loved it

The review has an average value of 2.67

Positive