Corey McFarland
Assignment 5.3

Question 1: Two words that would hash the same in function 1 but not in function 2. Why?

Since the first function builds a hash based on the sum of the Unicode number of each letter, the following would hash the same:

"are": (97 + 114 + 101 = 312)
"ear": (101 + 97 + 114 = 312)

Under the second function, the index multiplier would modify them in the following way:

"are": (1 * 97 + 2 * 114 + 3 * 101 = 97 + 228 + 303 = 648)
"ear": (1 * 101 + 2 * 97 + 3 * 114 = 101 + 194 + 342 = 637)

Because the index multiplier adds an additional level of complexity, you can create more unique values. If the keys were all 3 letter lowercase words, function 1 would have a hash range of 291 to 366 (or 70 potential unique values) and function 2 would have a hash range of 582 to 732 (or 150 potential unique values). This shows that for the 3 letter lowercase words example, an additional level complexity more than doubles the possible unique values for the hash to create.

Question 2: Why does the above make function 2 better than function 1?

With more possible unique values, you can reduce the number of shared indexes. This reduces the time needed to iterate through the LinkedList implemented in our case when searching for values to update, remove, or confirm they exist as the LinkedLists would overall be shorter.

Question 3: With the same input, is it possible to have different values of empty_bucket() and table_load() for function 1 and function 2?

For empty_bucket(), the value can differ as different keys for the same word and capacity can result in the word being placed in different buckets for either function. This can result in more or less empty buckets between the functions.

For table_load(), this would not change as the load factor is only dependent on the capacity of the hash table and the number of keys in the hash table. While function 2's increased possible unique values allows it to better utilize larger capacities, the variables for the load factor are the same for both functions causing both functions to have the same load factor.