Vinay Bikkina

1. Two examples would be EAT and ATE. Or GOAT and TOGA.
2. It will make it so the keys stored are more unique and spread out in the hash table.
3. No because the size measures the number of keys stored in the hash table, which will be the same no matter which hashing function you use.
4. tableLoad() depends on the size and capacity. Both of which are the same no matter what hashing function you use. Therefore, the tableLoad function will return the same value no matter what hashing function you use.
5. The number of empty buckets will be greater for StringHash1 than stringHash2 because the stringHash2 will have more spread out hashs, so that the keys use different buckets so that the read/write operations can still be in O(n) time.
6. When I tried it out, when the bucket size was changed to prime number, the buckets used size went down.
7. The bigger the table size, the faster the Concordance code is running. I tried with table sizes of 1, 10, 100, 1000, and the speeds get faster with more table size.