Sihao Xu (sihaoxu) **Lab 2**

Part 1

Firstly, I implement the insert function which can ensure that there will not be the record with same key value as formal ones. Then in the remove function, it can make the key value be “deleted” when it is removed. Plus, I use open addressing with double hashing as the hashing strategy and use multiplicative hashing as the hashing function.

Part 2

a. I extend the record the record class with a hashKey variable to store the value produced by the toHashKey function. And I embed the toHashKey function into the record class so that the key String value can be transfer to a int value HashKey. Thus, it can use the record.hashKey to run hashing function and compare the hashKey value first rather than run toHashKey function every time. When the hashKey is the same, then I continue to compare the String key. Because it does not ensure the correctness if only compare the int type hashKey value. As a result, our program will speed up.

b. My improved implementation works, and it satisfy all the test case. The constructor function will create a table with two slots. In the insert function, when the table is full, it will create a new table with double slots and rehash and reinsert the record into the new table. Finally make the new table replace the old table.