In my basic hash table I used an array to store each of my records. In order to also store the base hash key with each record I simply created a second array and stored each base hash key at the index of its corresponding record. I checked to see if the base hash keys of two records were the same whenever I would normally check to see if the records were the same record. If the keys were the same, then I would do a full comparison (since two different records could theoretically hash to the same thing), but if not then I would skip the full comparison as the lab said.

To make my hash table extensible I added a counter (n) that stores the number of elements in the table at any given time. This counter increases whenever an element is inserted and decreases if an element is removed. The table doubles in size whenever n is 1/4 the size of the table or more in order to ensure that the load factor of the table remains at or below 1/4 whenever possible. The table starts at size 1 but increases in size fairly quickly since even one element in the table means that the size needs to increase to 4 to maintain the load factor at or below 1/4. Whenever I copy the table into a new table I first store the old table locally, then iterate through it to populate the new table, using the insert method I already created.