Quadratic Probing: C1 = 1, C2 = 1, M = 2250, Average Comparisons = 1.942

Quadratic Probing: C1 = 2, C2 = 1/2, M = 2250, Average Comparisons = 2.0235

Quadratic Probing: C1 = 5, C2 = 3, M = 2051, Average Comparisons = 1.739 (I chose C1 and C2 in this case)

Double Hashing: M = 2251, h’ (k) = (key%M) h’’(k) = ((key+5) % M),
Average Comparisons = 1.932

Double Hashing : M = 2251, h’(k) = (key%M) h’’(k) = (((int) (Math.sqrt(key)))%M),
Average Comparisons = 1.5025

Double Hashing : M = 2251, h’(k) = (key%M) h’’(k) = MultiplicationMethod ,
AverageComparisons = 2.684

Multiplication Method

```
double V = (Math.sqrt(5)-1)/2; // Honour to Donald Knuth for inventing this good value for V. Referenced from Robin Dawe's code!
double wholeNumber = (V*key) - (long)V;
double remainder = wholeNumber - (long)wholeNumber;
int multiplicationMethod = (int)(M*remainder);
```

The results show that Quadratic Probing is more efficient than Double Hashing (which goes against my hypothesis that Double Hashing would be better for having a smaller Hash Table). In this case, we filled a table of size 2051 with Quadratic Probing which could of have not been achieved with Double hashing (with at least my hash functions). The hash function you choose heavily influences the average number of comparisons to find an available index in a hash table. As you can see I got a low 1.739 number of comparisons with quadratic probing which is just the 2nd lowest, under the double hashing function using the square root method. The main difference between my double hashing method and quadratic probing method was the usage of the hashing functions, and the constants chosen in quadratic probing. Double hashing incorporates it in two separate terms while quadratic uses it in one, which demonstrates that the hashing function will mainly dictate the average number of comparisons needed. The table was full using quadratic probing which may suggest that double hashing is weaker than Quadratic Probing when trying to compare the most unique addresses in the hash table, and that hashing functions play a significant role in attaining a key’s identity.