Homework 4 - Hashing

150 Points
Due: Tuesday Nov 1st

Grading:

120 Points for implementing linear probing
20 Points for implementing quadratic hashing (must also complete linear probing)
5 Points for test cases
5 points for comments / style

Non-compiling programs will receive 0 points.

Goals:

Implement linear probing to receive a “B” grade.
Implement quadratic probing to receive an “A” grade (must also complete linear probing)
Provide sufficient test cases to show that your program operates correctly.

Specifications:

Implement a hash table using vectors of strings. Strings are hashed based on the ASCII sum of their characters. See the provided starter code for the API’s you need to implement. You may add additional helper functions as needed but need to complete all the unfinished functions in the starter code.

To implement quadratic hashing you’ll need to create another class object that inherits from the Hash.h file, and change it to perform quadratic hashing. Call this method of hashing from main as part of your tests to show that it works. You’ll also need to modify the make file to include the extra class file.

When “showing” that your hash function works properly, provide tests that print out the table before and after insertions, interesting cases such as collisions, insertions on elements that were previously deleted and marked as available, looking up entries that must traverse past a deleted-element, insertions that wrap around the table end, and rehashing. Think of cases that test your hash functions robustness and showcase these with your tests.