Computer Science II — Homework 9 — Hashing with Quadratic Probing

This assignment is due **Tuesday, May 1** at 11:59:59pm and is worth 60 points toward your homework grade. You must submit your header file and a `readme.txt`

Your problem is to reimplement the `hash_set` using quadratic probing instead of the separate chaining implementation from Lecture 20 and Lab 11. The algorithm for quadratic probing is described in Lecture 19. Here are several specific details:

- Assuming that i is the hash location for a particular key and N is the size of the hash table vector, then when a collision occurs, the sequence of locations checked should be

  $$i{+}1 \mod N, i{+}3 \mod N, i{+}6 \mod N, i{+}10 \mod N, \ldots, i{+}j(j{+}1)/2 \mod N, \ldots$$

  until an empty location is reached.

- Initialize the hash table to be size 8 and every time the hash table must be reallocated, double its size. In this way, the hash table size will always be a power of 2. Combining a power-of-two table size with the above addressing scheme ensures that an empty location will always be found (eventually) if the table is not completely full.

- Reallocate the table when it is at least 80% full. You can play with other values, especially closer to 100% to see the effect on the performance of the algorithm.

- When a key is removed from the table, mark the location as "erased". When you are searching for a key you must skip over these locations. However, you should reuse these erased locations. In particular, when inserting a key, your program should first look to see if the key is already in the hash set. If it is not, your program should insert the key in the first "erased" or "empty" location it reaches.

The public interface to the `hash_set` should be identical to the previous versions. An initial example main program that uses this `hash_set` is posted on the course website.