ADS216 Tutorial 6
Roberto Togneri, 2000

1. Implement the `insert(Item item)` and `search(Key v)` methods for hash table linear probing assuming the global hash table is defined as `Item *st; st = new Item[M]`, where `M` is the size of the table and `hash(item.key(), a, M)` is the hash function. Both `st`, `M` and `a` are private (i.e. static) members of the hash table class. What happens when the hash table is full? Does this problem also occur with separate chaining?

2. Direct removal of an item from a hash table with linear probing requires rehashing affected keys. Implement the `remove(Item item)` method and re-implement the `insert(Item item)` and `search(Key v)` methods from the previous question by defining a special `nullItem` sentinel key so that rehashing is not necessary.

3. Give the contents of the hash table that results when you insert items with the keys E A S Y Q U T I O N in that order into an initially empty table of `M = 16` lists, using linear probing. Use the hash function `11k mod M` to transform the `k`th letter of the alphabet into a table index.

4. Give the contents of the hash table that results when you insert items with the keys E A S Y Q U T I O N in that order into an initially empty table of `M = 16` lists, using double hashing. Use the hash function `11k mod M` for initial probe and the second hash function `(k mod 3) + 1` for the search increment (when the key is the `k`th letter of the alphabet).