5613B - Introduction to C++
Hilary Term - 2014-2015
Homework 3 - Due Apr. 1st, 2015

1. Implement a templated `UniquePointer` class. This class behaves as an ordinary pointer, except it prevents multiple pointers to the same object in memory. `UniquePointer` should have the following interface:

   - a default constructor.
   - a constructor from an ordinary pointer.
   - a copy constructor and copy assignment operator from another `UniquePointer` of the same type. Note that this should 'move' the pointer from the input object to the current one. If you wish to use C++11, this should be implemented as a move constructor, with the copy constructor disabled.
   - overloaded `*` (dereference) and `->` (member function access) operators which behave like a normal pointer.

2. Implement a templated `SharedPointer` class. This class behaves like an ordinary pointer and allows multiple pointers to the same memory address. However, it also tracks the number of `SharedPointer`'s to this single memory block and will only free the memory (via `delete`) once the last one of these pointers has been destroyed. `SharedPointer` should have the following interface:

   - a default constructor.
   - a constructor from an ordinary pointer.
   - a copy constructor and copy assignment operator from another `SharedPointer` of the same type.
   - overloaded `*` (dereference) and `->` (member function access) operators which behave like a normal pointer.

   Make sure to test the functionality of both `UniquePointer` and `SharedPointer` to ensure that they perform as intended.

3. Implement a 'social network' consisting of some number of people ('nodes') each of which has a number of friends (other nodes). Start the network with n people that are all friends with each other. Then, considering each person in the network, perform one of the following actions (chosen at random with equal probability):

   - 'unfriend' one friend.
   - if possible, unfriend one friend and add a new friend from amoung the network members.

- add a new node to the network and become mutual friends with it.

If a node has no friends, it is removed from the network. Repeat this 'sweep' through the network 100 times. Find the minimum value of n such that the network contains at least one person after these 100 sweeps. Hint: you may find the `SharedPointer` class useful.