Homework 6

Software Systems **Allen B. Downey**
Spring 2008

Due: Thursday 1 May

The purpose of this assignment is to practice implementing data structures in C.

FIFO Queue

A queue is a data structure that provides methods like `add` and `remove`, where `add` puts a new element into the queue, and `remove` takes one out.

In a FIFO queue `remove` returns the element that was added least recently. Other types of queues behave differently.

One of the more efficient implementations of a FIFO queue is a double-headed linked list, so named because it keeps pointers to the first and last nodes in the list, which makes it possible to add to the end or remove from the beginning in constant time.

1. Read Chapter 17 of the Cow Book and the Wikipedia page on linked lists, `http://en.wikipedia.org/wiki/Linked_list`.
2. Write type definitions for `Queue` and `Node`, where a Queue contains pointers to the first and last nodes, and each Node contains a string as cargo and a pointer to the next node.
3. Write "constructors" for each type.
4. Write print functions for each type and test your code.
5. Write `queue_empty` to test whether the queue is empty.
6. Write `queue_add`, which takes a string and adds it to the end of the queue.
7. Write `queue_peek`, which returns the string at the front of the queue but doesn't modify the queue.
8. Write `queue_remove`, which removes the node at the head of the queue and returns the string it contains, or `NULL` if the queue is empty. Make sure you free the node you removed!
9. Write code that tests these functions.
10. Following the example on page 315 of the Cow Book, break your code into separate files and write a Makefile that compiles them separately.
11. Optional: make your queue implementation thread safe. See `http://en.wikipedia.org/wiki/Thread-safety` and `www.megacoder.com/files/presentation/Thread-Safe_Programming.pdf`.

    Hints:

    - No, `malloc` is not thread safe.
    - The simplest way to make your queue thread safe is to make all access to it exclusive. But if you want to get fancy, you could allow concurrent access for readers.