Homework 2

Due: 22 Feb 2017

Problem 2.1

Consider the task of writing functions

```
bool anyEmpty (NodeType* node)
bool allEmpty (NodeType* node)
```

that determine whether any of the strings in a given linked list are empty strings (i.e. `""`), and whether *all* of the strings in a given linked list are empty strings. (Assume for this problem that the relevant item type is `string`.)

For the moment I'm more interested in the algorithm, so you're better off writing it out by hand (and I won't fuss over things like semicolons). You should still test your work by tracing it on a concrete example!

Problem 2.2

Consider the following code, which assumes that the relevant item type is `int`:

```
NodeType* a = new NodeType(5, nullptr);
NodeType* b = new NodeType(8,
                  new NodeType(13,
                     new NodeType(21, a)));
a->setNext(b);
```

Draw a diagram of memory after those statements are executed, and describe as specifically as you can why that state of affairs can cause problems. Also comment on how you could detect inside a program if this type of thing had happened.

(Hint: think back to Lab 4, and specifically about what happened with the line `Card* d = c;` and how `==` worked with pointers.)