

Homework 2

Due: 1 Oct 2014

Problem 2.1

Consider the task of writing a function

```
int sum (Node<int>* node)
```

that computes the sum of the values in and attached to the given node. Similar to how we did for `contains`, write both an iterative and a recursive version.

For the moment I'm more interested in the algorithm, so you can write it out by hand (and I won't fuss over things like semicolons), but if you do that you should still test your work by tracing it on a concrete example.

Problem 2.2

Consider the following code:

```
Node<int>* a = new Node<int>(5, nullptr);
Node<int>* b = new Node<int>(8,
    new Node<int>(13,
        new Node<int>(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.)