Programming Exam Choice 20

Start by downloading the question 20 files from the moodle into a new directory on your machine. You will be changing the code in the question20.cpp file. You have a Makefile to handle building this code.

Part 1. 5 points

Imagine a circular doubly linked list, where the nodes connect in a ring, so that head_ptr->prev is tail_ptr and tail_ptr->next is head_ptr. These lists don't contain nullptrs. We can start at any node, and eventually circle back to the starting node by following the next pointers.

What if you don't know if your doubly linked list is circular or not? Suppose that all you are given is a node somewhere in the doubly linked list, and the list may be circular or linear. You could follow the links from this node until you find a nullptr, and decide that it's not circular. Or you could follow the links until you returned to your starting node, and decide that it's circular. Empty lists aren't circular. Lists with one node can be circular (check for this) or linear.

You will complete this function to tell if a list is circular:

```
bool circular_list_simple(node* some_ptr)
{

}
```

My code generates a bunch of lists that are linear or circular, and prints your answer for each list.

Part 2. 5 points

In part 1, it's hard to decide when we can quit checking long lists for circularity. There's a smarter way to tell circular and linear lists apart, by using 2 node* cursors from our starting node. We walk cursor 1 forward one node at a time. We walk cursor 2 forward 2 nodes at a time. Be careful! You may want to check something as you do this.

If cursor 2 catches up to cursor 1, then the list is circular. If cursor 2 finds the end of the list, the list is linear. Implement this logic for part 2.

```
bool circular_list_smarter(node* some_ptr)
{

}
```

Upload your question20.cpp file to the moodle when done. Your TA may ask you to zip other files in as well.

Logic of problem laid out in comments:	50%
Code compiles with no errors or warnings:	10%
Code has no run time errors:	10%
Code gives correct answers for all inputs:	20%

Code is clean and easy to read:	10%