Swapping Nodes in a Doubly Linked List

Due Tuesday by 11:59pm **Points** 60 **Submitting** an external tool

Given a sequence of nodes (struct **Node**) arranged in a doubly linked list containing a "dummy" node at each end (head and tail), implement the following function **void swapWithNext(Node * p)** to swap the node pointed to by **p** and its next node by **only adjusting the pointers linking the nodes** (the data itself must not be swapped).

For example given the following list:

(head node) <-> 1 <-> 2 <-> 3 <-> (tail node)

Using the helper function **getNode(head, 1)** (where head is a pointer to the head node) will return a **Node*** pointing to the second node containing a value (<-- 2 -->). Passing this pointer to **swapWithNext()** will swap this node with the next node resulting in the following list:

(head node) <-> 1 <-> 3 <-> 2 <-> (tail node)

This will happen only by updating the **next** and **prev** pointers; there should be no assignment to the node's **value**. Additionally, the swap should only take place if both the node being pointed to and its next node both contain values so it should not be possible to swap the head node with the first value-containing node or the tail node with the last value-containing node.

Download the starter code and use **doublyLinkedSwap.h** for the definition of the **Node** struct along with the prototype for **swapWithNext()** as well as some helper functions to help you with creating and viewing the doubly linked list. You should write the implementation for **swapWithNext()** (and any main function you use to test it) in **doublyLinkedSwap.cpp**.

Swapping Nodes in a Doubly Linked List Starter Code.zip
(https://clemson.instructure.com/courses/198274/files/17721166?wrap=1) 
(https://clemson.instructure.com/courses/198274/files/17721166/download?download_frd=1)

Doubly Linked Swap Sample Test Cases.zip
(https://clemson.instructure.com/courses/198274/files/17721176?wrap=1) 
(https://clemson.instructure.com/courses/198274/files/17721176/download?download_frd=1) - Main functions for a subset of the test cases on Gradescope (**main() function in doublyLinkedSwap.cpp must be commented out**): T#.cpp is the main function for test case #, returns 0 when test case passes, returns 1 when test case fails.