Question 7 - 20 PTS

A.) It is often useful to randomize the order of a list of elements. Implement a function in C++, called Shuffle() that takes as a parameter the Node pointer to the head of a doubly-linked list.
The function should split the list in half (rounding down) and then interleave nodes from the first half with nodes from the second half, see the example below.

Write good working C++ code to implement this function.

An empty list and a list with a single Node will remain unchanged:
An original list:

1 <-> 2 <-> 3 <-> 4 <-> 5 <-> 6 <-> 7 <-> 8 <-> 9

Splits to:

1 <-> 2 <-> 3 <-> 4 5 <-> 6 <-> 7 <-> 8 <-> 9

and will become:

1 <-> 5 <-> 2 <-> 6 <-> 3 <-> 7 <-> 4 <-> 8 <-> 9

The Node structure may contain multiple data elements of different types as well as a Node pointer named next and a Node pointer named prev.

Note: You must change the order of the Nodes in the list and not the values within the Nodes.

B.) What is the time complexity of your solution for a doubly-linked list with N elements?

C.) You think this might be easier if the values were stored in an array instead of a doubly-linked list, explain why or why not. Show your reasons with code or psudo-code demonstrating how you would shuffle an array of elements instead.

D.) What is the time complexity of this solution for an array with N elements?