After graduating from MSU, you're a full time programmer, using a library for doubly linked lists that someone from 30 years ago in your company developed for internal use. It turns out that this library lacks some essential functionality.

You must complete one function: ***remove_middle***

remove_middle(data: DLL) -> DLL:
- Removes the middle node(s) from a doubly linked list. If the list is of odd length, this will be one node. If the list is of even length, this will be two.
- param **data**: A doubly linked list, defined by the class provided
- return: the modified DLL
- **MUST NOT** use any extra containers, such as python lists or dictionaries. Use of an extra container will result in a 0 for the problem.
- **MUST NOT** instantiate a new list, instead modify and return the existing list

Examples:
1 <-> 2 <-> 3 <-> 4 <-> 5
->
1 <-> 2 <->4 <-> 5

___
1 <-> 2 <-> 4 <-> 5
->
1 <-> 5

___
1 <-> 2
->
Empty List

___
1
->
Empty List

___
Empty List
->
Empty List