Submission Instructions Please a single single text file (.py) with a single python file that contains the two classes ‘LinkedList‘ and ‘Deque‘. Please name the file as follows:

`tcss501-hw02.py`

Countable and Reversible Linked List (5pts)

Question 1: 10pts (5pts each function) Starting with the provided LinkedList class, add two methods with the following definitions.

```
def charCount(self, aggregated=False):
    """ Returns the total number of characters contained in the nodes of the linked list.
    If the ‘aggregated’ argument is True, the result should be a single integer
    representing the sum of character counts of all elements in the LinkedList.  If False,
    the result should be a list of integers representing the count of characters of each
    node.  In both cases, if the list is empty, return ‘None’.  The aggregated version of
    this method should be O(1) and the dis-aggregated version should be O(n).

    :param aggregated: If True, aggregate the counts into a single value. If False, return
    a list of counts.
    :return: A single integer or list of integers as described above.
    """


def reverse(self, in_place=False):
    """ Reverses the order of elements of the list, either in place (modifies the existing
    list) or replicates and returns a new copy of the list.  If in_place==True then
    modifications should be made to ‘self’. If in_place==False do not modify self, but
    rather create a new list and return the new list to the caller.  If the list is empty,
    in_place==True should do nothing, and in_place==False should return a new empty
    LinkedList.  Reverse should be no worse than O(n).

    :param in_place: If True, operations are performed on this instantiation of the List.
    Returns a new reversed version of the list otherwise.
    :return: If in_place == True, returns None, otherwise returns a new LinkedList object.
    """
```

For charCount, you can assume that the data provided will be a string. Throwing a TypeError exception is acceptable when attempting to measure the length of ‘data’ that is acceptable. Also, you can consider the data of a node to be immutable. Meaning, once it is set, it won’t be changed. This means that if a node is created with data of ”Hello” and is later changed to ”ThisIsLonger” you needn’t consider this case for your charCount implementation.

Please add any additional fields or methods to either the inner DoubleLinkNode class or the LinkedList class to help.

Question 2: 5pts Using the provided Queue class as a starting point, refactor to be a Double Ended Queue as a class named 'Deque'. A double ended queue works in a very similar way of a classic Queue object, but contains the following methods.

```
def addFirst(self, data):
""" Creates a new node at the beginning of the queue, containing the value of data.

:param data: The data to be contained in the newly created element.
:return: None
"""

def addLast(self, data):
""" Creates a new node at the beginning of the queue, containing the value of data.

:param data: The data to be contained in the newly created element.
:return: None
"""

def removeFirst(self):
""" Returns the data of the node that is at the "start" of the Deque.
:return: The data stored in the node that is at the "start" of the Deque.
"""

def removeLast(self):
""" Returns the data of the node that is at the "end" of the Deque.
:return: The data stored in the node that is at the "end" of the Deque.
```