Merging two queues V45106_en

In the `public_files` section of the problem statement, a class called **LinkedQueue**, which implements the **Queue ADT** using a singly-linked list, is defined. Extend the implementation of this class with a new public method **merge(self, other)**. This method merges the elements of two queues $q1$ and $q2$ as follows: If $q1$ represents the queue $e_1, e_2, \ldots, e_n$ and $q2$ represents the queue $o_1, o_2, \ldots, o_m$, after executing `q1.merge(q2)` queue $q1$ represents

- the queue $e_1, o_1, e_2, o_2, ...e_n, o_n, o_{n+1}, \ldots, o_m$, if $n \leq m$; or
- the queue $e_1, o_1, e_2, o_2, ...e_m, o_m, e_{m+1}, \ldots, e_n$, if $n > m$.

In both cases, after executing `q1.merge(q2)`, the queue $q2$ is empty, because its elements have been transferred to $q1$.

For example, if **q1** is an instance of the class `LinkedQueue` that represents the *queue*

```
front 1, 3, 5, 7 back
```

and **q2** is an instance of the class `LinkedQueue` that represents the *queue*

```
front 2, 4, 6 back
```

after executing the statement **q1.merge(q2)**, **q1** will represent the *queue*

```
front 1, 2, 3, 4, 5, 6, 7 back
```

and **q2** will be empty.

Your implementation of `merge(self, other)` should not use any public method of the class `LinkedQueue`. It should work direclty with the representation of the class (i.e. attributes of the classes `LinkedQueue` and `_Node`), and it should not create any new node.

You should also override the *special method* `__str__` of the class **LinkedQueue** so that the contents of an instance of this class representing a queue of integer numbers can be printed without making any call to the public method **dequeue**.
In particular, you should add the following public methods to the **LinkedQueue** class:

```
def merge(self, other):
  # Insert your implementation below


def __str__(self):
  # In the implementation of this method, assume the queue instance
  # can only contain integer numbers. This is only true in the context
  # of this problem.
  # Insert your implementation below
```

Sample input

```
1 3 5 7
2 4 6
```

Sample output

```
v[0]: 1 3 5 7
v[1]: 2 4 6
```

```
After calling v[0].merge(v[1])
v[0]: 1 2 3 4 5 6 7
```

```
v[1]:
```

Problem information

Author :
Generation : 2024-10-12 13:38:20