Jutge.org
The Virtual Learning Environment for Computer Programming

Add rotate method to class LinkedQueue T43800_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 **rotate()**. This method returns the element at the front of the queue and moves it to the back of the queue.
For example, if **q** is an instance of the class `LinkedQueue` that represents the following *queue*

```
front 5, 2, -1, 3, 0, 8, 6 back
```

after executing the statement **x** = **q.rotate()**, **x** will be **5** and the object **q** will represent the *queue*

```
front 2, -1, 3, 0, 8, 6, 5 back
```

Although the effect of **rotate()** on a non-empty queue **q** is the same as the combination

```
{\bf q.enqueue(q.dequeue())}
```

your implementation should be more efficient than making two separate calls.

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 rotate(self):
  # 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

```
5 2 -1 3 0 8 6
```

Sample output

```
5 2 -1 3 0 8 6
rotate returns: 5
2 -1 3 0 8 6 5
```

Problem information

Author :
Generation : 2024-10-11 18:00:56