
Queue Concatenation**X60068_en**

Modify the **LinkedList** class (included in the `public_files` section of the problem statement) by implementing a new method called **concatenate**. A call to `concatenate` of the form

```
q1.concatenate(q2)
```

takes all elements of the `LinkedList` instance `q2` and appends them to the end of the `LinkedList` instance `q1`. The operation should run in $O(1)$ time and should result in `q2` being an empty queue.

You should also write a program that uses the class `LinkedList` to process a sequence of orders consisting of an integer number indicating the queue the operation should be applied to, a string of the form `enqueue`, `dequeue`, `first`, `last`, `concatenate`, `len`, `print` and `empty`, and an integer number that represents the element to be enqueued if the string was “enqueue”, or the queue that should be concatenated with the queue identified by the number read before the order if the order was “concatenate”. The input begins with an integer number $n > 0$ indicating the number of queues to be handled, followed by a sequence of orders. The program should perform each order requested, if it can be executed.

Observation: Although the built-in **print** method is not defined for **queue**, you may override the *special method* `__str__` so that the contents of a **LinkedList** instance 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 **LinkedList** class:

```
def concatenate(self, q):
    # 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

```
2
1 enqueue 5
2 enqueue 3
1 enqueue 7
2 first
1 print
2 print
1 dequeue
2 len
1 dequeue
1 print
2 dequeue
2 is_empty
```

<pre>2 enqueue 5 2 enqueue 6 2 print 1 enqueue 2 1 enqueue 3 1 enqueue 4 1 print 1 is_empty 2 first 1 concatenate 2</pre>

Sample output 1

```
queue 1: 5 enqueued
queue 2: 3 enqueued
queue 1: 7 enqueued
queue 2 first element: 3
queue 1: 5 7
queue 2: 3
queue 1: 5 dequeued
queue 2 has 1 element(s)
queue 1: 7 dequeued
queue 1:
queue 2: 3 dequeued
```

```
queue 2 is empty
queue 2: 5 enqueued
queue 2: 6 enqueued
queue 2: 5 6
queue 1: 2 enqueued
queue 1: 3 enqueued
queue 1: 4 enqueued
queue 1: 2 3 4
queue 1 is not empty
queue 2 first element: 5
queues 1 and 2 concatenated
queue 1: 2 3 4 5 6
queue 2:
```

Problem information

Author: Josefina Sierra Santibáñez

Generation: 2026-01-25T17:13:12.223Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>