CS256 Advanced Programming – Assign7

Place all the files in **folder XXX_Assign7**, **submit** a compressed tar file **XXX_Assign7.tar.gz**.

1) **Linked_queue.py**
2) **positional_list.py**
3) **positional_list_2.py**

R-7.7 (35 points) Our CircularQueue class of Section 7.2.2 provides a rotate( ) method that has semantics equivalent to Q.enqueue(Q.dequeue( )), for a nonempty queue. Implement such a **method** rotate() for the LinkedQueue class of Section 7.1.2 without the creation of any new nodes.
Hint Adjust links so that the first node is moved to the end of the list. You could download the implementation of the class LinkedQueue from Moodle (linked_queue.py).
Requirements:

(1) Your method should **raise** an Empty exception if the queue is empty. (refer to dequeu() method in class LinkedQueue)

(2) Test code for using your function(s)should be placed in if __name__ == '__main__':

(3) Your test code should include the following:

- start from an empty queue
- call the rotate() method
- each time if the function you call may raise an exception, your should catch that exception and print out corresponding error message
- add several elements to the queue
- print out all the elements in the queue using rotate() and first() method

R-7.11 (35 points) Implement a **function**, with calling syntax max(L), that returns the maximum element from a PositionalList instance L containing comparable elements.
Hint Keep track of the maximum thus far while walking the list.
Requirements:

- Function max(L)accepts an instance of PositionalList L, and return the maximum element of L. If the instance is an empty list, you should raise an Empty exception.
- Provide test code for call the max() function and print out this max element. Here, you should also catch all possible exceptions, and print out corresponding error message.

R-7.13 (30 points) Update the PositionalList class to support an additional method find(e), which returns the position of the (first occurrence of) element *e* in the list
(or None if not found).
Hint Start looking at the beginning of the list.
Requirements:

- Function find() implemented as a member function of the PositionalList class. It accepts an element e, and return the position of the first occurrence of e in the list. If e is not found in the list, you should return None.
- Provide test code for call the find() function and print out this position if found and “not found” if the return value is None.