CMPSC-132: Programming and Computation II
Spring 2020

Lab #5

Due Date: 03/22/2020, 11:59PM

Read the instructions carefully before starting the assignment. Make sure your code follows the stated guidelines to ensure full credit for your work.

Instructions:

- The work in this lab must be completed alone and must be your own.
- **Download the starter code file from the LAB5 Assignment on Canvas. Do not change the function names on your script.**
- A doctest is provided as an example of code functionality. Getting the same result as the doctest does not guarantee full credit. You are responsible for debugging and testing your code with enough data, you can share ideas and testing code during your recitation class.
- Each function must return the output (Do not use print in your final submission, otherwise your submissions will receive a -1 point deduction)
- **Do not include test code outside any function in the upload. Printing unwanted or ill-formatted data to output will cause the test cases to fail. Remove all your testing code before uploading your file (You can also remove the doctest). Do not include the input() function in your submission.**
- **As a REMINDER, you are not allowed to use the internet to search for a solution or to use code written by other students that have taken (or are currently taking) the class**

Goal:

[10 pts] In our video lecture, we discussed the abstract data type Queue. A queue is a collection of items where the addition of new items happens at one end (tail) and the removal of existing items occurs at the other end (head) (FIFO). Use the Node class (an object with a data field and a pointer to the next element) to implement the queue data structure with the following operations:

- *enqueue*(item) adds a new Node with value=item to the tail of the queue. It needs the value of the Node and returns nothing.
- *dequeue*() removes the head Node from the queue. It needs no parameters and returns the value of the Node removed from the queue. The queue is modified.
- *isEmpty*() tests to see whether the queue is empty. It needs no parameters and returns a boolean value.
- *len*(queue_object) returns the number of items in the queue. It needs no parameters and returns an integer.

- You are not allowed to use any other data structures to copy the elements of the Queue for manipulating purposes in the enqueue and dequeue methods.
- You are not allowed to modify the given constructor
- You can't use queue functions from the Python library, otherwise no credit will be given

Tips:

- Starter code contains the special methods __str__ and __repr__, use them to ensure the queue operations are updating the elements in the queue correctly
- When calling dequeue, you always need to check the size of the queue first to ensure there are elements in the queue.
- Remember that calling dequeue when the size is 1 should update the head and tail pointers to None

NOTE: To grade this assignment, the grading script will perform a series of mixed queue operations and compare the final status of your queue. Verify that all your methods work correctly when mixed together.

EXAMPLE

```
>>> x=Queue()
>>> x.isEmpty()
True
>>> x.dequeue()
>>> x.enqueue(1)
>>> x.enqueue(2)
>>> x.enqueue(3)
>>> x.enqueue(4)
>>> print(x)
Head:Node(1)
Tail:Node(4)
Queue:1 2 3 4
>>> x.isEmpty()
False
>>> len(x)
4
>>> x.dequeue()
1
>>> x.dequeue()
2
>>> x.dequeue()
3
>>> x.dequeue()
4
>>> x.dequeue()
>>> print(x)
Head:None
Tail:None
Queue:
>>> x.enqueue(3)
>>> x.enqueue(2)
>>> print(x)
Head:Node(3)
Tail:Node(2)
Queue:3 2
>>> x.dequeue()
3
>>> print(x)
Head:Node(2)
Tail:Node(2)
Queue:2
```

```
>>> x.dequeue()
2
>>> print(x)
Head:None
Tail:None
Queue:
```

Deliverables:

- Submit your code in a file name LAB5.py to the Lab5 GradeScope assignment before the due date