CMPSC-122: Intermediate Programming
Spring 2018

Lab #11

Due Date: 04/06/2018, 11:59PM

Instructions:

- The work in this lab must be completed alone.
- If you need guidance, attend to your recitation class.
- Read the “Submitting assignments to Vocareum” file for instructions on how to submit this lab
- Do not change the function names or given code on your script
- The file name must be LAB11.py (incorrect name files will get a 0 score)
- You are responsible for testing your code. Use `python -i LAB11.py` in your terminal (or command prompt) to provide input to your functions. Test with as many data as you feel comfortable
- Each function must return the output (Do not use print in your final submission)
- Do not include test code outside any function in the upload. Remove all your testing code before uploading your file. If you are using input() to insert values in your functions and print to see the values, remove them.

Exercise 1 [10 pts]
NOTE: There is no partial credit for methods that do not work properly. You must ensure the entire code works after the addition of new methods.

In class, 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). Implement the queue data structure with the following operations:

- *Queue*() creates a new queue that is empty. It needs no parameters and returns nothing
- *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. [3 pts]
- *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. [3 pts]
- *isEmpty*() tests to see whether the queue is empty. It needs no parameters and returns a boolean value. [2 pts]
- *size*() returns the number of items in the queue. It needs no parameters and returns an integer. [2 pts]

EXAMPLE
```
>>> queue=Queue()
>>> queue.isEmpty()
True
>>> queue.enqueue(1)
>>> queue.enqueue(2)
>>> queue.enqueue(3)
>>> queue.enqueue(4)
>>> queue.printQueue()
1 2 3 4
>>> queue.isEmpty()
False
>>> queue.size()
4
>>> queue.dequeue()
1
>>> queue.dequeue()
2
>>> queue.dequeue()
3
>>> queue.dequeue()
4
>>> queue.dequeue()
'Queue is empty'
>>> queue.enqueue(3)
>>> queue.enqueue(2)
>>> queue.printQueue()
3 2
>>> queue.dequeue()
3
>>> queue.printQueue()
2 >>>
```

Tips:

- 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