CMPSC-132: Programming and Computation II
Fall 2018

Lab #9

Due Date: 10/19/2018, 11:59PM

Instructions:

- The work in this lab must be completed alone and must be your own. Do not copy code from online sources. That is considered plagiarism.
- Use the starter code provided on this CANVAS assignment. Do not change the function names or given started code on your script
- The file name must be LAB9.py (incorrect name files will get a -1 point deduction)
- **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 testing your code with enough data as possible.**
- Each function must return the output (Do not use print in your final submission otherwise, you will get a -1 pt deduction)
- Do not include test code outside any function in the upload. Remove all your testing code before uploading your file. Do not include the input() function in your submission.

Goal:

[10 pts] In class, we discussed the abstract data structure Stack. A stack is a collection of items where items are added to and removed from the top (LIFO). Use the Node class (an object with a data field and a pointer to the next element) to implement the stack data structure with the following operations:

- *Stack*() creates a new stack that is empty. It needs no parameters and returns nothing
- push(item) adds a new Node with value=item to the top of the stack. It needs the item and returns nothing.
- pop() removes the top Node from the stack. It needs no parameters and returns the **value** of the Node removed from the stack. Modifies the stack.
- peek() returns the **value** of the top Node from the stack but does not remove it. It needs no parameters. The stack is not modified.
- isEmpty() tests to see whether the stack is empty. It needs no parameters and returns a boolean value.
- len() returns the number of items on the stack. It needs no parameters and returns an integer. (You can add count in the Stack's constructor)

EXAMPLE

```
>>> x=Stack()
>>> x.pop()
'Stack is empty'
>>> x.push(2)
>>> x.push(4)
>>> x.push(6)
>>> x
Top:Node(6)
Stack:
6
4
2
```

```
>>> x.pop()
6
>>> x
Top:Node(4)
Stack:
4
2
>>> len(x)
2
>>> x.isEmpty()
False
>>> x.push(15)
>>> x
Top:Node(15)
Stack:
15
4
2
>>> x.peek()
15
>>> x
Top:Node(15)
Stack:
15
4
2
```

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

Tips:

- Make sure you update the top pointer according to the operation performed
- Starter code contains the special methods __str__ and __repr__, use them to ensure the stack operations are updating the elements in the stack correctly
- When a method is asking to return the value of a node, make sure you are returning node.value and not a Node object

Deliverables:

- Submit your code in a file name LAB9.py to the Lab9 CANVAS assignment before the due date