Assignment No.4

You must create three classes

MyCollection

MyQueue

MyStack

**MyCollection class**:

Your root class will be MyCollection and it will consist of two variables name\_of\_collection = None

and

container = None

The variable container must be initialized with empty list. This list will be latter used as either queue (in MyQueue class) or stack (in MyStack class).

This class will also implement/override \_\_lt\_\_ and \_\_eq\_\_ methods. The method \_\_eq\_\_ will check the stack/queue are equal or not, it will check both stack/queue contents and return True or False accordingly. For this purpose, it will compare the container (the list) and if both have same contents then it will return True, otherwise False.

The method \_\_lt\_\_ will check the size of the two stack/queues (which are compared) and will decide according to their size. For example, if two queues q1 and q2 (or stacks) have size of 3 and 4 respectively then q2 is bigger than q1 and vice versa.

These two methods will be implemented in the MyCollection class and will be inherited by both classes of MyStack and MyQueue and will not override in MyStack and MyQueue classes.

**MyStack class**:

This class will implement the following methods (just like in the stack data structure)

push() # **push the element in the stack**

pop(element) # **it will also return that popped element**

is\_empty() # **check if that stack is empty**

\_\_str\_\_ and \_\_repr\_\_ methods will be override to display the stack content as follows,

if \_\_name\_\_ == "\_\_main\_\_":

st0 = MyStack()  
  
st0.push(1)  
st0.push(2)  
st0.push(3)  
st0.push(4)  
st0.push(5)  
print(st0)  
st0.pop()  
st0.pop()  
st0.pop()  
st0.pop()  
st0.pop()  
st0.push('a')  
print(st0)  
st1 = MyStack()  
st1.push(1)  
st1.push(2)  
st1.push(3)  
print(st1)  
st2 = MyStack()  
st2.push(1)  
st2.push(2)  
st2.push(3)  
print(st2)  
if st1==st2:  
 print('Both stacks are equal')  
else:  
 print('Stacks are not equal')  
  
st2.push(4)  
if st1==st2:  
 print('Both stacks are equal')  
else:  
 print('Stacks are not equal')

The output should be as follows,

==================

5 -----> stack top

-----

4

-----

3

-----

2

-----

1

==================

==================

a -----> stack top

==================

3 -----> stack top

-----

2

-----

1

==================

==================

3 -----> stack top

-----

2

-----

1

==================

Both stacks are equal

Stacks are not equal

**MyQueue class**:

You will implement the following methods in this class,

enqueue(element) # **it will insert the element in the queue**

dequeue() # **it will return that dequeued element**

is\_empty() # it will check is the queue empty

\_\_str\_\_ and \_\_repr\_\_, methods will be override to display the stack content as follows,

If you run the following code (using your implemented classes),

if \_\_name\_\_ == "\_\_main\_\_":  
 q1 = MyQueue()  
 q2 = MyQueue()  
  
 print(q1)  
 print(q2)  
 q1.enqueue(1)  
 q1.enqueue(2)  
 q1.enqueue(3)  
  
 q2.enqueue(1)  
 q2.enqueue(2)  
 q2.enqueue(3)  
 q2.enqueue(4)  
  
 if q1 == q2:  
 print('Both queues are equal')  
 else:  
 print('Queues are not equal')  
  
 if q1 < q2:  
 print('Queues q1 is smaller than q2')  
 else:  
 print('Queues q1 is bigger than q2')  
  
 print(q1)  
 q1.dequeue()  
 q1.dequeue()  
 print(q1)  
 q1 = MyQueue()  
 print(q1)

Then its output should looks like follows,

Queue is empty

Queue is empty

Queues are not equal

Queues q1 is smaller than q2

1 --- Head of Queue

2

3 --- Tail of Queue

1 --- Head of Queue

2

3 --- Tail of Queue

3 --- Head/Tail of Queue

List is empty

Note: You must submit the source code (.py file), in any other format will

not be accepted and zero point will be awarded.

In case of copying or plagiarism, zero will be given to both or all the

candidates who are involve in plagiarism. Also, after dead line no

assignment will be accepted. If you have any question in mind you can

discuss it in class or email or in office