Queue Using Stack

**Build a Queue From Stacks**

In this exercise we are going to create a queue with just stacks.

**Code**

In [2]:



**class** Stack:

**def** \_\_init\_\_(self):

self.items **=** []

**def** size(self):

**return** len(self.items)

**def** push(self, item):

self.items.append(item)

​

**def** pop(self):

**if** self.size()**==**0:

**return** **None**

**else**:

**return** self.items.pop()

​

**class** Queue:

**def** \_\_init\_\_(self):

self.instorage**=**Stack()

self.outstorage**=**Stack()

**def** size(self):

**return** self.outstorage.size() **+** self.instorage.size()

**def** enqueue(self,item):

self.instorage.push(item)

**def** dequeue(self):

**if** **not** self.outstorage.items:

**while** self.instorage.items:

self.outstorage.push(self.instorage.pop())

**return** self.outstorage.pop()

**Test Cases**

In [3]:



*# Setup*

q **=** Queue()

q.enqueue(1)

q.enqueue(2)

q.enqueue(3)

​

*# Test size*

print ("Pass" **if** (q.size() **==** 3) **else** "Fail")

​

*# Test dequeue*

print ("Pass" **if** (q.dequeue() **==** 1) **else** "Fail")

​

*# Test enqueue*

q.enqueue(4)

print ("Pass" **if** (q.dequeue() **==** 2) **else** "Fail")

print ("Pass" **if** (q.dequeue() **==** 3) **else** "Fail")

print ("Pass" **if** (q.dequeue() **==** 4) **else** "Fail")

q.enqueue(5)

print ("Pass" **if** (q.size() **==** 1) **else** "Fail")

Pass

Pass

Pass

Pass

Pass

Pass