

Problem Set 6

Justin Ely

615.202.81.FA15 Data Structures

13 October, 2015

1)

```
# Nodes are initialized to Null left and right pointers
class Node(value)
    value = value
    left = Null
    right = Null

class Deque:
    RightNode = Node(Null)
    LeftNode = Node(Null)

    def isEmpty()
        if RightNode.left == Null and LeftNode.right == Null:
            return True
        else
            return False

    def InsertLeft(value)
        NewNode = Node(value)

        if isEmpty()
            RightNode = NewNode
            LeftNode = NewNode
        else
            NewNode.right = LeftNode
            NewNode.left = LeftNode.next
            LeftNode.left = NewNode

    def DeleteRight()
        if isEmpty()
            raise Exception
```

```

    else
        tmp = RightNode
        RightNode.left.right = RightNode.right
        RightNode = RightNode.left

    return tmp.value

```

2)

```

# Nodes are initialized to Null left and right pointers
class Node(value)
    value = value
    left = Null
    right = Null

class Deque:
    HeadNode = Node(Null)
    HeadNode.right = HeadNode
    HeadNode.left = HeadNode

    RightNode = Header.right
    LeftNode = Header.left

    def isEmpty()
        if RightNode.left == LeftNode.right:
            return True
        else
            return False

    def InsertRight(value)
        NewNode = Node(value)

        RightNode.right = NewNode
        NewNode.left = RightNode

        LeftNode.left = NewNode
        NewNode.right = LeftNode

    def DeleteLeft()
        if isEmpty()
            raise Exception

```

```

else
    tmp = LeftNode
    LeftNode.right.left = LeftNode.left
    LeftNode = LeftNode.right

    return tmp.value

```

3)

```

class SharedArray(N_NODES)
    data = Array(N_NODES)

    FreeIndx = Stack()

    for (i=0; i<N_NODES; i++)
        FreeIndex.push(i)

    def Borrow():
        if FreeIndx.isEmpty():
            raise Excetion

        return FreeIndx.pop()

    def Return(index):
        FreeIndx.push(index)

//-----

class SharedStack()
    data = SharedArray
    StartNode = -1

    def isEmpty()
        if StartNode == -1
            return True
        else
            return False

    def Push(value)
        newIndex = data.Borrow()

        if startNode == -1:

```

```

        startNode = newIndex

    data[newIndex].value = value
    data[newIndex].next = -1

def Pop()
    if isEmpty()
        raise Exception

    node = StartNode
    prev = Null
    while not node.next == -1:
        prev = node
        node = data[node.next]

    data[prev.index].next = -1
    data.Return(node.index)

    return node.value
//-----

class SharedQueue(sharedArrayInstance)
    def Insert

    def Delete

```