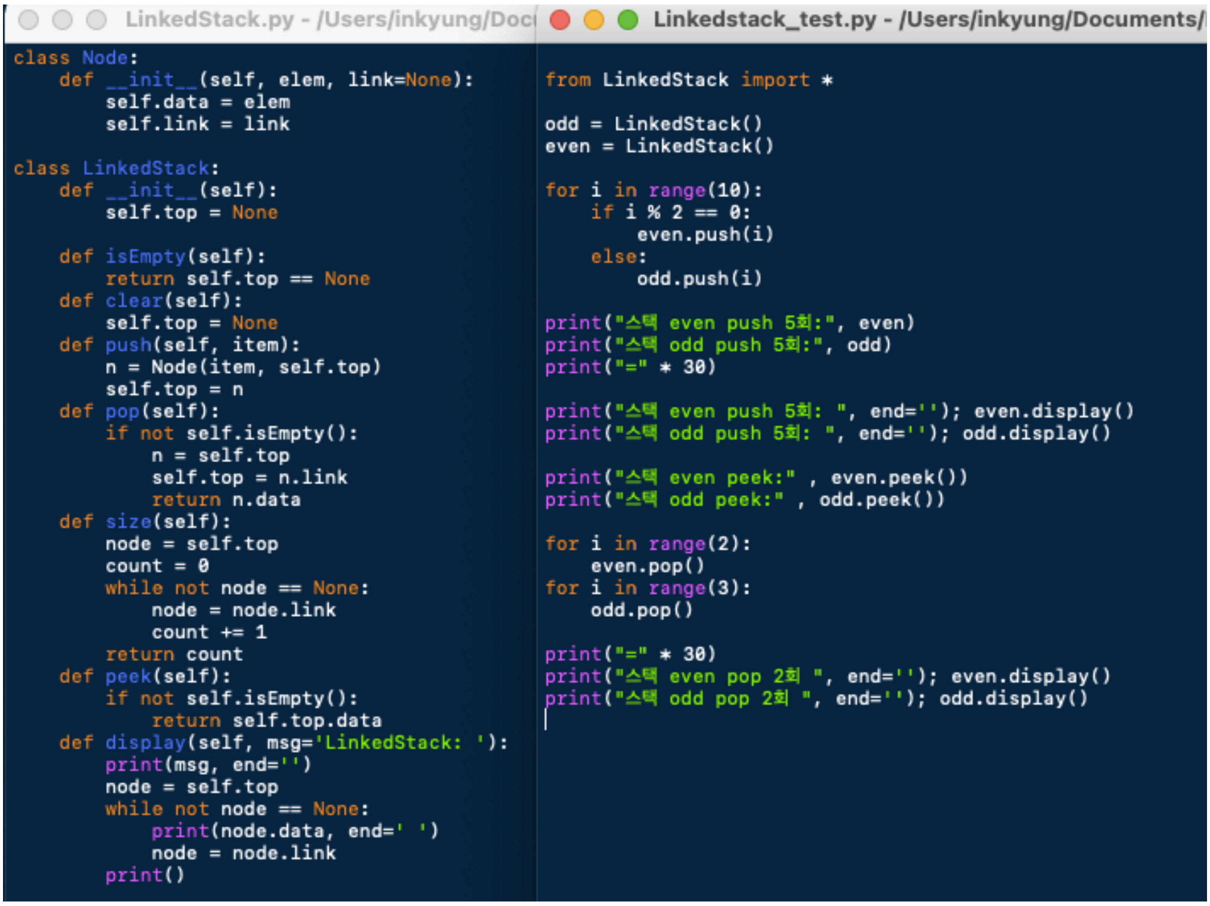
자료구조 6장 과제

Global Business & Technology
201904385 우인경

1. 단순 연결리스트를 이용한 스택 구현 및 실습

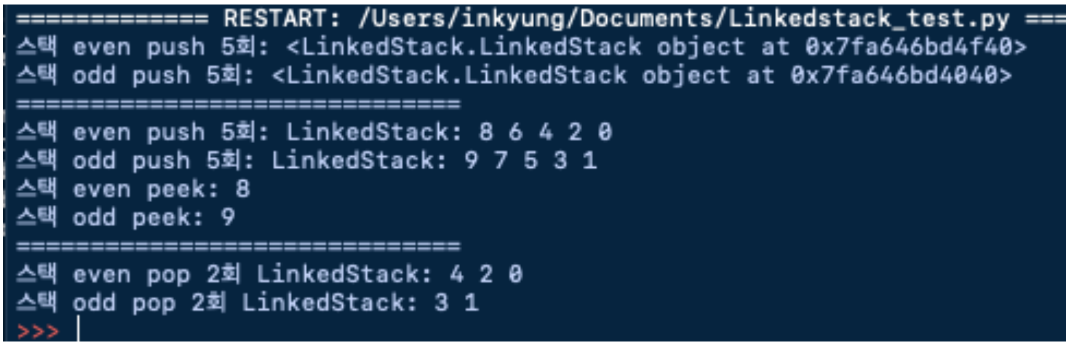
- 코드화면


LinkedStack.py - /Users/inkyung/Doc
Linkedstack_test.py - /Users/inkyung/Documents/

```python
class Node:
    def __init__(self, elem, link=None):
        self.data = elem
        self.link = link

class LinkedStack:
    def __init__(self):
        self.top = None

    def isEmpty(self):
        return self.top == None
    def clear(self):
        self.top = None
    def push(self, item):
        n = Node(item, self.top)
        self.top = n
    def pop(self):
        if not self.isEmpty():
            n = self.top
            self.top = n.link
            return n.data
    def size(self):
        node = self.top
        count = 0
        while not node == None:
            node = node.link
            count += 1
        return count
    def peek(self):
        if not self.isEmpty():
            return self.top.data
    def display(self, msg='LinkedStack: '):
        print(msg, end='')
        node = self.top
        while not node == None:
            print(node.data, end=' ')
            node = node.link
        print()
```

```python
from LinkedStack import *

odd = LinkedStack()
even = LinkedStack()

for i in range(10):
    if i % 2 == 0:
        even.push(i)
    else:
        odd.push(i)

print("스택 even push 5회:", even)
print("스택 odd push 5회:", odd)
print("=" * 30)

print("스택 even push 5회: ", end=''); even.display()
print("스택 odd push 5회: ", end=''); odd.display()

print("스택 even peek:" , even.peek())
print("스택 odd peek:" , odd.peek())

for i in range(2):
    even.pop()
for i in range(3):
    odd.pop()

print("=" * 30)
print("스택 even pop 2회 ", end=''); even.display()
print("스택 odd pop 2회 ", end=''); odd.display()
```

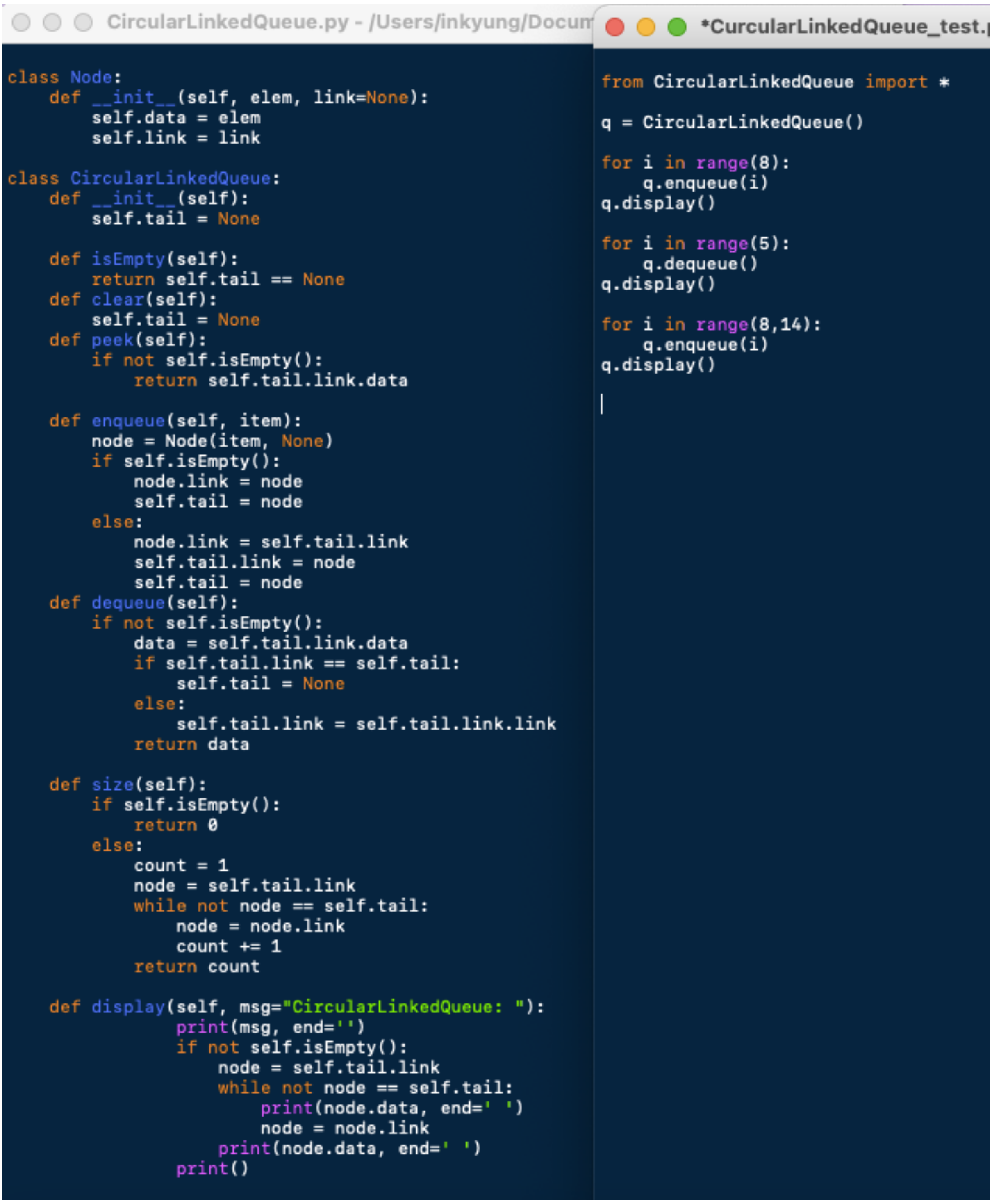
- 실행화면



```
============= RESTART: /Users/inkyung/Documents/Linkedstack_test.py ===
스택 even push 5회: <LinkedStack.LinkedStack object at 0x7fa646bd4f40>
스택 odd push 5회: <LinkedStack.LinkedStack object at 0x7fa646bd4040>
==============================
스택 even push 5회: LinkedStack: 8 6 4 2 0
스택 odd push 5회: LinkedStack: 9 7 5 3 1
스택 even peek: 8
스택 odd peek: 9
==============================
스택 even pop 2회 LinkedStack: 4 2 0
스택 odd pop 2회 LinkedStack: 3 1
>>> 
```

2. 원형 연결리스트를 이용한 큐 구현 및 실습

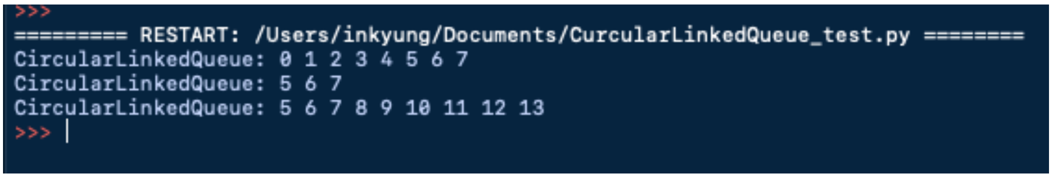
- 코드화면


CircularLinkedQueue.py - /Users/inkyung/Docum
*CurcularLinkedQueue_test.

```python
class Node:
    def __init__(self, elem, link=None):
        self.data = elem
        self.link = link

class CircularLinkedQueue:
    def __init__(self):
        self.tail = None

    def isEmpty(self):
        return self.tail == None
    def clear(self):
        self.tail = None
    def peek(self):
        if not self.isEmpty():
            return self.tail.link.data

    def enqueue(self, item):
        node = Node(item, None)
        if self.isEmpty():
            node.link = node
            self.tail = node
        else:
            node.link = self.tail.link
            self.tail.link = node
            self.tail = node
    def dequeue(self):
        if not self.isEmpty():
            data = self.tail.link.data
            if self.tail.link == self.tail:
                self.tail = None
            else:
                self.tail.link = self.tail.link.link
            return data

    def size(self):
        if self.isEmpty():
            return 0
        else:
            count = 1
            node = self.tail.link
            while not node == self.tail:
                node = node.link
                count += 1
            return count

    def display(self, msg="CircularLinkedQueue: "):
                print(msg, end='')
                if not self.isEmpty():
                    node = self.tail.link
                    while not node == self.tail:
                        print(node.data, end=' ')
                        node = node.link
                    print(node.data, end=' ')
                print()
```

```python
from CircularLinkedQueue import *

q = CircularLinkedQueue()

for i in range(8):
    q.enqueue(i)
q.display()

for i in range(5):
    q.dequeue()
q.display()

for i in range(8,14):
    q.enqueue(i)
q.display()
```

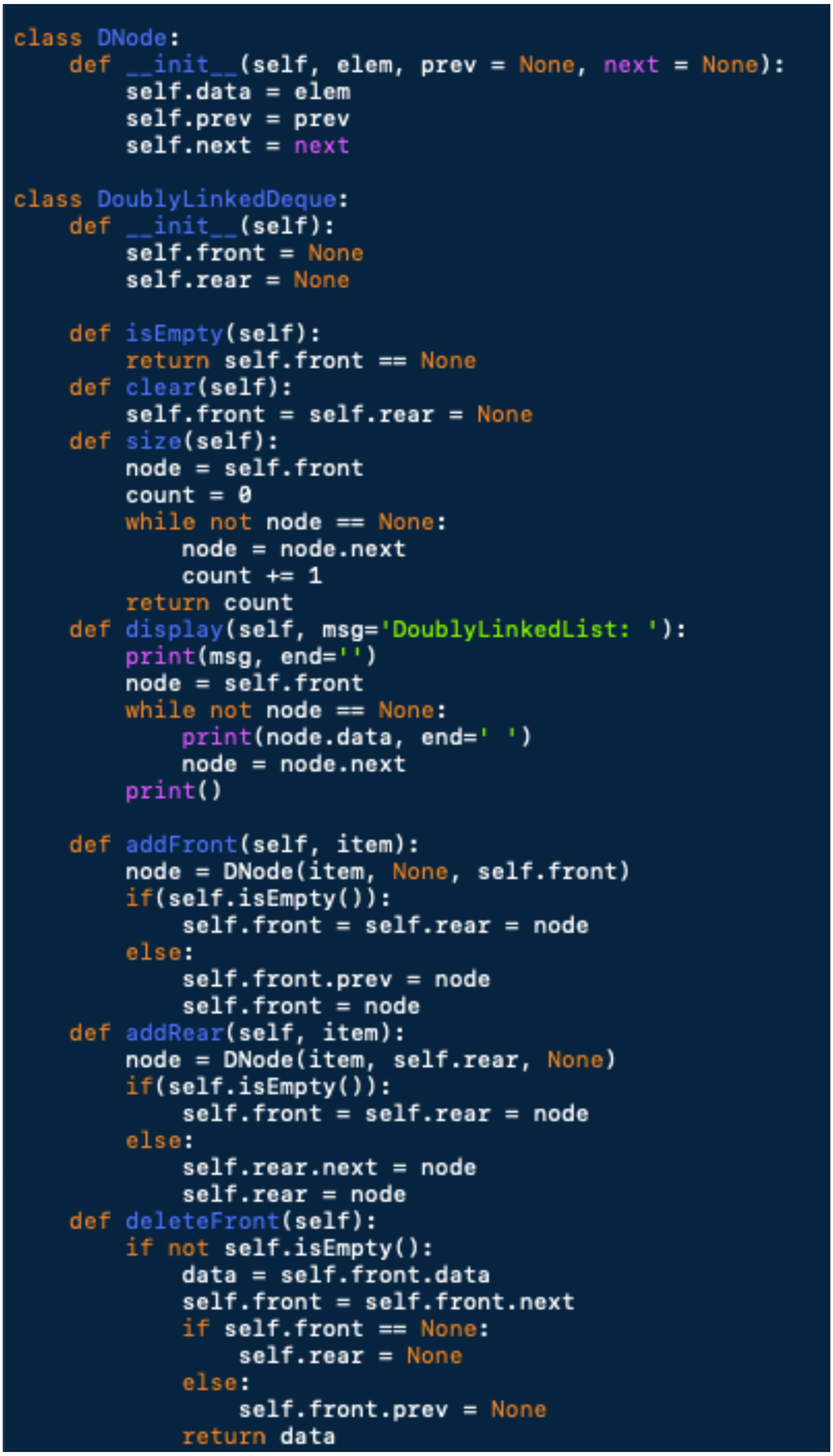
- 실행화면



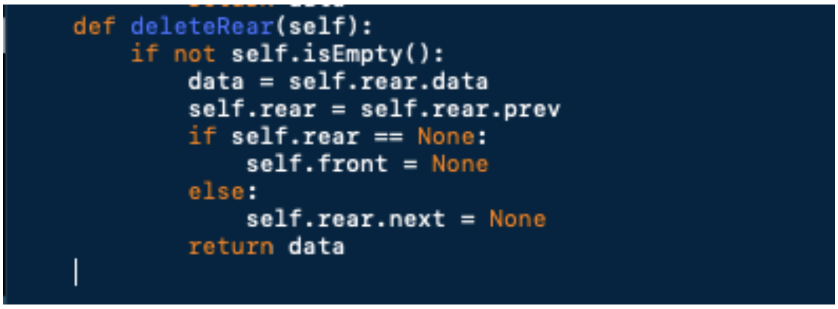
```
>>>
========= RESTART: /Users/inkyung/Documents/CurcularLinkedQueue_test.py ========
CircularLinkedQueue: 0 1 2 3 4 5 6 7
CircularLinkedQueue: 5 6 7
CircularLinkedQueue: 5 6 7 8 9 10 11 12 13
>>>
```

3. 이중 연결리스트를 이용한 덱 구현 및 실습

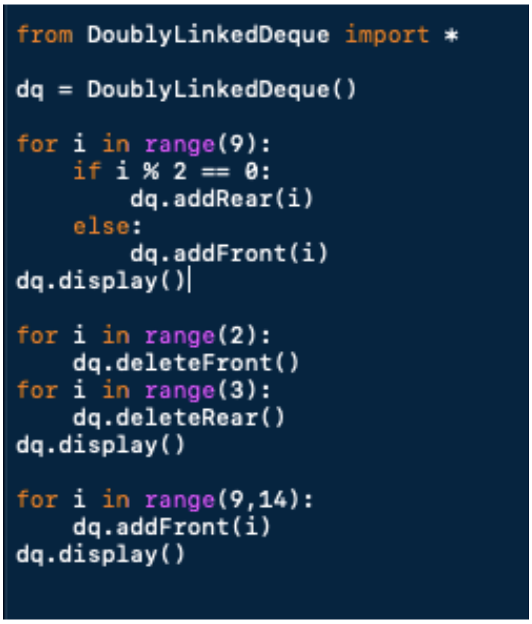
- 코드화면



```python
class DNode:
    def __init__(self, elem, prev = None, next = None):
        self.data = elem
        self.prev = prev
        self.next = next

class DoublyLinkedDeque:
    def __init__(self):
        self.front = None
        self.rear = None

    def isEmpty(self):
        return self.front == None
    def clear(self):
        self.front = self.rear = None
    def size(self):
        node = self.front
        count = 0
        while not node == None:
            node = node.next
            count += 1
        return count
    def display(self, msg='DoublyLinkedList: '):
        print(msg, end='')
        node = self.front
        while not node == None:
            print(node.data, end=' ')
            node = node.next
        print()

    def addFront(self, item):
        node = DNode(item, None, self.front)
        if(self.isEmpty()):
            self.front = self.rear = node
        else:
            self.front.prev = node
            self.front = node
    def addRear(self, item):
        node = DNode(item, self.rear, None)
        if(self.isEmpty()):
            self.front = self.rear = node
        else:
            self.rear.next = node
            self.rear = node
    def deleteFront(self):
        if not self.isEmpty():
            data = self.front.data
            self.front = self.front.next
            if self.front == None:
                self.rear = None
            else:
                self.front.prev = None
            return data
```

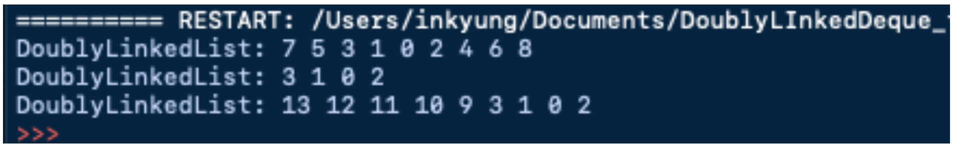


```
    def deleteRear(self):
        if not self.isEmpty():
            data = self.rear.data
            self.rear = self.rear.prev
            if self.rear == None:
                self.front = None
            else:
                self.rear.next = None
            return data
```



```
from DoublyLinkedDeque import *

dq = DoublyLinkedDeque()

for i in range(9):
    if i % 2 == 0:
        dq.addRear(i)
    else:
        dq.addFront(i)
dq.display()

for i in range(2):
    dq.deleteFront()
for i in range(3):
    dq.deleteRear()
dq.display()

for i in range(9,14):
    dq.addFront(i)
dq.display()
```

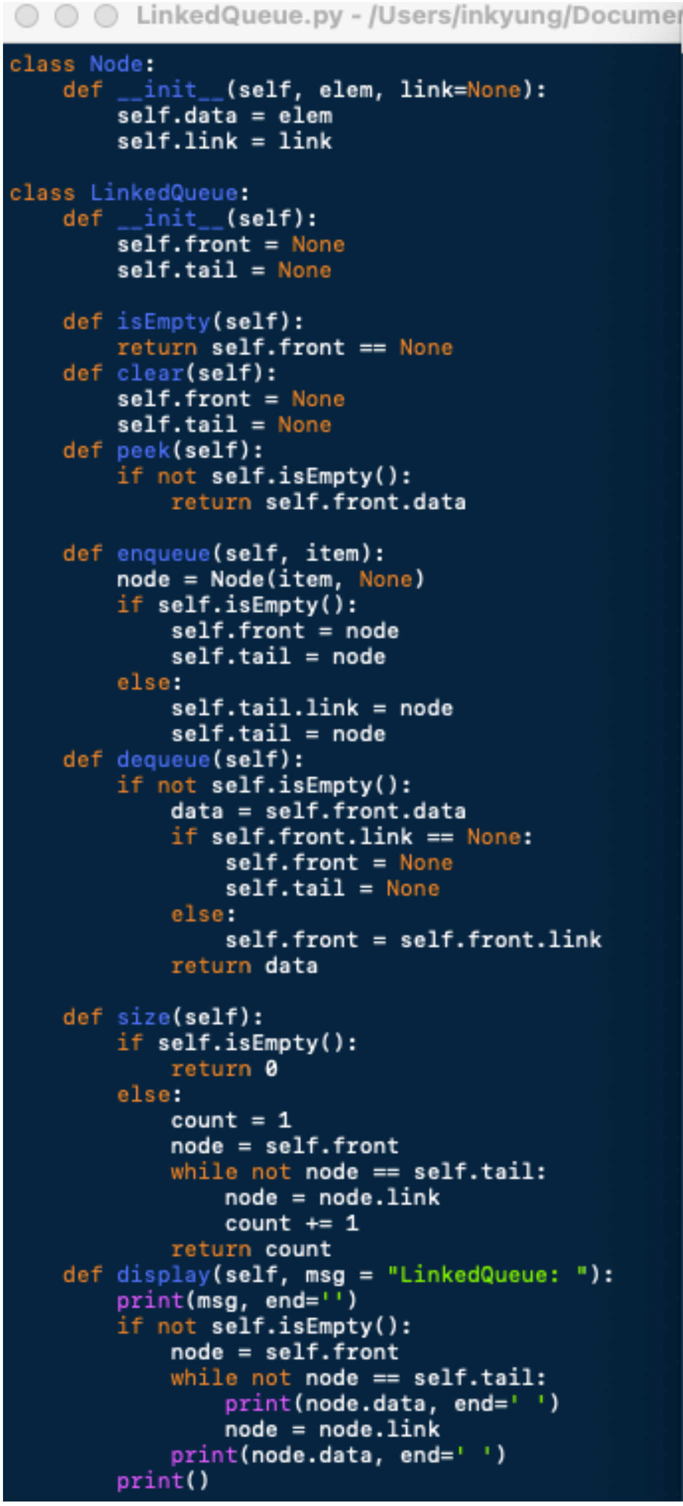
- 실행화면



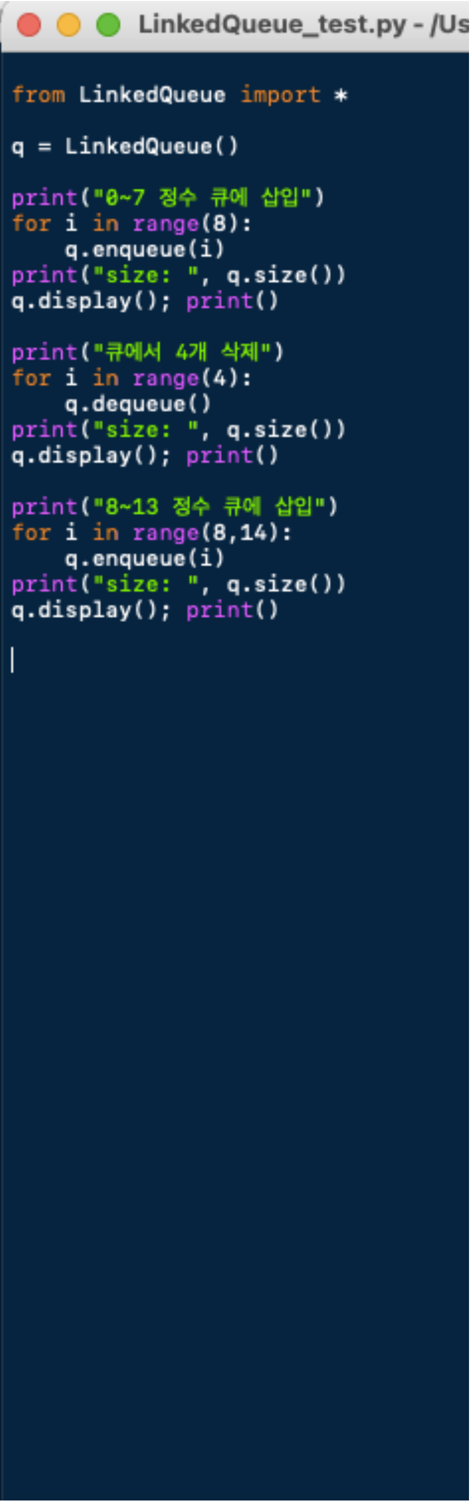
```
========== RESTART: /Users/inkyung/Documents/DoublyLInkedDeque_
DoublyLinkedList: 7 5 3 1 0 2 4 6 8
DoublyLinkedList: 3 1 0 2
DoublyLinkedList: 13 12 11 10 9 3 1 0 2
>>>
```

4. 실습문제 6.3번: 단순 연결리스트를 이용한 큐 구현

- 코드화면

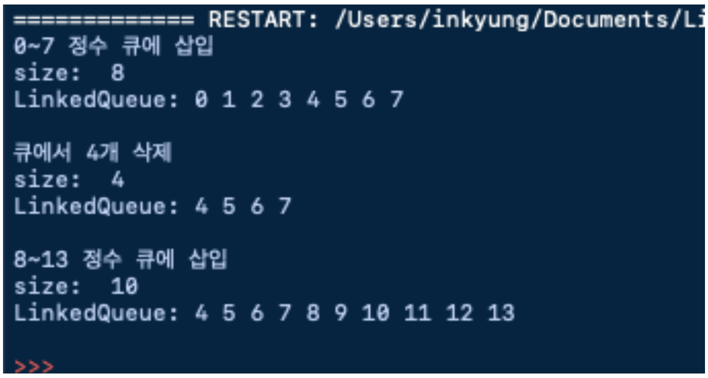


```python
class Node:
    def __init__(self, elem, link=None):
        self.data = elem
        self.link = link

class LinkedQueue:
    def __init__(self):
        self.front = None
        self.tail = None

    def isEmpty(self):
        return self.front == None
    def clear(self):
        self.front = None
        self.tail = None
    def peek(self):
        if not self.isEmpty():
            return self.front.data

    def enqueue(self, item):
        node = Node(item, None)
        if self.isEmpty():
            self.front = node
            self.tail = node
        else:
            self.tail.link = node
            self.tail = node
    def dequeue(self):
        if not self.isEmpty():
            data = self.front.data
            if self.front.link == None:
                self.front = None
                self.tail = None
            else:
                self.front = self.front.link
            return data

    def size(self):
        if self.isEmpty():
            return 0
        else:
            count = 1
            node = self.front
            while not node == self.tail:
                node = node.link
                count += 1
            return count
    def display(self, msg = "LinkedQueue: "):
        print(msg, end='')
        if not self.isEmpty():
            node = self.front
            while not node == self.tail:
                print(node.data, end=' ')
                node = node.link
            print(node.data, end=' ')
        print()
```



```python
from LinkedQueue import *

q = LinkedQueue()

print("0~7 정수 큐에 삽입")
for i in range(8):
    q.enqueue(i)
print("size: ", q.size())
q.display(); print()

print("큐에서 4개 삭제")
for i in range(4):
    q.dequeue()
print("size: ", q.size())
q.display(); print()

print("8~13 정수 큐에 삽입")
for i in range(8,14):
    q.enqueue(i)
print("size: ", q.size())
q.display(); print()
```

- 실행화면



```
============= RESTART: /Users/inkyung/Documents/Li
0~7 정수 큐에 삽입
size:  8
LinkedQueue: 0 1 2 3 4 5 6 7

큐에서 4개 삭제
size:  4
LinkedQueue: 4 5 6 7

8~13 정수 큐에 삽입
size:  10
LinkedQueue: 4 5 6 7 8 9 10 11 12 13

>>>
```