第3次隨堂-隨堂-QZ3

學號：112111219
姓名：陳恩偉
作業撰寫時間：20 (mins，包含程式撰寫時間)
最後撰寫文件日期：2024/11/25

本份文件包含以下主題：(至少需下面兩項，若是有多者可以自行新增)

- ☑ 說明內容
- ☑ 個人認為完成作業須具備觀念

說明程式與內容

1. 請參閱投影片Topic5的第31至35頁，請用物件導向方式進行新增與刪除。 (請參照題目pdf)

Ans:

```
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None

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

    def push(self, item):
        new_node = Node(item)
        new_node.next = self.top
        self.top = new_node
        print(f"Pushed: {item}")

    def pop(self):
        if self.top is None:
            print("Stack is empty")
            return None
        popped_node = self.top
        self.top = self.top.next
        print(f"Popped: {popped_node.data}")
        return popped_node.data

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

    def enqueue(self, item):
        new_node = Node(item)
        if self.rear is None:
            self.front = self.rear = new_node
```

```
            else:
                self.rear.next = new_node
                self.rear = new_node
            print(f"Enqueued: {item}")

        def dequeue(self):
            if self.front is None:
                print("Queue is empty")
                return None
            dequeued_node = self.front
            self.front = self.front.next
            if self.front is None:
                self.rear = None
            print(f"Dequeued: {dequeued_node.data}")
            return dequeued_node.data

    stack = LinkedStack()
    stack.push(1)
    stack.push(2)
    stack.pop()
    stack.pop()


    queue = LinkedQueue()
    queue.enqueue(1)
    queue.enqueue(2)
    queue.dequeue()
    queue.dequeue()
```

push/enqueue：在鏈結堆疊中以push操作新增節點；鏈結佇列enqueue操作新增節點到尾端。
pop/dequeue：在鏈結堆疊中以pop操作刪除頂端節點；鏈結佇列dequeue操作刪除前端節點。

個人認為完成作業須具備觀念

本次練習旨在掌握鏈結堆疊與鏈結佇列的基本操作與概念。需理解如何利用物件導向方法實現資料結構，包括節點的建立與指標的應用。動態記憶體配置是關鍵概念，需學會如何新增與刪除節點，並能有效管理資料的排列順序，還需熟悉後進先出與先進先出的原則，這些在處理資料時非常重要。