

แบบฝึกหัดชุดที่ 6

1. จงเติม Class ให้สมบูรณ์ พร้อมแสดงการเรียกใช้งาน

```
class Car:
    def __init__(self, license, brand, color):
        # c = Car('AA1234', 'Honda', 'White')
        # มีตัวแปร report สำหรับเก็บข้อมูลประวัติการซ่อมบำรุง โดยกำหนดค่าเริ่มต้นเป็นลิสต์ว่าง

    def __str__(self):
        # คืนสตริงของรถยนต์ เช่น 'AA1234 - White Honda'

    def __lt__(self, rhs):
        # เรียงลำดับรถยนต์โดยเปรียบเทียบป้ายทะเบียนรถแบบสตริง

    def add_report(self, new_report):
        # เพิ่มประวัติการซ่อมบำรุง โดยไม่ต้องคืนค่า
        # ตัวแปร new_report เก็บ tuple (วันที่, คำอธิบาย, ราคา)
        # เช่น c.add_report( ('25 May 2017', 'change tires', 1500) )

    def total_payment(self):
        # คืนค่าใช้จ่ายทั้งหมดที่ใช้ในการซ่อมบำรุงที่ผ่านมา

    def max_payment(self):
        # คืนลิสต์ของประวัติการซ่อมบำรุง (วันที่, คำอธิบาย, ราคา) ทุกรายการ ที่มีค่าใช้จ่ายมากที่สุด
        # กรณีที่รถยนต์ไม่มีประวัติการซ่อมบำรุงเลย ให้คืนค่าลิสต์ว่าง
```

2. จงเติม Class ให้สมบูรณ์ พร้อมแสดงการเรียกใช้งาน

```
class ShoppingCart:
    def __init__(self, id):
        self.id = id
        self.books = []
        # books เก็บลิสต์ของหนังสือในตะกร้าพร้อมจำนวน เช่น [[b1,2],[b3,7]]

    def add_book(self, book, n):
        # เพิ่มข้อมูลการซื้อหนังสือ book เพิ่มอีก n เล่ม โดยไม่ต้องคืนค่า
        # หากไม่มีหนังสือเล่มนี้ในตะกร้า ให้เพิ่มลิสต์ [book, n] ต่อท้าย books
        # หากเคยมีข้อมูลหนังสือเล่มนี้ในตะกร้าแล้ว ให้เพิ่มจำนวนที่ซื้ออีก n เล่ม
        # เช่น ถ้า books = [[b1,2]] และเราสั่ง add_book(b1,3) จะได้ books = [[b1,5]]

    def delete_book(self, book):
        # ลบข้อมูลการซื้อหนังสือ book ออกจากตะกร้า โดยไม่ต้องคืนค่า
        # ถ้าในตะกร้าไม่มีหนังสือ book ไม่ต้องทำอะไร

    def get_total(self):
        # คืนค่าราคารวมของหนังสือทั้งหมดในตะกร้า

    def __lt__(self, rhs):
        # ตะกร้าที่มีราคารวมของหนังสือน้อยกว่า จะเป็นตะกร้าที่น้อยกว่า
```