Assignment #6 64015030

ชินพัฒน์ ลิ้มประธาน

1. จงเติม Class ให้สมบูรณ์ พร้อมแสดงการเรียกใช้งาน

```python
class Car:
    def __init__(self, license, brand, color):
        self.license = license
        self.brand = brand
        self.color = color
        self.report = []
        self.current_payment = 0

    def __str__(self):
        return f"{self.license} - {self.color} {self.brand}"

    def __lt__(self, rhs):
        return self.license < rhs.license

    def add_report(self, new_report):
        self.report.append(new_report)
        self.current_payment += new_report[2]

    def total_payment(self):
        return self.current_payment

    def max_payment(self):
        self.report.sort(key = lambda report: report[2])
        return self.report[len(self.report)-1] if self.report else []

car_1 = Car('AA1234', 'Honda', 'White')
car_2 = Car('AA1235', 'Toyota', 'Black')

car_1.add_report(('25 May 2017', 'change tires', 1500))
car_1.add_report(('26 May 2017', 'change tires', 3000))
car_1.add_report(('30 May 2017', 'change choke', 2500))
car_1.add_report(('1 June 2017', 'change front hood', 5500))

print(car_1)
print(f"report {car_1.report}")
print(f"max payment {car_1.max_payment()}")
print(f"total payment {car_1.total_payment()}")

print(car_2 > car_1)
```

```
AA1234 - White Honda
report [('25 May 2017', 'change tires', 1500), ('26 May 2017', 'chang
e tires', 3000), ('30 May 2017', 'change choke', 2500), ('1 June 2017
', 'change front hood', 5500)]
max payment ('1 June 2017', 'change front hood', 5500)
total payment 12500
True
```

2. จงเติม Class ให้สมบูรณ์ พร้อมแสดงการเรียกใช้งาน

```python
class ShoppingCart:
    def __init__(self, id):
        self.id = id
        self.books = []
        self.total_price = 0

    def add_book(self, book, n, price = 0):
        book_list = self.books
        no_duplicate_books = True
        for current_book in book_list:
            if current_book[0] == book:
                self.total_price += n * current_book[2]
                current_book[1] += n
                no_duplicate_books = False
        if no_duplicate_books:
            self.books.append([book, n, price])
            self.total_price += n * price

    def delete_book(self, book):
        book_list = self.books
        for current_book in book_list:
            if current_book[0] == book:
                self.total_price -= current_book[1] * current_book[2]
                book_list.remove(current_book)

    def get_total(self):
        return self.total_price

    def __lt__(self, rhs):
        return self.total_price < rhs.total_price
```

```python
cart_1 = ShoppingCart(1)
cart_1.add_book('b1', 2, 100)
cart_1.add_book('b1', 3)
cart_1.add_book('b2', 1, 103.5)
print(f"books in cart 1: {cart_1.books}")

cart_1.delete_book('b2')
print(f"books in cart 1: {cart_1.books}")

cart_2 = ShoppingCart(2)
cart_2.add_book('b1', 1, 200)
cart_2.add_book('b2', 3, 250)

print(f"cart 1 total price: {cart_1.get_total()}")

print(f"books in cart 2: {cart_2.books}")
print(f"cart 2 total price: {cart_2.get_total()}")

print(cart_1 < cart_2)
```

```
books in cart 1: [['b1', 5, 100], ['b2', 1, 103.5]]
books in cart 1: [['b1', 5, 100]]
cart 1 total price: 500.0
books in cart 2: [['b1', 1, 200], ['b2', 3, 250]]
cart 2 total price: 950
True
```