عبارت assert که در کد قرار گرفته شده برای بررسی کردن یک شرط استفاده می‌شه. اگه شرط برقرار نباشد، یعنی مثلا مقدار قیمت نهایی کمتر یا مساوی قیمت اصلی یا کمتر از صفر بشه میاد ، یه خطای AssertionError درست میکنه. اما تو کدی تو سوال دادید، عبارت assert به شکل غلط نوشته شده و شرط اون به صورت(0 < final_price <= price) نوشتید.

به خاطر نوشتن عملگر >= تو شرط کد، هرموقع final_price برابر با price میشه، شرط به این صورت برقرار می‌شه و خطای AssertionError ایجاد نمیشه برای درست شدن این مشکل، شرط رو به شکل (0 < final_price < price)تغییر میدیم تا موقعی که final_price برابر با price شد شرط برقرار نشه و خطای AssertionError ایجاد شود.

حالا کد درست این شکلی میشه :

```
def apply_discount(price: int, discount: float = 0.0) -> int:
    final_price = int(price * (1 - discount))
    assert 0 < final_price < price, "The final price is not within the expected range."
    return final_price
```

جواب قسمت دوم :

assert راه حل مناسبی هستش به خاطر اینکه

assertبررسی می‌کنه که شرط مورد نظر برقرار باشه. تو این کد، شرط رو بررسی می‌کنه که قیمت نهایی (final_price) باید بیشتر از صفر و کوچکتر از قیمت اصلی (price) باشه. اگه که این شرط برقرار نباشه، یعنی قیمت نهایی خارج از محدوده مورد انتظاره و خطای AssertionErrorایجاد میشه

دوباره در ترمینال با کامند O–خروجی گرفتم ، خروجی کد همون خروجی بود که داخل برنامه کامنت کرده بودم و این دستور تغییر خاصی برای بهینه کردن تو کد و خروجی نداد فکر میکنم به خاطر این بود که کد من به اندازه کافی بهینه بود و به خوبی کار میکرد.