برای تمام سوال‌ها یک فایل با نام تابع یا کلاس مورد نظر با پسوند js. بسازید و تابع یا کلاس خواسته شده را `export` کنید.

۱. در این سوال می‌خواهیم یک فروشگاه ساده را مدل‌سازی کنیم. برای این سوال دو فایل با نام‌های models.js و store.js را اینجاد کنید و ویژگی‌های زیر را در آن‌ها پیاده‌سازی کنید:

models.js باید شامل سه مدل Product,User و Comment باشد.

Product کلاسی با فیلدها و توابع زیر باشد.

- id (number): یک مقدار یکتا برای مشخص شدن کد کالا. برای تولید این id نیز از یک متد static در کلاس Product .بنویسید.
- name (string)
- price (number)
- Comment (Array of Comments)
- addComment()
- constructor(name, price, categories)

User کلاسی با فیلدهای

- userName
- purchaseHistory (Array of Product ids)

Comment کلاسی با فیلدهای

- user (a User Object)
- rating(Number)
- text (string)

در فایل store.js باید یک دیکشنری از کالاها با کلیدهای ProductIds و مقدار product و count داشته باشد {productId: { product: Object, count: Number}

همچنین یک آرایه از کاربران نیز نگه داری کند.

با استفاده از models.js توابع زیر را پیاده سازی کنید:

- addProduct(product, count)
- removeProduct(product, count)

- addUser(username)
- getTotalInventoryCount() تعداد کل کالاهای موجود
- getCommentsOfUser(user)
- getTotalProfit() درآمد فروشگاه را حساب کند
- getCommentsByUser(user) یک لیست از متن نظرات کاربر برمی‌گرداند
- getRating(product) میانگین رای کسانی که این کالا را خریداری کرده‌اند محاسبه کند.

برای این سوال جزییات پیاده‌سازی به شما واگذار شده و کیفیت کد شما نیز بررسی می‌شود. به عنوان مثال برای تابع removeProduct باید تصمیم بگیرید که در صورت وجود نداشتن چنین محصولی خطای استفاده کننده از تابع را چگونه مدیریت کنید.

۲. تابع `deepEqual(obj1, obj2)` برای مقایسه‌ی عمیق دو مقدار پیاده‌سازی کنید. برای جزئیات به مثال‌ها توجه کنید. برای این سوال مجاز به استفاده از کتابخانه‌های npm نیستید. مقادیر null و undefined و دیگر حالت‌های خاص در ورودی‌ها وجود ندارد.

مثال

```
deepEqual(true, true) //true
deepEqual(1, [1]) //false
deepEqual([1,[2,3]], [1,[2,3]]) //true
deepEqual([1,2], [2,1]) //false
deepEqual({a: 1, b:2}, {b: 2, a: 1}) //true
deepEqual({a: [1,2,3], b: {c: '4'}}, {a: [1,2,3], b: {c: '4'}) //true
deepEqual({a: [1,2,3], b: {c: 4}}, {a: [1,2,3], b: {c: 5}) //false
```