سؤال ۱. مرکز خرید (۳۵ نمره)

در این سوال باید یک مرکز خرید بزرگ را شبیه سازی کنید. مرکز خرید به این صورت است که شامل تعدادی مغازه میباشد. هر مغازه تعدادی جنس دارد. و از هر جنس چند نمونه محدود. مشتری ها با ورود به مغازه خرید میکنند و با هر بار خرید، تعداد آن جنس کم میشود. همچنین مغازه دار میتواند جنس خریده و با اینکار تعداد جنس مورد نظر را زیاد کند.

دستورات ورودی

- `add store_1 productA 2 2 productB 3 1 productC 2 1 productD 7 1 productE 6 2`

  این دستور به این معنی است که مغازه ای به نام store_1 در این فروشگاه افتتاح میشود. پس از آمدن اسم مغازه به ترتیب، نام محصولات موجود در مغازه، تعداد آن ها و قیمت آن ها آورده میشود. برای مثال در این مغازه، از محصول productE که قیمت آن ۲ واحد پول است، ۶ نمونه داریم. دقت کنید از سری ورودی های غلط این است اگر پس از نام مغازه، محصولات در دستور وارد نشوند، شما نباید این مغازه را ایجاد کنید. به عبارت دیگر، مغازه ای بدون محصول ساخته نمیشود.

- `buy from store_1 productC 2`

  یک مشتری از مغازه store_1، دو تا از کالای productC خریداری میکند. دقت کنید دیگر این مغازه این جنس را ندارد و اگر دوباره با دستور خرید این جنس رو به رو شود باید پیغام we don't have it(همگی با حروف کوچک) چاپ شود.(اگر محصول درخواستی جز محصولات مغازه نبود و یا به آن تعداد محصول نداشت باید این عبارت چاپ شود). پس از اجرا شدن این دستور، مشتری مبلغ "تعداد * قیمت محصول درخواستی" را میپردازد و به صندوق مغازه این مقدار اضافه میشود. دقت کنید که در این دستور، قیمت محصول به شما داده نشده و باید از قیمتی که فروشگاه روی محصول گذاشته است استفاده کنید.

- `buy for store_1 productF 12 9`

  به این معنی است که این مغازه از کالای productF، که قیمت هر نمونه آن ۹ واحد پول است، ۱۲ نمونه خریداری میکند. پس از اجرای این دستور مبلغ ۱۲*۹ از صندوق مغازه کم میشود. دقت کنید که اگر صندوق مغازه این مقدار پول را برای پرداخت نداشته باشد باید عبارت not enough money چاپ شود. همچنین در صورت اجرا شدن این خط دستور، این محصول به محصولات مغازه اضافه میشود.

- `change price productA of store_1 5`

  این دستور به این معنی است که ازین پس قیمت محصول productA از مغازه store_1 عوض شده و برابر با ۵ واحد پول میگردد.

- `show store_1`

  با وارد شدن این دستور، تمامی اجناس این فروشگاه و تعداد و قیمت آن ها به ترتیب صعودی بر اساس تعداد موجودی چاپ شوند، در صورتی که موجودی دو محصول برابر بود، آن که از لحاظ الفبایی زودتر می‌آید چاپ شود.

```
ProductA 2 5
productB 3 1
productE 6 2
productD 7 1
productF 12 9
```

دقت کنید چون جنس productC تمام شده است، نباید عبارت productC 0 1 چاپ شود.

همچنین هر مغازه ای که افتتاح میشود در صندوقش ۲۰ واحد پول دارد. به گرفتن دستورات تا زمانی ادامه دهید که دستور end در ورودی بیاید.

خطاهای ورودی و پیام‌های خطا

- اسم هیچ مغازه ای با عدد شروع نمیشود. در صورت وارد شدن ورودی به این صورت، نباید این مغازه را اضافه کنید.
- اگر در دستوری با اسم مغازه ای رو به رو شدید که در مرکز خرید وجود ندارد باید عبارت no such store را خروجی بدهید.
- اگر در دستور اضافه کردن مغازه، قیمت محصولی یا تعداد آن وارد نشود پیغام invalid input را چاپ کنید.
- در دستور اضافه کردن مغازه، اگر نام محصولی دو بار آورده شود، خطا بوده و باید از انجام این دستور صرف نظر شود و عبارت invalid input چاپ شود.
- در دستور خرید اجناس برای مغازه، اگر تنها یک عدد وارد شود، خطا بوده و نباید کاری انجام شود و باید عبارت invalid input را چاپ کنید.
- در دستور تغییر قیمت اجناس، در صورت نیامدن قیمت تنها باید عبارت invalid input را خروجی بدهید.

دقت کنید تنها برای موارد ذکر شده خروجی invalid input را چاپ کنید و در صورتی که دستوری یک یا بیش از یکی از مواردی که منجر به خروجی invalid input را در خود داشت، همین خروجی را چاپ کنید.

نکات شئ‌گرایی

که در این سوال، تمامی کلاس ها باید مطابق با نمودار UML داده شده پیاده سازی شوند. در صورتی که نیاز به پیاده‌سازی یک متد یا کلاس کمکی که در UML ذکر نشده دارید، می‌توانید آن را پیاده کنید اما چارچوب کلی کد شما بایستی مبتنی بر UML داده‌شده باشد و باید متدها و متغیرهای آن را پیاده کنید.

نمودار UML کلاس‌ها

«ShoppingCenter»
- stores : ArrayList<Store>
+ addStore(storeName : String, products : HashMap<Product, Integer>) : void + show(storeName : String) : void + sell(storeName : String, productName : String, number : int) : void + buy(storeName : String, productName : String, number : int, price : int) : void + changePrice(storeName : String, productName : String, newPrice : int) : void

«Store»
- name : String - money : int - products : HashMap<Product, Integer>
+ setName(name: String) : void + getName( ) : String + start(storeName : String, products : HashMap<Product, Integer>) : void + getProducts( ) : HashMap<Product, Integer> + changePrice(productName : String, newPrice : int) : void + buy(product : Product, number : int) : void + sell(productName : String, number : int) : void + show( ) : void

«Product»
- name : String - price : int
+ setName(name: String) : void + getName( ) : String + setPrice(price: int) : void + getPrice( ) : int

نمونه‌ی ورودی و خروجی

نمونه‌ی ورودی	نمونه‌ی خروجی
add store_1 p1 1 1 p2 1 2 p3 1 4 p4 1 5 add store_2 p1 1 1 p1 1 3 add store_2 add store_2 p1 1 p2 3 4 add store_2 p1 10 5 p2 3 5 p5 8 1 buy from store_1 p1 1 buy from store_1 p2 1 add 3store p6 3 2 buy for 3store p7 1 2 buy for store_2 p3 4 buy for store_1 p1 3 1 change price p3 of store_1 5 show store_1 show store_2	 invalid input invalid input no such store invalid input p3 1 5 p4 1 5 p1 3 1 p2 3 5 p5 8 1 p1 10 5