

سوال اول : ماشین حساب ماتریسی تکرار پذیر

مهدی، متین و سجاد برای درس جبرخطی نیاز به برنامه ی ماشین حسابی دارند که برایشان سه عمل جمع و تفریق و ضرب ماتریس ها را بصورت تکرارشونده انجام دهد. اما وقت کافی برای طراحی چنین ماشین حسابی ندارند. بنابراین از شما برای طراحی این برنامه کمک خواسته اند.

برای پیاده سازی تمرین پیشنهاد می شود از طراحی زیر استفاده کنید.

در ابتدا یک کلاس Matrix تعریف کنید. هدف این کلاس پیاده سازی متدهای مربوط به کار با ماتریس ها است.

```
public class Matrix{;
```

یک متد سازنده برای این کلاس تعریف کنید که ابعاد ماتریس را می گیرد.

rows: تعداد سطرهای ماتریس

columns: تعداد ستون های ماتریس

```
public Matrix(int rows, int columns){};
```

یک متد برای مقدار دهی خانه های ماتریس بنویسید.

row: سطر آرگمان

column: ستون آرگمان

value: مقداری که می خواهیم در سطر و ستون مورد نظر قرار بدهیم

```
public void setValue(int row, int column, int value){};
```

یک متد نیز برای دریافت مقدار آرگمان ماتریس بنویسید.

```
public int getValue(int row, int column){};
```

سپس متدهای ضرب و جمع و تفریق را به تعریف اضافه کنید.

```
public Matrix multiply(Matrix m){};
```

پیاده سازی تابع multiply باید اینگونه باشد که اگر داشته باشیم:

```
Matrix m1;  
Matrix m2;  
Matrix m3 = m1.multiply(m2)
```

ماتریس m^3 اینگونه تعریف می شود: (ماتریس m_1 عملوند اول است)

$$m_3 = m_1 \times m_2$$

با همین شیوه متدهای جمع و تفریق را نیز تعریف کنید.

```
public Matrix add(Matrix m){};  
public Matrix subtract(Matrix m){};
```

نکته: در صورتی که عملیات ضرب بین دو ماتریس تعریف نشده بود (ابعاد دو ماتریس برای ضرب سازگار نبودند)، متد باید null بازگرداند.

فرمت ورودی برنامه بدین صورت است که در خط اول ورودی، تعداد تکرار عملیات خواسته شده (n) را می گیرد (که در ادامه نحوه ی استفاده از آن توضیح داده خواهد شد)؛ سپس در خط بعدی، به ترتیب تعداد سطر (r) و ستون (c) ماتریس اول وارد می شود و در r خط بعدی، در هر خط c عدد می آید که نمایش دهنده ی سطرهای ماتریس اول هستند؛

پس از دریافت ماتریس اول، در خط بعدی، تعداد سطر و ستون ماتریس دوم می آید و در خطوط بعد از آن، اعداد ماتریس دوم مانند ماتریس اول داده می شوند.

در خط بعدی، یکی از سه عبارت + یا - یا * می آید که مشخص می کند چه عملیاتی باید روی این دو ماتریس انجام شود. (ماتریس اول، عملوند اول است)

پس از دریافت نوع عملیات ماتریسی، برنامه باید عملیات خواسته شده را بین این دو ماتریس انجام دهد؛ سپس نتیجه ی بدست آمده را به جای ماتریس اول قرار دهد و n بار (عددی که در خط اول ورودی داده شده بود) همان عملیات را به این شکل بین ماتریس آخرین نتیجه (به عنوان ماتریس اول) و ماتریس دوم (که در ورودی داده شده بود) انجام دهد و در نهایت ماتریس نتیجه را خروجی دهد (هر سطر در یک خط).

در صورتیکه حداقل یکی از عملیات مورد نیاز قابل انجام نبود (مانند ضرب)، عبارت "ERROR" را چاپ کنید (با حروف بزرگ). برای مثال اگر n برابر با ۰ باشد ولی دومین ضرب ممکن نباشد، بدون ادامه دادن عملیات عبارت "ERROR" را چاپ کرده و برنامه به پایان می رسد.

توجه: اگر n یک باشد، یعنی فقط یکبار عملیات داده شده را روی دو ماتریس ورودی انجام دهد.

ورودی نمونه 1 :

3
2 2
1 2
3 4
2 2
2 0
0 1
*

خروجی نمونه 1 :

8 2
24 4

ورودی نمونه 2 :

2
1 2
5 6
1 2
7 8
*

خروجی نمونه 2 :

ERROR

سوال دوم : دستگاه فروش خوراکی

با توجه به زیاد شدن تعداد دانشجویان دانشگاه و محدود بودن فضای دگه ی آقا یعقوب، دانشگاه تصمیم به نصب تعدادی دستگاه فروش خوراکی در دانشکده ها گرفته است؛ اما کنترل این دستگاه ها نیازمند یک نرم افزار مناسب است و دانشگاه طراحی این برنامه را به شما محول کرده است.

برای این برنامه کلاس های زیر را تعریف کنید :

نکته: تعریف بعضی از متدها مانند getter و setter ها در لیست متدها آورده نشده است. در صورت نیاز به آنها در یک طراحی خوب خودتان آنها را اضافه کنید.

ابتدا کلاس دستگاه فروش خوراکی را بنویسید:

```
public class VendingMachine{};
```

نکته: برای کلاس ها و متدها و پارامترها از نام های مناسب استفاده کنید.

این کلاس حداقل باید شامل موارد زیر باشد:

فیلدها:

1. id (شماره ی دستگاه که یک عدد صحیح مثبت است) - int
2. saleCount (تعداد محصول فروخته شده توسط این دستگاه) - int
3. saleAmount (مبلغی که دستگاه فروخته است) - int
4. products (HashMap از محصولات و تعدادشان) -
HashMap<Product,Integer>

متدها:

Buy، که در ورودی یک محصول را می گیرد و در صورتی که موجودی آن بیشتر از صفر باشد، از موجودی آن محصول را یک واحد کم کرده و به saleCount دستگاه فعلی یک واحد اضافه می کند و همچنین قیمت آن محصول را به saleAmount دستگاه اضافه می کند و true بر می گرداند؛ ولی اگر موجودی کافی نبود هیچ کاری انجام نمی دهد و false بر می گرداند.

```
public boolean buy (Product product) {};
```

addProduct: این تابع وظیفه دارد تا محصولی را به دستگاه اضافه کند. ورودی‌های این تابع شامل یک Product است که همان محصولی است که باید اضافه شود و یک عدد که تعداد آن محصول را بیان می‌کند.

```
public void addProduct(Product p, int amount){};
```

کلاس محصول را نیز به صورت زیر تعریف می‌کنیم:

```
public class Product{};
```

کلاس محصول، حداقل باید شامل موارد زیر باشد:

فیلدها:

1. id (شناسه محصول) – int

2. name (نام محصول) – String

3. price (قیمت محصول) – int

نحوه‌ی ورودی گرفتن برنامه بدین صورت است:

در خط اول تعداد محصولات متمایزی که قرار است در دستگاه‌ها به فروش برسد آورده می‌شود (m). سپس در خطوط بعدی مشخصات محصولات را با فرمت زیر وارد می‌کنیم:

شناسه محصول

نام محصول

قیمت محصول

بعد از آنکه مشخصات محصولات وارد شد، در خط بعدی تعداد دستگاه‌های فروش خوراکی (n) می‌آید؛ در n خط بعدی، به ترتیب اطلاعات محصولات دستگاه‌ها با id (شماره دستگاه) یک تا n می‌آیند. (خط اول مربوط به دستگاه 1 و ... و خط n ام مربوط به دستگاه n ام است.) در هر خط، ابتدا یک عدد (p) می‌آید که نشان دهنده‌ی تعداد کل محصولاتی است که در ادامه‌ی خط آمده است (با احتساب تکراری‌ها)؛ سپس در ادامه‌ی خط، شناسه p محصول می‌آید که باید به دستگاه متناظر اضافه شوند. (وقتی شناسه یک محصول چندبار می‌آید یعنی موجودی آن باید اضافه شود و شناسه‌های مشابه الزاماً پشت سر هم نیستند).

پس از اتمام ورود شناسه‌های محصولات برای تمام دستگاه‌ها، در خط بعدی، تعداد خریدها (b) می‌آید و در b خط بعدی، در هر خط، ابتدا شماره ی دستگاه و سپس شناسه محصول جهت خریداری شدن ذکر می‌شود.

نهایتاً، در n خط (به تعداد دستگاه‌های فروش خوراکی) در هر خط ابتدا شماره ی دستگاه چاپ شده و پس از کاراکتر "؛" ، ابتدا saleCount آن دستگاه و سپس کاراکتر "," و در نهایت مقدار saleAmount آن دستگاه چاپ می‌شود.

توجه: شماره ی دستگاه‌ها به ترتیب ورودی محصولات است و از 1 شروع می‌شود.

ورودی نمونه 1 :

```
3
1
hibye
100
2
gumdrop
200
3
apple juice
10
3
5 1 2 3 1 1
3 3 2 3
2 3 1
7
2 2
2 2
1 3
1 1
3 3
3 3
1 1
```

خروجی نمونه 1 :

```
1:3,210
2:1,200
3:1,10
```