

مجموعه ها

در این سوال قصد داریم کدی بنویسیم که مجموعه های ریاضی را پیاده سازی کند و حاصل عملیات های بین آنها رو محاسبه بنماید. تضمین می شود که اعضای مجموعه ها فقط int باشند.

کد شما باید قادر به انجام دستورات زیر باشد:

تعریف یک مجموعه :

برای مثال با دادن ورودی زیر به برنامه، مجموعه A تعریف و ذخیره می شود

$A = \{1, 2, 3\}$

- نکته: لازم به ذکر نیست که در مجموعه ها تکرار اعضا وجود ندارد و در صورتی که ورودی دو عضو یکسان داشت، باید یکبار ذخیره شود.
- نکته : مجموعه ها را باید با یک حرف بزرگ نمایش داده شوند و تضمین می شود این قاعده در دادن ورودی رعایت شود.
- نکته: قرار دادن اسپیس های اضافی (و کم کردن اسپیس ها) نباید در کارکرد کد خللی ایجاد کند، برای مثال، کد شما باید تمام وروی های زیر را هندل کند.

$A = \{ 1, 2, 3\}$

$A = \{1, 2, 3\}$

$A=\{1,2,3\}$

در صورتی که مجموعه با موفقیت ساخته شد باید پیام زیر به کاربر نمایش داده شود :

set created successfully

در صورتی که مجموعه A قبلا ساخته شده باشد کد شما باید مقادیر آن رو بر اساس ورودی جدید به روز کند و پیام زیر را به کاربر دهد:

set updated successfully

چاپ یک مجموعه :

ورودی :

```
print B
```

در صورتی که مجموعه B قبلا تعریف شده باشد، باید اعضای آن را مشابه زیر چاپ کنید. توجه کنید که اعضا باید از کوچک به بزرگ مرتب شده باشند.

```
B = {3,4,5}
```

در صورتی که مجموعه B تعریف نشده باشد باید پیام خطای زیر را چاپ کنید:

```
B is not defined
```

اضافه کردن یک عضو به یک مجموعه :

ورودی :

```
add 7 to C
```

در صورتی که مجموعه C تعریف نشده باشد باید پیام خطای زیر را چاپ کنید:

```
C is not defined
```

در صورتی که عدد 7 از قبل عضو C باشد باید پیام زیر چاپ شود:

```
7 is already in C
```

در صورت موفقیت پیام زیر به کاربر نمایش داده می‌شود

```
added successfully
```

اعمال ساده بین مجموعه ها :

کد شما باید توانایی محاسبه اشتراک، اجتماع و تفاضل دو مجموعه را داشته باشد.

نکته : اشتراک را با * و اجتماع را با + نشان می‌دهیم

ورودی :

```
E = A*B  
print E
```

اشتراک A و B محاسبه می‌شود و حاصل در E ذخیر می‌شود. خروجی:

```
E = {3}
```

در صورتی که A یا B تعریف نشده باشند، پیام خطای زیر برای کاربر چاپ شود:

```
some sets are not defined
```

چاپ زیر مجموعه های یک مجموعه:

ورودی :

```
subsets of A
```

کد شما باید تمام زیر مجموعه های مجموعه A را چاپ کند(ترتیب اهمیتی ندارد). مانند زیر :

{}
{3}
{2}
{2,3}
{1}
{1,3}
{1,2}
{1,2,3}

در صورتی که مجموعه A تعریف نشده باشد باید پیام خطای زیر نشان داده شود:

B is not defined

محاسبه عبارات جبر مجموعه ها

کدتان را به گونه ای توسعه بدهید که توانایی محاسبه عبارات پیچیده تری از جبر مجموعه ها را داشته باشد. مثال :

عبارت 1:

$$D = A \cap (B \cup C)$$

ورودی متناظر با عبارت 1:

$$D = A * (B + C)$$

عبارت 2:

$$G = (E \cup F) \cap ((A - B) \cup ((C \cap B) - D))$$

ورودی متناظر با عبارت 2:

$$G = (E + F) * ((A - B) + ((C * B) - D))$$

این عبارات شامل پرانتز های تو در تو می باشند و کد شما باید حاصل نهایی را با رعایت کامل ترتیب عملیات ها و سایر قواعد جبر مجموعه ها محاسبه کند.

- نکته: در صورتی که یک یا چند تا از مجموعه های موجود در عبارت، تعریف نشده باشند، باید پیام زیر چاپ شود:

some sets are not defined

- نکته: همانطور که گفته شد، اشتراک با * و اجتماع با + نشان داده می شود.
- نکته: تضمین می شود که ورودی، پرانتز گذاری مناسبی داشته باشد. برای مثال اگر اجتماع سه مجموعه A ,B ,C خواسته شود، به صورت زیر نوشته می شود:

$$D = A+(B+C)$$

در پایان با دستور end کار برنامه تمام می شود.

زبان ساختاریافته کوئری

- محدودیت زمان: 5 ثانیه
- محدودیت حافظه: 128 مگابایت

هدف از این تمرین آشنایی اولیه با **Structured Query Language** و پیاده‌سازی یک نوع **RDBMS** ساده (که به صورت **offline** کار می‌کند) با زبان جاوا است. **SQL** یک زبان استاندارد برای ذخیره‌سازی، تغییر و دریافت داده‌ها در یک پایگاه داده (**Database**) است. باید بدانید این زبان در سازمان بین‌المللی استانداردسازی (ISO) ثبت شده است. وظیفه شما در این تمرین، نوشتن برنامه‌ای است که دستورات **SQL** را ورودی بگیرد و تغییرات لازم را بر روی پایگاه داده اعمال کند. اما قبل از آشنایی با **SQL** لازم است با ساختار یک دیتابیس آشنا شویم.

- اگر با مفاهیم دیتابیس و **SQL** آشنایی دارید می‌توانید مستقیم به سراغ خواسته‌های سوال بروید.

هر پایگاه داده از تعدادی جدول (**TABLE**) تشکیل شده است و با یک نام مشخص می‌شود. برای مثال جدول مشخصات مشتریان یک شرکت :

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

همچنین هر جدول از مفاهیمی به نام **record** و **column** و **Field** تشکیل شده است. اگر بخواهیم برای هر کدام تعریف درستی ارائه کنیم می‌توان گفت :

- Record:

هر ورودی که در یک جدول وجود دارد، یک record یا همان ردیف (row) محسوب می شود. آیا می دانید چرا به ورودی های جدول ردیف می گوییم؟ به این دلیل که اگر بخواهید یک ورودی جدید را در جدولی وارد کنید، باید آن را به صورت یک ردیف وارد کنید!

- Column :

ستون ها در جدول، داده های عمودی هستند که اطلاعات یک دسته از فیلدها را در خود دارند.

- Field :

همانطور که گفتیم هر جدول دارای تعدادی ستون و ردیف است. تلاقی هر ردیف و ستون را یک فیلد یا یک خانه از جدول گوییم.

تصویر زیر در درک تعاریف بالا به شما کمک می کند :

The diagram shows a table with 7 columns and 5 rows. Red arrows and boxes highlight specific parts: 'Column' points to the header row; 'Record' points to the first three rows; 'Field' points to three specific cells: 'Antonio Moreno', 'UK', and 'Berglunds snabbköp'.

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

حال به سراغ آشنایی با چند دستور ساده SQL می رویم :

SELECT

دستور **SELECT** برای انتخاب داده ها از یک پایگاه داده مورد استفاده قرار می گیرد. داده هایی که با دستور **SELECT** برمی گردند در جدول نتایج به نام **result_set** ذخیره می شوند و به عنوان خروجی نمایش داده می شوند. این دستور به دو شکل مورد استفاده می گردد :

```
SELECT * FROM table_name;
```

این دستور تمام داده‌های موجود از جدولی به نام table_name را خروجی می‌دهد. برای مثال اگر از دستور زیر استفاده کنیم :

```
SELECT * FROM Customers;
```

باید تمام جدول مشتریان که در قسمت قبل دیدیم با تمام ردیف‌ها و ستون‌های آن به نمایش در بیاید :

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

شکل دیگر استفاده از این دستور به روش زیر است :

```
SELECT column1,column2, ... FROM table_name;
```

با استفاده از این دستور تنها ستون‌های ذکر شده به همین اسم و به همین ترتیب در دستور، از جدولی به نام table_name انتخاب و در جدول result_set ذخیره و در خروجی نمایش داده می‌شوند. برای مثال اگر از دستور زیر استفاده کنیم :

```
SELECT City, CustomerName FROM Customers;
```

جدول زیر در خروجی نمایش داده می‌شود:

City	CustomerName
Berlin	Alfreds Futterkiste
México D.F.	Ana Trujillo Emparedados y helados
México D.F.	Antonio Moreno Taquería
London	Around the Horn
Luleå	Berglunds snabbköp

- در صورتی که جدولی به اسم ذکر شده، در دیتابیس ما وجود نداشت و یا اینکه اسامی ستون‌های ذکر شده در ورودی با اسامی ستون‌های جدول همخوانی نداشت؛ باید عبارت "ERROR" در خروجی چاپ شود.
- دقت کنید که ترتیب ستون های result_set به ترتیب اسامی ستون هایی هستند که در ورودی آمده است.
- تکرار اسامی ستون ها نباید مشکلی ایجاد کند و تکرار های اسم یک ستون در ورودی باید نادیده گرفته شوند به مثال زیر دقت کنید :

```
SELECT CustomerName, City, CustomerName, Country, Country, City FROM Customers;
```

CustomerName	City	Country
Alfreds Futterkiste	Berlin	Germany
Ana Trujillo Emparedados y helados	México D.F.	Mexico
Antonio Moreno Taquería	México D.F.	Mexico
Around the Horn	London	UK
Berglunds snabbköp	Luleå	Sweden
Blauer See Delikatessen	Mannheim	Germany
Blondel père et fils	Strasbourg	France
Bólido Comidas preparadas	Madrid	Spain
Bon app'	Marseille	France
Bottom-Dollar Marketse	Tsawassen	Canada
B's Beverages	London	UK

INSERT INTO

این دستور برای وارد کردن داده جدید به دیتابیس استفاده می‌شود. استفاده از آن به دو شکل صورت می‌گیرد :

```
INSERT INTO table_name VALUES (value1,value2,value3,...);
```

این دستور یک ریکورد(ردیف) جدید با مقادیر داده شده در ورودی را به جدولی به اسم table_name اضافه می‌کند و در خروجی عبارت زیر چاپ می‌شود :

```
You have made changes to the database. Row affected: 1
```

برای مثال اگر ورودی زیر را به برنامه بدهیم. یک ردیف به جدول مشتریان با مقادیر داده شده اضافه می‌شود.

```
INSERT INTO Customers VALUES ('93','Roham kaveie','Roham','Ekbatan','Tehran','
```

اگر از دستور SELECT برای نمایش جدول مشتریان استفاده کنیم می‌توانیم ریکورد اضافه شده را در ته جدول پیدا کنیم :

```
SELECT * FROM Customers;
```

	Importadora		Mercado, 12			
89	White Clover Markets	Karl Jablonski	305 - 14th Ave. S. Suite 3B	Seattle	98128	USA
90	Wilman Kala	Matti Karttunen	Keskuskatu 45	Helsinki	21240	Finland
91	Wolski	Zbyszek	ul. Filtrawa 68	Walla	01-012	Poland
93	Roham kaveie	Roham	Ekbatan	Tehran	12345	Iran

شکل دیگر دستور INSERT INTO به شکل زیر استفاده می‌گردد :

```
INSERT INTO Customers (CustomerID,CustomerName,City) VALUES ('94','Roham kavei
```

با استفاده از این دستور یک ریکورد(ردیف) با مقادیر داده شده به ترتیب در ستون‌های ذکر شده قرارگرفته شده و به جدول اضافه می‌شود. و در خروجی عبارت "You have made changes to the database. Raw" affected: 1 چاپ می‌شود.

اگر با دستور SELECT جدول را در خروجی چاپ کنیم می‌توانیم ریکورد اضافه شده را در خروجی ببینیم :

89	White Clover Markets	Karl Jablonski	305 - 14th Ave. S. Suite 3B	Seattle	98128	USA
90	Wilman Kala	Matti Karttunen	Keskuskatu 45	Helsinki	21240	Finland
91	Wolski	Zbyszek	ul. Filtrowa 68	Walla	01-012	Poland
93	Roham kavele	Roham	Ekbatan	Tehran	12345	Iran
94	Roham kavele	null	null	Tehran	null	null

- همانطور که مشاهده می‌کنید در فیلدهایی که مقادیر آنها داده نشد؛ مقدار null ذخیره شده است.
- اگر تعداد مقادیر داده شده و اسامی ستون‌های داده شده با ستون‌های جدول همخوانی نداشته باشد باید عبارت "ERROR" در خروجی چاپ شود.

WHERE

از این عبارت برای اضافه کردن شرط به دستور های دیگر SQL استفاده می‌شود. به مثال زیر دقت کنید:

```
SELECT * FROM Customers WHERE Country = 'France';
```

این دستور در result_set تنها ریکوردهایی که مقدار ستون Country آنان برابر با مقدار 'France' هستند را ذخیره و در خروجی نمایش می‌دهد.

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
7	Blondel père et fils	Frédérique Citeaux	24, place Kléber	Strasbourg	67000	France
9	Bon app'	Laurence Lebihans	12, rue des Bouchers	Marseille	13008	France
18	Du monde entier	Janine Labrune	67, rue des Cinquante Otages	Nantes	44000	France
23	Folies gourmandes	Martine Rancé	184, chaussée de Tournai	Lille	59000	France
26	France restauration	Carine Schmitt	54, rue Royale	Nantes	44000	France
40	La corne d'abondance	Daniel Tonini	67, avenue de l'Europe	Versailles	78000	France
41	La maison d'Asie	Annette Roulet	1 rue Alsace-Lorraine	Toulouse	31000	France
57	Paris spécialités	Marie Bertrand	265, boulevard Charonne	Paris	75012	France
74	Spécialités du monde	Dominique Perrier	25, rue Lauriston	Paris	75016	France
84	Victuailles en stock	Mary Saveley	2, rue du Commerce	Lyon	69004	France
85	Vins et alcools Chevalier	Paul Henriot	59 rue de l'Abbaye	Reims	51100	France

یا در مثال زیر، دستور زیر تنها ID مشتریانی که شهر سکونت آنان پاریس است را به عنوان خروجی نمایش می‌دهد :

```
SELECT CustomerID FROM Customers WHERE City = 'Paris';
```

CustomerID
57
74

- در زبان SQL می‌توانید با عبارات AND و OR و NOT و همینطور اپراتورهایی مانند <, >, =, <=, >=, != شرایط پیچیده‌تری گذاشت و یا حتی با عبارت LIKE پترن هایی مشابه به ریجکس را برای پیدا کردن ریکورد های خاص استفاده کرد؛ اما برای سادگی در این تمرین تضمین می‌شود هیچکدام از آنها ورودی داده نمی‌شود و تنها ورودی‌هایی مشابه با دو مثال ذکر شده ورودی داده می‌شود.

UPDATE

این دستور برای تغییر و بروزرسانی مقادیر ذخیره شده در پایگاه داده استفاده می‌شود. ساختار کلی این دستور به شکل زیر است :

```
UPDATE table_name SET column1 = value1, column2 = value2, ... WHERE condition;
```

این دستور مقادیر خانه‌های ریکورد هایی که در شرایط condition صدق می‌کنند را بروزرسانی می‌کند و مقادیر ذکر شده را در ستون‌های گفته شده جایگذاری می‌کند. و عبارت زیر را در خروجی چاپ می‌کند:

```
You have made changes to the database. Row affected: %d
```

- تعداد ریکوردهایی است که در اثر این دستور تغییر پیدا کرده‌اند %d
- استفاده نشود تمام ریکوردهای جدول انتخاب و آپدیت می‌شوند WHERE توجه کنید اگر از عبارت

حال به مثال دیگری توجه کنید:

```
UPDATE Customers SET CustomerName = 'Arsham',ContactName = 'AK' WHERE Customer
```

90	Wilman Kala	Matti Karttunen	Keskuskatu 45	Helsinki	21240	Finland
91	Wolski	Zbyszek	ul. Filtrowa 68	Walla	01-012	Poland
93	Arsham	AK	Ekbatan	Tehran	12345	Iran
94	Roham kavele	null	null	Tehran	null	null

همانطور که مشاهده می‌کنید تنها ریکوردی که شماره ID آن برابر با 93 بود مقادیر ستون‌های CustomerName و ContactName آن به مقادیر ذکر شده در ورودی تغییر پیدا کرد.

به مثال دیگری توجه کنید:

```
UPDATE Customers SET Country = 'Iran' WHERE Country = 'USA';
```

با اجرا این دستور تمام ریکوردهای جدول Customers که مقدار ستون Country آنان برابر با 'USA' است با مقدار 'Iran' جایگذاری می‌شود. و خروجی زیر چاپ می‌شود :

```
You have made changes to the database. Row affected: 13
```

و اگر پس از این دستور، دستور زیر را ورودی بدهیم با این خروجی مواجه می‌شویم :

```
SELECT * FROM Customers WHERE Country = 'Iran';
```

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
32	Great Lakes Food Market	Howard Snyder	2732 Baker Blvd.	Eugene	97403	Iran
36	Hungry Coyote Import Store	Yoshi Latimer	City Center Plaza 516 Main St.	Elgin	97827	Iran
43	Lazy K Kountry Store	John Steel	12 Orchestra Terrace	Walla Walla	99362	Iran
45	Let's Stop N Shop	Jaime Yorres	87 Polk St. Suite 5	San Francisco	94117	Iran
48	Lonesome Pine Restaurant	Fran Wilson	89 Chiaroscuro Rd.	Portland	97219	Iran
55	Old World Delicatessen	Rene Phillips	2743 Bering St.	Anchorage	99508	Iran
65	Rattlesnake Canyon Grocery	Paula Wilson	2817 Milton Dr.	Albuquerque	87110	Iran
71	Save-a-lot Markets	Jose Pavarotti	187 Suffolk Ln.	Boise	83720	Iran
75	Split Rail Beer & Ale	Art Braunschweiler	P.O. Box 555	Lander	82520	Iran
77	The Big Cheese	Liz Nixon	89 Jefferson Way Suite 2	Portland	97201	Iran
78	The Cracker Box	Liu Wong	55 Grizzly Peak Rd.	Butte	59801	Iran
82	Trail's Head Gourmet Provisioners	Helvetius Nagy	722 DaVinci Blvd.	Kirkland	98034	Iran
89	White Clover Markets	Karl Jablonski	305 - 14th Ave. S. Suite 3B	Seattle	98128	Iran
93	Arsham	AK	Ekbatan	Tehran	12345	Iran

DELETE

این دستور برای حذف داده‌ها در یک پایگاه داده استفاده می‌شود. ساختار کلی این دستور به شکل زیر است :

```
DELETE FROM table_name WHERE condtions;
```

این دستور تمام ریکوردهایی که در شرایط table_name وجود دارند را حذف می‌کند و سپس عبارت زیر را به عنوان خروجی چاپ می‌کند :

You have made changes to database. Row affected: %d

- تعداد ریکوردهایی است که در اثر این دستور تغییر پیدا کرده‌اند %d
- استفاده نشود تمام ریکوردهای جدول انتخاب و حذف می‌شوند. به WHERE توجه کنید اگر از عبارت :مثال زیر دقت کنید

```
DELETE FROM Customers WHERE Country = 'UK';
```

پس از اجرای این دستور اگر با دستور SELECT ریکوردهایی که مقدار ستون Country آنان برابر با 'UK' است را خروجی بگیریم. در خروجی تنها عبارت "No result" نمایش داده می‌شود (چون تمام ریکوردهایی با این مشخصه از قبل حذف شده‌اند):

```
SELECT * FROM Customers WHERE Country = 'UK';
```

.

- توجه کنید در صورت حذف تمام ریکوردهای یک جدول، آن جدول خالی می‌شود اما حذف نمی‌شود و هنوز جدولی به این اسم در پایگاه داده ما موجود است. (صرفاً خالی از ریکورد است).

CREATE TABLE

با این دستور می‌توانید یک جدول بسازید. ساختار کلی این دستور به شکل زیر می‌باشد:

```
CREATE TABLE table_name (column1,column2,column3,...);
```

با اجرای این دستور یک TABLE به اسم table_name با ستون‌های column1,column2,... ساخته می‌شود و عبارت زیر به عنوان خروجی چاپ می‌شود:

```
You have made changes to the database.
```

لازم به ذکر است که جدول ساخته‌شده در ابتدا هیچ ریکوردي ندارد. برای مثال:

```
CREATE TABLE Company (CompanyID,CompanyName,Address,PhoneNumber,City,Country);
```

با اجرای این دستور یک جدول به اسم Company با ستون‌هایی که در ورودی ذکر شده است ساخته می‌شود. که اگر با دستور SELECT آن را نمایش دهیم عبارت No result. خروجی داده می‌شود (چون هنوز

هیچ ورودی به جدول داده نشده است).

اگر جدولی با این اسم قبلا وجود داشته است یا اسامی ستون‌های ذکر شده در ورودی تکراری باشد باید عبارت ERROR چاپ شود.

DROP TABLE

از این دستور برای حذف جدول‌های یک پایگاه داده استفاده می‌شود. ساختار کلی این دستور به شکل زیر است :

```
DROP TABLE table_name;
```

این دستور جدولی به اسم table_name را از دیتابیس به طور کل حذف می‌کند و دیگر جدولی به این اسم نداریم. و در خروجی عبارت زیر را چاپ می‌کند:

```
You have made changes to the database.
```

برای مثال برای حذف جدول Company که در قسمت قبل ساختیم باید بنویسیم :

```
DROP TABLE Company;
```

• اگر جدولی به این اسم وجود نداشته باشد باید در خروجی عبارت "ERROR" چاپ شود.

خواسته های سوال

از شما می خواهیم که ساختار یک دیتابیس را پیاده سازی کنید و برنامه خود را به گونه ای توسعه دهید که بتواند دستورات

۱. CREATE TABLE

۲. DROP TABLE

۳. INSERT INTO

۴. SELECT (WHERE)

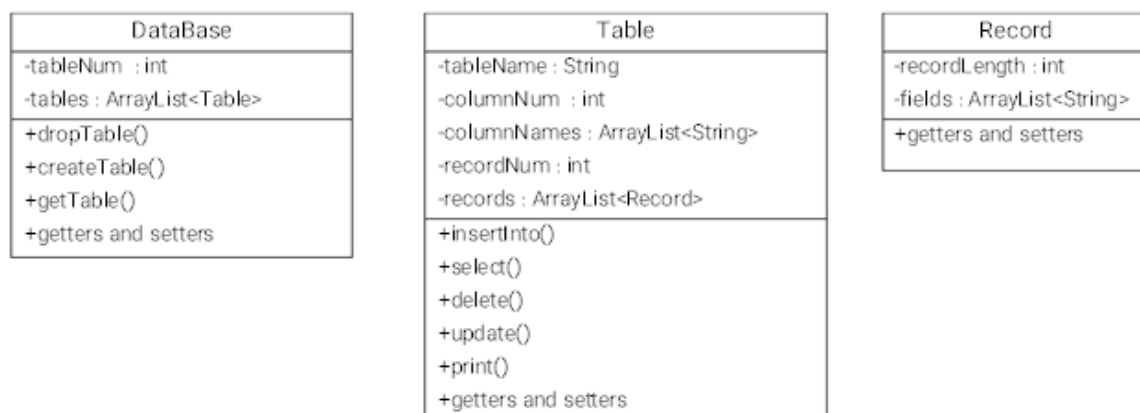
۵. DELETE (WHERE)

۶. UPDATE (WHERE)

را ورودی بگیرد و تغییرات لازم را بر اساس تعریف هر دستور بر روی دیتابیس اعمال کند.

به این منظور از شما می‌خواهیم که ابتدا در برنامه خود دو کلاس به اسامی Manager و InputProcessor به وجود آورید. کلاس InputProcessor وظیفه گرفتن ورودی از کاربر و صدا زدن توابع لازم از Manager بر پایه ورودی داده شده و چاپ کردن خروجی لازم را بر عهده دارد.

در ادامه باید 3 کلاس به اسامی Table و Record و Datebase با attribute و method هایی که در classdiagram های زیر مشخص شده‌اند به وجود آورید. ساختار هر کدام به شکل زیر می‌باشد :



* نکات :

- هرگونه پیاده‌سازی ساختار کلاس‌های نامبرده شده که در ازای ورودی، خروجی درست و معتبر چاپ شود در صورت رعایت اصول شیگرایی و حفظ چارچوب‌های کلی یک پایگاه داده، قابل قبول است.
- اسامی جدول ها و ستون ها فقط باید از اعداد و حروف لاتین (بزرگ یا کوچک) تشکیل شده باشند و این اسامی نباید با عدد شروع شوند و در صورت تلاش کاربر برای نام گذاری خلاف این قاعده، باید عبارت "ERROR" چاپ شود.
- برای چاپ کردن جدول result_set ابتدا نام‌های هر ستون در سطر اول (با فاصله) از هم چاپ شود و سپس ریکوردهای جدول در سطرهای بعدی(هر کدام در یک سطر) چاپ شود.

- تمرین به صورت داوری در کوئرا با تست کیس تصحیح می شود پس در نمایش خروجی های تعریف شده دقت کنید و دقیقا مطابق حالت خواسته شده خروجی را چاپ کنید. (مورد قبلی)
- تضمین می شود که کوئری های تو در تو به عنوان ورودی داده نمی شود.
- تضمین می شود که هر دستور با کاراکتر سمیکولن خاتمه پیدا کند.
- تضمین می شود که هر دستور جداگانه در یک خط ورودی داده شود.
- زبان SQL به کوچک و بزرگ بودن حروف کلمات رزرو شده (SELECT, UPDATE, ...) حساس نیست با این حال تضمین می شود که تمام کلمات رزور شده باحروف بزرگ ورودی داده شوند.
- تضمین می شود کلمات رزور شده مانند SELECT, UPDATE, DELETE, ... به عنوان اسامی جدول ها و ستون ها ورودی داده نشود.
- برای یادگیری و آشنایی بیشتر با SQL می توانید از این لینک استفاده کنید. (برای علاقه مندان)

<https://www.roxo.ir/sql-language-introduction>

موفق باشید.

دانشگاه شما دانشگاه نیست!

- محدودیت زمان: ۵۰ ثانیه
- محدودیت حافظه: ۵۰ مگابایت

بیگ مسعود بعد از اینکه تمرین سری یک را انجام داد، تصمیم گرفت تا انتقام مشکلات انتخاب واحد خود را از مسئولین آموزش بگیرد برای همین با تکنیک‌های که بلد بود تلاش کرد تا به سامانه‌ی آموزش نفوذ کند اما سامانه‌ی آموزش با پیام "دانشگاه شما دانشگاه نیست" تحمل این نفوذ را نکرد و به طور کامل از دسترس خارج شد. مسئولین آموزش پس از اطلاع از این جریان تصمیم گرفتند به بیگ مسعود فرصت جبران بدهند (چرا که اگر او را اخراج می‌کردند میانگین معدل دانشگاه افت میکرد!!). بیگ مسعود پس از تفکرات فراوان برای این سامانه، توانست UMLی طراحی کند اما از آنجایی که این حجم از تفکرات باعث شد او از درس‌های دیگرش عقب بیفتد از شما به عنوان دوستش کمک خواسته تا این UML را به کد تبدیل کنید و خودش درس بتواند درس‌های دیگرش را بخواند. او در عوض به شما قول می‌دهد در صورت قبول این زحمت، نمره یک سوال از تمرین ۲ درس OOP را به شکل مخفیانه برای شما وارد سامانه کند.

Course
<u>- courses : ArrayList<Course></u> - courseID : int - unit : int - capacity : int - numberOfRegisteredStudents : int - score : double
<u>+ addNewCourse(course : Course) : void</u> <u>+ getCourses() : ArrayList<Course></u> + getNumberOfRegisteredStudents() : int + setNumberOfRegisteredStudents(number : int) : void + incrementNumberOfRegisteredStudents() : void + decrementNumberOfRegisteredStudents() : void + getCourseID() : int + addCapacity(number : int) : void + getScore() : double + getUnit() : int + setScore(score : double) : void + getCapacity() : int <u>+ showCourse(courseID : int, string : String) : void</u>

Student
<ul style="list-style-type: none"> - <u>students : ArrayList<Student></u> - studentID : int - studentCourses : ArrayList<Course> - average : double
<ul style="list-style-type: none"> + getAverage() : double + <u>addNewStudent(student : Student) : void</u> + <u>registerCourse(studentID : int, courseID : int) : void</u> + addCourse(course : Course) : void + <u>deleteCourse(studentID : int, courseID : int) : void</u> + setMark(courseID : int, mark : double) : void + <u>setAverages() : void</u> + <u>getStudents() : ArrayList<Student></u> + getStudentID() : int + getStudentCourses() : ArrayList<Course> + <u>deleteStudent(studentID : int) : void</u> + <u>showAverage(studentID : int) : void</u> + <u>showRanks() : void</u> + <u>showRanks(courseID : int) : void</u>

Lecturer
<ul style="list-style-type: none"> - <u>lecturers : ArrayList<Lecturer></u> - lecturerID : int - courses : ArrayList<Course>
<ul style="list-style-type: none"> + <u>getLecturers() : ArrayList<Lecturer></u> + <u>addNewLecturer(lecturer : Lecturer) : void</u> + addCourse(courseID : int) : void + getCourses() : ArrayList<Course> + getLecturerID() : int + <u>addCapacity(lecturerID : int, courseID : int, number : int) : void</u> + <u>setMark(lecturerID : int, courseID : int, mark : double, studentID : int) : void</u>

نکته: تمامی کلاس‌ها باید مطابق با نمودار UML داده شده پیاده‌سازی شده شود. در صورتی که نیاز به متد یا کلاس کمکی که در UML ذکر نشده دارید، می‌توانید آن را پیاده سازی کنید اما چارچوب کلی کد شما باید

مبتنی بر UML داده شده باشد. در غیر این صورت بیگ مسعود اخراج شده و در نتیجه شما هم نمره‌ی سوال ۱ تمرین ۲ این درس را نمی‌گیرید!

ویژگی‌های سامانه‌ی آموزش:

addStudent <sID>

- دانشجویی با شناسه‌ی sID به سامانه اضافه می‌شود.

addLecturer <ltID> <cID1> <cID2>

- استاد با شناسه‌ی ltID به سامانه اضافه می‌شود. این استاد دروس cID1، cID2 و... را ارائه می‌دهد. همچنین ممکن است یک استاد هیچ درسی ارائه نکند یا فقط یک درس ارائه کند.

W <cID> <sID>

- دانشجویی با شماره‌ی sID درس cID را حذف می‌کند.

<sID> register <cID>

- دانشجویی با شماره‌ی sID در درس cID ثبت‌نام می‌کند.

<sID> register <cID1> <cID2> ...

- دانشجویی با شماره‌ی sID در درس‌های cID1 و cID2 و... ثبت‌نام می‌کند.

<ltID> capacitate <cID> <n>

- استاد با شناسه‌ی ltID ظرفیت درس cID را به اندازه‌ی n افزایش می‌دهد.

<ltID> mark <cID> <sID1> <mark1> <sID2> <mark2> ...

- استاد با شناسه‌ی ltID که درس cID را ارائه می‌دهد، برای دانشجوی sID1 نمره‌ی mark1، برای دانشجوی sID2 نمره‌س mark2 و... را ثبت می‌کند. تضمین می‌شود نمره‌ی تمام دانشجویان درس cID وارد می‌شود.

<ltID> mark <cID> <mark> -all

- استاد با شناسه‌ی ltID برای دانشجویان درس cID نمره‌ی mark را رد می‌کند.

start semester

- شروع ترم جاری

end semester

- پایان ترم جاری

end registration

- پایان زمان ثبت‌نلم

addCourse <sID> <unit>

- درس با شناسه‌ی cID که unit واحد دارد به سامانه اضافه می‌شود.

دستورات show:

showCourse <cID> <students|lecturer|capacity|average>

- برای درس cID ویژگی خواسته شده نمایش داده می‌شود. (اگر students بود لیست تمامی دانشجویان ثبت‌نامی به ترتیب شماره‌ی دانشجویی، اگر lecturer بود، شناسه‌س استاد ارائه‌دهنده‌ی درس، اگر capacity بود، ظرفیت درس و اگر average بود، معدل درس را چاپ کند) در صورت معتبر

نبودن شماره‌ی درس یا حذف شدن آن (به حد نصاب نرسیدن)، پیام shoma daneshjoo nistid را چاپ کنید.

showRanks <SID>

- شناسه‌ی سه دانشجویی که بالاترین نمرات درس را کسب کرده‌اند چاپ می‌کند. (در صورت برابر بودن نمره‌ی چند دانشجو، سه نفر اول به ترتیب شماره‌ی دانشجویی چاپ شوند)

shawAverage <SID>

- معدل یک دانشجو را چاپ می‌کند. در صورت وجود نداشتن دانشجوی با شناسه‌ی SID، پیام shoma daneshjoo nistid را چاپ کنید.

showRanks -all

- نمایش دانشجویان با رتبه‌ی 1 تا 3 از نظر معدل. (در صورت برابر بودن، به ترتیب شماره‌ی دانشجویی چاپ شوند)

نکات دیگر:

- هر درس به طور پیش فرض دارای 15 نفر ظرفیت می‌باشد و باید دارای حداقل 3 نفر برای تشکیل باشد.
- هر دانشجو باید حداقل 12 واحد در یک ترم داشته باشد.
- هر درس تنها توسط یک استاد ارائه می‌شود.
- هر دانشجو تنها 3 واحد را می‌تواند W کند به این صورت که تا زمانی که واحدهایی که درخواست W آن‌ها را داده از 3 واحد بیشتر نیستند، آن‌ها حذف می‌شوند و سپس دیگر حذف نمی‌شوند. مثلاً اگر دستور حذف یک درس 4 واحدی را وارد کند، نه تنها درس حذف نمی‌شود بلکه هیچ واحد دیگری نیز نمی‌تواند حذف کند.
- شناسه اساتید و دانشجویان 5 رقمی هستند.
- با شروع ترم، ثبت‌نام نیز شروع می‌شود.

- پس از پایان زمان ثبت‌نام، ابتدا تمام دروسی که به حد نصاب نرسیده‌اند حذف شده و دانشجویان دیگر آن درس‌ها را نخواهند داشت. سپس تعداد واحد تمام دانشجویان چک می‌شود. پس از این مرحله دیگر درس‌ها چک نشده و ترم رسماً آغاز می‌گردد.
- در صورتی که یک دانشجو از کف مجاز واحد کمتر داشته باشد، ترم وی حذف می‌شود.
- دستورات اضافه شدن دانشجو و استاد قبل شروع ترم وارد می‌شوند.
- پس از اتمام ترم دستورات show وارد می‌شوند و برنامه با دستور endShow پایان می‌پذیرد.
- درصوت وارد شدن دستورات بی‌ربط چیزی چاپ نمی‌شود.
- تمامی محاسبات اعشاری با دقت یک رقم اعشار محاسبه شود.

ورودی نمونه

```

addCourse 40101 3
addCourse 40102 4
addCourse 40103 4
addCourse 40104 3
addCourse 25001 4
addCourse 25002 3
addCourse 25003 4
addCourse 65001 3
addCourse 65002 3
addCourse 65003 4
addStudent 95101
addStudent 95102
addStudent 95103
addStudent 95103
addStudent 95104
addStudent 94101
addStudent 94102
addStudent 94103
addLecturer 10001 40101 40103
addLecturer 10002 40102 40104
addLecturer 10003 25001 25002
addLecturer 10004 65001 65002 65003
start semester
95101 register 40101 40102 40103 25001 25002

```



```

95102 register 40101 40102 40103 25001 25002
95103 register 40101 40102 40103 25001 25002
95104 register 40101 40102 40103 25002 65002
94101 register 40102 40103 25001 25002 65001
94102 register 40102 40103 25001 25002 65001
94103 register 40102 40103 25001 65001
end registration
10001 mark 40101 95101 12 95102 18 95103 18.5 95104 9.9
10002 mark 40102 95101 19.8 95102 20 95103 16.5 95104 19
94101 17 94102 10 94103 15.9
10001 mark 40103 95101 17.4 95102 19 95103 15.7 95104 11.2
94101 19 94102 17.8 94103 20
10003 mark 25001 95101 9 95102 10 95103 10 94101 10 94102
19 94103 12
10003 mark 25002 95101 20 95102 14 95103 18 95104 9.5
94101 20 94102 18 94103 19.6
10004 mark 65001 94101 18 94102 10 94103 15
W 40101 95101
end semester
showAverage 95103
showCourse 25001 average
showRanks 65001
endShow

```

خروجی نمونه

```

15.5
11.7
94101 94103 94102

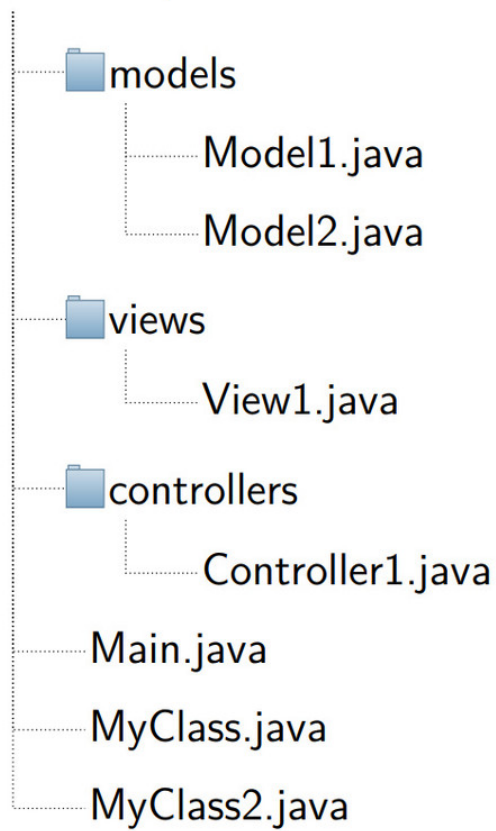
```

توجه داشته باشید که در این سوال محدودیت زمانی دارید و از سریع ترین روش‌ها استفاده کنید.

این سوال توسط سامانه کوئرا داوری و نمره‌دهی خواهد شد. توجه داشته باشید به هیچ‌وجه تصحیح این سوال به صورت دستی انجام نخواهد شد. لطفاً به موارد زیر توجه کنید:

- با توجه به اینکه SDK کوئرا نسخه 1.8 می باشد لطفا از نسخه استفاده کنید.
- در کل پروژه فقط یک شی Scanner داشته باشید.
- جهت آپلود کدهای خود به یکی از دو روش زیر اقدام کنید:
 - برای آپلود همه ی کلاس ها رو کپی کنید زیر یک کلاس (توی یک فایل java) فقط توجه کنید که توی یک فایل فقط می تونید یک کلاس public داشته باشید. بعد از کپی کردن کلاس ها توی یک فایل پیشوند پابلیک کلاس هایی که کپی کردید رو بردارید تا فقط یک کلاس پابلیک باقی بماند. (حتما کلاسی که حاوی تابع `public static void main(string[] args)` بصورت public باشد)
 - یک نمونه مثال ساده برای اینکه چقدر کلاس توی یک فایل داشته باشید.
 - شما باید تمامی فایل های کلاس هایی که با فرمت java نوشته اید را در قالب فشرده با فرمت zip درآورده و در سامانه کنید. توجه کنید که حتما فایل حاوی کلاسی که تابع `public static void main(string[] args)` در آن قرار دارد و اجرای برنامه از آنجا شروع می شود، باید بدون قرار گرفتن در هیچ پوشه ای درون فایل zip قرار بگیرد و نام آن هم Main.java باشد. امکان پوشه بندی و ایجاد package های مختلف برای سایر فایل ها امکان پذیر است. به عنوان نمونه می توانید به ساختار قابل قبولی و درستی که در زیر رسم شده است، توجه کنید. توجه کنید که پوشه src که فایل ها در آن قرار دارد، نباید در فایل zip باشد.

answer.zip



- ورودی و خروجی شما باید عیناً شبیه به نمونه‌های ورودی و خروجی باشد؛ لذا عبارت‌هایی همچون Enter your number را قبل از گرفتن ورودی نباید چاپ کنید.

آرایه پویا

توجه کنید که جنریک در فایل آورده شده آنچنان مهم نیست و فقط شما از آن استفاده کنید و بعدا این مبحث توسط اساتید تدریس خواهد شد.

قسمت اول)

ابتدا فایل CrudOperation.java را از کانال دانلود کنید.

پارامتر E که داخل این کلاس وجود دارد نشانه این است که وقتی ما این کلاس را در کلاس Main.java خود تعریف می کنیم . به این صورت برای مثال وارد میکنیم :

```
public class MainCrud {  
  
    public static void main(String[] args) {  
  
        CrudOperation<String> crudOperation = new CrudOperation<>(new String[]{});  
    }  
}
```

گرچه با این مورد برخورد کرده اید نمونه بارز آن استفاده از ArrayList بوده است.

هدف از این تمرین پیاده سازی عملیات CRUD به صورت ساده می باشد. CRUD مخفف چهار کلمه ی CREATE , READ , UPDATE , DELET می باشد.

خود کلاس CrudOperation.java شامل یک سازنده دارد که ورودی آن آرایه از تایپ E می باشد که ما هنگام ساختن نمونه یا instance از این کلاس آن تایپ را همانند تصویر بالا تعیین می کنیم. و همچنین این کلاس شامل متدهای زیر می باشد که به اختصار کاری که هر کدام باید بکنند را توضیح می دهیم :

- متد read یک پارامتر به عنوان index می گیرد و مقداری از تایپ E بر میگرداند که بستگی دارد ابتدا شما آن تایپ را چی تعریف کرده باشید.
- متد insert که یک پارامتر از نوع E می گیرد و المان را به فیلد data اضافه می کند.
- متد insert(E element , int index) : در اینجا ما میخواهیم که المان را در خانه ای خاصی از آرایه ذخیره کنیم و اگر index داده شده بزرگتر از طول آرایه بود پیغام مناسبی به کاربر نمایش داده شود و index جدیدی گرفته شود.
- متد update : این متد برای این است که مقدار یکی از خانه های آرایه را تغییر دهد و در نهایت آن آرایه را بر میگرداند (data) .
- متد delete(int index) : این متد index آرایه را میگیرد و آن المانی که در این index آرایه قرار گرفته است را حذف کند.
- متد delete(E element) : این متد المان آرایه را گرفته و آن المان را از آرایه حذف می کند.

توضیحی راجب تست کردن متد ها :

در کلاس Main خود آرایه ای از تایپ مورد نظر خودتون تعریف کنید و یک

سایز پیش فرض به آن بدهید و طبق طول آرایه خود ورودی ها رو از کاربر

دریافت کنید.

و این آرایه را به سازنده کلاس CrudOperation.java پاس دهید. (طبق عکس

بالا به جای new String[]{} آن آرایه خود را پاس دهید و همچنین باید Type

CrudOperation<Type> از جنس آرایه ای باشد که شما همان ابتدا تعریف کرده

اید و مقادیرش را از کاربر گرفته اید.

حال باید منویی به کاربر نمایش دهید که شامل این موارد باشد :

Create .1

Read .2

Update .3

Delete .4

وقتی کاربر هر کدام از این گزینه ها را زد پیغامی چاپ شود که ایندکس یا المان مورد نظر

را بگیرد و همچنین گزینه Back داشته باشد که به منوی اصلی نیز باز گردد.

برای مثال وقتی کاربر گزینه 2 را می زند پیغامی چاپ که شود که بگوید ایندکس مورد نظر را وارد کنید و وقتی کاربر ایندکسی را وارد کرد که داخل آرایه بود مقدار آن چاپ شود و اگر این ایندکس داخل آرایه نبود یا منفی بود پیغام مناسبی چاپ شود و دوباره به کاربر همان منوی قبلی نمایش داده شود (منظور این است که همان منویی نمایش داده شود که از کاربر ایندکس برای Read میخواهد و همچنین گزینه Back هم داشته باشد).

برای گزینه های دیگه هم همین منطق نیز پیاده سازی شود.

قسمت دوم)

در این قسمت می خواهیم از متد هایی که در کلاس CrudOperation.java درست کرده ایم استفاده کنیم. متدی داخل این کلاس بنویسید که پارامتری از آرایه بگیرد و زوج های مرتب نامرتب آن را حذف کنید. زوج مرتب نامناسب دو عنصر در همسایگی یک دیگر هستند که عنصر سمت چپ زوج مرتب بزرگتر از عنصر سمت راست آن باشد. هر عضو سمت چپ زوج مرتب، دارای اندیس زوج در آرایه می باشد و هر عضو سمت راست زوج مرتب دارای اندیس فرد در آرایه می باشد. برای راحتی می توانید ابتدا از ArrayList استفاده کنید و آن را به آرایه تبدیل کنید همانند تصویر زیر :

```

ArrayList<Integer> arrayList = new ArrayList<>();

arrayList.add(1);
arrayList.add(23);
arrayList.add(34);

CrudOperation<Integer> integerCrudOperation =
    new CrudOperation<Integer>(arrayList.toArray(new Integer[0]));

```

وقتی این کار را انجام دادید آن متد داخل کلاس CrudOperation.java را کال کنید و در نهایت زوج مرتب را چاپ کنید.

برای مثال :

{3 , 7 , 9 , 2 , 5 , 5 , 8 , 5 , 6 , 3 , 4 , 7 , 3 , 1 }

که ما می توانیم آن را به صورت زوج مرتب های زیر در نظر بگیریم :

{3,7} , { 9 , 2 } , { 5 , 5 } , { 8 , 5 } , { 6 , 3 } , {4,7} , {3 , 1}

زوجهای :

{3 , 1} , {6 , 3} , {8 , 5} , {9 , 2}

عنصر اول از عنصر دوم بزرگتر هست، پس این زوج ها مناسب نیستند و کاندید حذف

شدن هستند؛ فلذا، پس از فراخوانی متد مذکور ، آرایه به صورت زیر باید تغییر کند :

{3,7,5,5,4,7}

اگر لیستی که ابتدا آن را به آرایه تبدیل می کنیم طولش فرد باشد ، آخرین عنصر را به

عنوان یک زوج مرتب در نظر نمی گیریم و خودش هم به عنوان یک عنصری که مناسب

نیست، حذف می شود. بنابراین باید توسط متد شما نیز حذف گردد. اگر لیست خالی به متد

ارسال شود، این لیست باید در پایان فراخوانی متد گفته شده، همچنان خالی باشد.

تذکر: مجاز به استفاده از سایر ساختمان های داده جز همان ابتدای کار (تبدیل

کردن لیست به آرایه) نیستید و فقط باید از آرایه ها استفاده کنید. همچنین برای

این قسمت لازم است کلاسی جدا از قسمت اول خود داشته باشید. برای مثال اگر

کلاس اصلی خود برای قسمت اول را CrudOperationMain.java تعریف

کرده اید ، برای قسمت دوم سوال یک کلاس مجزا درست کنید.

همچنین این تمرین به صورت دستی تصحیح خواهد شد بنابراین کلاس های خود

را زیپ کرده و در سایت آپلود کنید. (دقت کنید که فقط کلاس ها را زیپ کنید نه

پوشه src را).

متدی که شما باید آن را پیاده سازی کنید برای این قسمت این متد زیر می باشد :

```
/*  
 * Eliminate inappropriate pairs method  
 * */  
  
public E[] eliminateInappropriatePairs(E[] elements) {  
    /*  
     * implement this method  
     * */  
  
    return elements;  
}
```

و همچنین از متد `getData` برای بازگرداندن `data` در قسمت اول تمرین هر

جایی که نیاز شد استفاده کنید.

همچنین برای اطلاعات بیشتر راجب E این عبارت را سرچ کنید :

Generics in Java.

عیدتون مبارک 😊 موفق و تندرست باشید.

تقلب یاب

همانطور که می دانید از وقتی که کرونا وارد دنیا شده، انواع شیوه های تقلب در همه ی سطح ها گرفته از امتحانات مدرسه، آزمون های آزمایشی کنکور، امتحان دانشجو ها تا سطوح بالاتر به وجود آمده است. حالا در این سوال ما می خواهیم ابزاری طراحی کنیم که با کمک اطلاعات موجود و تحلیل برخی داده ها شامل افراد، گروه های تقلب و امتحانات برگزار شده، افراد خطا کار را شناسایی کنیم. اول از همه سامانه ی ما در صفحه ی اصلی قرار دارد که در این صفحه می توان عملیات های login یا signup را انجام داد دستور signup به شکل زیر وارد میشود:

```
register [username] [password] [teacher|student]
```

همانطور که مشخص است، یک کاربر با نام کاربری و پسورد مشخص شده را ایجاد می کند. نام کاربری و پسورد باید فقط شامل حروف الفبای انگلیسی، اعداد و کاراکتر آندرلاین باشند. خطاهای مربوط به این دستور به این ترتیب چک می شوند و هر خطا که رخ بدهد، پیغام مربوط به همان خطا چاپ شده و سایر خطاها بررسی نمی شوند. اگر هم هیچ خطایی رخ نداد و عملیات موفقیت آمیز بود پیغام زیر چاپ می شود

```
register successful
```

خطاها: اگر نام کاربری شامل کاراکترهایی به جز کاراکتر های ذکر شده بود، پیغام:

```
username format in invalid
```

اگر پسورد شامل کاراکتر هایی به جز کاراکتر های ذکر شده بود، پیغام:

```
password format is invalid
```

اگر کاربری با username گفته شده از قبل وجود داشت پیغام:

a user exists with this username

دستور بعدی ، دستور

login [username] [password]

برای لاگین کردن به حساب کاربری مشخص با نام کاربری و پسورد داده شده استفاده می شود. نام کاربری و پسورد باید فقط شامل حروف الفبای انگلیسی، اعداد و کاراکتر آندرلاین باشد. خطاهای مربوط به این دستور به این ترتیب چک می شوند و هر خطا که رخ داده بود، پیغام همان خطا چاپ شده و سایر خطاها بررسی نمی شوند. اگر هم هیچ خطایی رخ نداد و وضعیت موفقیت آمیز بود، پیغام

login successful

چاپ خواهد شد و پس از آن کاربر به طور خودکار به صفحه ی اصلی خود وارد می شود

خطاها:

اگر پسورد شامل کاراکتر هایی به جز کاراکترهای ذکر شده بود، پیغام:

username format is invalid

اگر نام کاربری شامل کاراکتر هایی به جز کاراکترهای ذکر شده بود، پیغام:

password format is invalid

اگر کاربری با username وارد شده وجود نداشت پیغام:

no user exists with this username

گر پسورد غلط بود پیغام:

incorrect password

چاپ خواهد شد

و بعد از ورود به منوی اصلی می توان داده هایی را وارد کرد و خواسته ها را نسبت به آن ها صادر کرد. با دستور

logout

هم میتوان از حساب کاربری خارج شد و خروجی این دستور هم جمله ی

logout successssful

می باشد

دقت داشته باشید خروج از برنامه و در واقع پایان یافتن برنامه تنها از منوی ثبت نام و ورد امکان پذیر است و با دستور

exit

این اتفاق می افتد و برنامه تمام می شود . برای حل مسئله برخی فرض هارا در نظر می گیریم که شامل موارد زیر هست.

1-هر امتحان با آیدی ای که از یک شروع می شود، مشخص می شود.

2-هر فرد هم با آیدی که از یک شروع میشود، مشخص میشود.

3-هر فرد میتواند در هیچ، یک یا چند گروه تقلب باشد.

4- اگر فردی به عنوان مثال در دو گروه تقلب باشد، وقتی تقلب کند، کل اعضای دو گروهی که آن شخص در آن ها عضو است هم پایشان گیر است و در حالت مشکوک قرار می گیرند.

5- تاریخ آزمون و تاریخ زمان که فردی تقلب می کند هم به فرمت yyyy/mm/dd داده میشود.

6- این موضوع را حتما می دانید ولی به عنوان تذکر ذکر می کنیم که تعداد روز ماه های میلادی به ترتیب از ژانویه تا دسامبر برابر با ۳۱، ۲۸ یا ۲۹، ۳۱، ۳۰، ۳۱، ۳۰، ۳۱، ۳۱، ۳۰، ۳۱، ۳۰، ۳۱ و ۳۱ است.

7- خیلی از اوقات پیش می آید که امتحان شخصی پشت سر هم باشد یعنی بدون فاصله بعد از یک امتحان، به محل برگزاری امتحان بعدی می رود و یا حتی در حین برگزاری یک امتحان، اجازه خواسته و به دستشویی یا بالای سطل آشغال برای تراشیدن مدادش یا انداختن آشغالی درون سطل آشغال می رود. پس ما یک سری جابه جایی در حین امتحان داریم که یا درون امتحانی است و یا برون امتحانی. در این موارد این انتقال به صورت ورودی داده میشود و آیدی امتحان مبدا و مقصد مشخص می شود و در ادامه گفته می شود که خواسته سوال از شما در برخورد با این انتقالات به چه شکل باشد

8- در مورد قبل اگر جا به جایی درون امتحانی باشد، آیدی مبدا و مقصد یکسان می باشد.

9- در صورتی که شخصی در روز x ام تقلبی انجام دهد، تا روز $x+7$ ام در حالت مشکوک به تقلب باقی ما ماند و در ادامه گفته می شود که در صورت انجام چه حرکتی، تقلب او لو می رود.

10- حالد صورتی تقلب شخص لومی رود که در بازه ی ۸ روزه ای که مشکوک به تقلب هست، جا به جایی بین امتحان ها داشته باشد. یعنی اگر در یک امتحان تقلبی انجام دهد و در بازه ی ۸ روزه در امتحان دیگری شرکت کند، تقلب او بر همگان آشکار میگردد.

11- دقت کنید جا به جایی درون امتحانی تخلف محسوب نمی شود و فرد مشکوک در این حالت لو نمی رود.

ورودی

دقت کنید دستورات زیر فقط زمانی قابل اجرا هستند که شخص دارای role دانشجو باشد که در زمان ثبت نام برای او قرار داده شده است و اگر استادی بخواهد این دستورات را وارد کند باید برای او invalid command برگردانید

همه ی ورودی های زیر بعد از لاگین معتبر هستند و منظور از خط اول یا n ام در این ورودی ها، بعد از دستورات مرتبط با لاگین است.

- در خط اول ورودی n می آید که همان افراد مورد بررسی است یعنی افراد از آیدی ۱ تا n

n

- در خط دوم m می آید که تعداد گروه های تقلب موجود هست

m

- بعد در m خط بعد اعضای گروه تقلب i ام با یک فاصله بین هم داده میشود

- بعد c داده می شود که تعداد امتحانات است و در واقع این یعنی امتحانات ما آیدی بین ۱ تا c دارند

c

- بعد p وارد می شود که در واقع تعداد جابه جایی های انجام شده در دانشگاه است

p

- در p خط بعد اول آیدی فرد جا به جا شده می آید و بعد آیدی امتحان مبدا و بعد آیدی امتحان مقصد و در نهایت هم تاریخ این جا به جایی وارد م شود به فرمت yyyy/mm/dd

personId sourceExam destinationExam date

- سپس t وارد می شود که تعداد تلاش ها برای تقلب است

t

• در خط بعد، در هر خط اول آیدی شخصی که تلاش به تقلب کرده وارد می شود و بعد نتیجه ی تقلب وی داده میشود که در دو حالت positive و negative است که اگر positive بیاید یعنی وی موفق به تقلب شده است و در غیر این صورت یعنی موفق به تقلب نشده است. و بعد آیدی امتحانی که آن شخص در آن تلاش به تقلب کرده است داده می شود و در نهایت هم تاریخ روزی که آن شخص خواسته تقلب کند به همان فرمت yyyy/mm/dd داده میشود

personId cheatResult examId date

دقت کنید دستوراتی که چند بخشی هستند که شامل مشخص کردن اعضای هر گروه تقلب، جا به جایی های صورت گرفته توسط افراد و تلاش ها برای تقلب می شوند، تضمین می شود که معتبر هستند و نیازی نیست شما از صحت ورودی ها اطمینان حاصل کنید ولی در هر دستور دیگری که ذکر نشد، شما

باید از صحت فرمت ورودی اطمینان حاصل کنید و اگر معتبر نبود، پیام invalid command چاپ کنید. بعد از اینکه این دستورات وارد شد، یک round برای دانشجوی لاگین شده تمام می شود و در این قسمت، ۲ کار می تواند انجام دهد، یا round بعدی را با وارد کردن nextround شروع کند و یا اینکه با زدن دستور logout، از پنل خود خارج شود. در صورتی که round بعدی را شروع کند، اطلاعاتی که وارد می کند، در ادامه ی اطلاعات قبلی م ی آید، به عنوان مثال اگر در round اول در خط اول عدد ۶ را وارد کرده باشد، یعنی ۶ دانشجو از آیدی ۱ تا ۶ داریم و بعد ادامه اطلاعات را وارد می کند و در round بعدی اگر عدد مثلا ۳ را وارد کند یعنی دانشجویان جدیدی از آیدی ۷ تا ۹ داریم و در اطلاعات بعدی مثلا اعضای هر گروه، همینطور برای امتحان ها هم به همین شکل که برای دانشجویان بوده عمل کنید. به این نکته هم دقت کنید که اگر یک دانشجو logout کند و بعد دانشجوی دیگری login کند و اطلاعاتی را وارد کند، آن اطلاعات هم در ادامه ی اطلاعات قبلی می آید

درخواست ها

درخواست ها دقت کنید این درخواست ها فقط زمانی قابل وارد شدن هستند که شخصی که لاگین کرده دارای role استاد باشد و اگر role آن دانشجو باشد، در هنگام وارد کردن این درخواست ها با invalid command مواجه می شود

• خواسته ها وارد می شود که پایین تر توضیحات آن ها را می توانید ملاحظه کنید و این وارد شدن دستورات تا زمان وارد شدن دستور clear data و یا end و یا logout ادامه پیدا می کند. دقت کنید در صورت وارد شدن دستور clear data کل اطلاعات به شکل کامل پاک می یاشود و همه چیز از نو شروع می شود و در صورت وارد شد دستور end بدون اینکه اطلاعاتی پاک شود، صرفا گرفتن درخواست پایان می یابد و با وارد شدن دستور logout از حساب کاربری استاد خارج می یاشویم. اگر logout وارد شود که به منوی ثبت نام و ورود بر می گردیم ولی اگر clear data یا end وارد شد در حالت انتظار می مانیم که یا می توان باز logout کرد و از حساب کاربری خارج شد و یا می توان یکی از دستور های wrist-up را وارد کرد و به چرخه ی گرفتن دستور بر گردیم

۱. درخواست wrist-up1 :

با این درخواست از شما می خواهیم که افراد با بیشترین تخلف ثبت شده را به ترتیب نزولی برای ما برگردانید

wrist-up1 count

که در واقع count تعداد افرادی است که در خروجی باید نمایش دهید و در حالتی که تعداد تخلف ثبت شده دو نفر با هم برابر بود، فرد با آیدی کوچک تر، بالاتر قرار می گیرد. ما این دستور را به شکل زیر می دهیم و شما می خواهید و آن به این شکل است که تاریخ شروع و پایان را هم در دستور به شکل زیر می دهیم و شما باید افراد با بیشترین تخلف ثبت شده در آن بازه زمانی را به ترتیب نزولی برای ما پیدا کنید. در این بخش هم فرمت تاریخ به شکل yyyy/mm/dd داده می شود

wrist-up1 count startDate endDate

2. درخواست wrist-up2 : در این درخواست از شما Dm خواهیم که افراد با بیشترین تقلب positive شده را به ما نشان دهید. دقت کنید در این حالت تقلب positive شده ی خود فرد منظور است و اگر یکی از هم گروهی هایش لو رفته باشد و این شخص را هم لو داده باشد، شما در شمارش این بخش نباید آن را وارد کنید و فقط تقلب positive شده ی خود آن شخص را در نظر بگیرید. و باز هم به شکل نزولی نمایش دهید و در صورت برابر شدن تعداد تقلب های positive شده ی دو نفر، فرد با آیدی کوچک تر، بالاتر قرار بگیرد

wrist-up2 count

که در اینجا هم count تعداد افرادی هست که شما به ما باید نشان بدهید. این دستور هم به شکلی دیگر می تواند وارد شود و آن هم به این شکل هست که تاریخ شروع و پایان بازه ی مورد نظر ما وارد می شود و شما افراد با بیشترین تقلب positive شده در آن بازه زمانی را برای ما باید برگردانید. در اینجا هم مانند بقیه بخش ها تاریخ با فرمت yyyy/mm/dd وارد میشود

wrist-up2 count startDate endDate

۳. درخواست wrist-up3 : در این درخواست از شما می خواهیم امتحان هایی با بیشترین تقلب positive شده در آن را برای ما نشان دهید. در اینجا هم دقت کنید که گفته شد امتحاناتی که بیشترین تقلب positive شده در آن رخ داد و این یعنی آن شخص خودش تقلبش positive شده باشد و نه اینکه توسط گروهش لو داده شده باشد. در اینجا هم آن امتحان ها را به ترتیب نزولی نمایش دهید

wrist-up3 count

که در اینجا هم count تعداد امتحاناتی است که شما باید در خروجی برای ما برگردانید و در صورت برابر بودن تعداد افرادی که تقلب آن ها positive شده در دو امتحان، امتحان با آیدی کوچک تر، بالاتر قرار می گیرد. این دستور هم مانند دستور های قبل به این نوع هم می توانند وارد شوند که ما تاریخ شروع و پایان بازه ی مورد نظر خود را بگوییم و شما امتحاناتی که در آن بازه بیشترین تقلب positive شده را داشتند را برگردانید. و اینجا هم تاریخ با فرمت yyyy/mm/dd به شما داده می شود

wrist-up3 count startDate endDate

۴. درخواست wrist-up4 : در این درخواست، آیدی یک امتحان را به شما می دهیم و روز هایی که بیشترین تقلب positive شده در آن رخ داده را به ما نشان می دهید و به ترتیب نزولی در خروجی برگردانید. در

این بخش شما تاریخ را باید به فرمت yyyy/mm/dd به ما بدهید

wrist-up4 examId count

که در اینجا هم count تعداد روزهایی است که شما در خروجی باید نمایش دهید و در صورتی که در دو روز تعداد تقلب های positive شده برابر بود، روز قبل تر در خروجی، بالاتر قرار می گیرد.

۵. درخواست wrist-up5 : در این درخواست از شما می خواهیم روز هایی از سال با بیشترین تعداد افراد مشکوک به تقلب را به شکل نزولی در خروج چاپ کنید. به این نکته ی خیلی مهم توجه کنید، همانطور که گفته شد در هر روزی که شخصی تقلب کند، خود آن روز و تا ۷ روز بعد از آن در حالت مشکوک به تقلب هست

wrist-up5 count

که باز هم count تعداد روز هایی است که در خروجی باید برگردانید و در صورت برابر بودن تعداد افراد مشکوک به تقلب در یک روز، روز قبل تر، بالاتر قرار می گیرد. و در اینجا هم فرمتی که ما تاریخ را از شما می خواهیم به شکل

yyyy/mm/dd

است

خروجی

برای هر درخواستی که داده میشود در ی خط شما اسم در خواست را چاپ می کنید یعنی یکی از مقادیر

wrist-up5 or wrist-up4 ،wrist-up3 ،wrist-up2 ،wrist-up1

و سپس نوع خروجی هر دستور متفاوت خواهد بود و به شکل زیر است:

- خروجی درخواست wrist-up1 شامل count خط است و در هر خط اول آیدی فردی که تقلب اون آشکار شده می آید و بعد تعداد تقلب های آشکار شده ی او می آید.
- خروجی درخواست wrist-up2 هم شامل count خط است که اول آیدی فردی که تقلب positive دارد می آید و سپس تعداد تقلب های positive آن شخص می آید.
- خروجی درخواست wrist-up3 هم شامل count خط است و در هر خط ابتدا آیدی امتحان و سپس تعداد تقلب های positive شده در آن امتحان می آید.
- خروجی درخواست wrist-up4 هم شامل count خط است که در هر خط ابتدا تاریخ روز به فرمت yyyy/mm/dd را نمایش می دهید و سپس تعداد تقلب های positive شده در آن روز می آید.
- خروجی درخواست wrist-up5 هم شامل count خط است و در هر خط ابتدا تاریخ روز به فرمت yyyy/mm/dd می آید و سپس تعداد افراد مشکوک در آن روز می آید

ورودی نمونه ۱

registre asqar 1234 student login asqar 1234

6

3

1 2

3 4

2 5

3

2

1 1 3 2021/01/20

1 3 1 2021/01/30

1 negative 1 2021/01/20

1 positive 2 2021/01/28

3 negative 1 2021/12/26

logout

register akbar 1234 teacher

login akbar 1234

wrist-up1 3 end

```
logout
exit
```

خروجی نمونه ۱

```
register successful
login successful
logout successful
register successful
login successful
wrist-up1
1 1
2 0
3 0
logout successful
```

توضیحاتی در مورد این نمونه در این نمونه فرد با id شماره یک در روز ۲۸ ام یک تلاش برای تقلب داشته است و همینطور در روز ۳۰ ام یک جا به جایی بین امتحانی داشته که در واقع از امتحان شماره ۳ به امتحان شماره یک رفته است. و چون در این مدت هنوز مشکوک بوده است، طبق توضیحات گفته شده، تقلب این شخص بر همه آشکار میشه و یک مورد تخلف برای او ثبت مش شود. دقت کنید که فرد شماره ۲ هم با فرد شماره یک در یک گروه بوده است و این شخص شماره یک در جا او را لو می دهد ولی چون شخص شماره ۲ هیچ جابه جایی ای نداشته در نتیجه با اینکه مشکوک بوده است ولی تقلب اون بر همه آشکار نمیشه و موردی براش ثبت نمیشه

ورودی نمونه ۲

```
register asqar 1234 student
login asqar 1234
6
3
1 2
3 4
2 5
3
```

```
2
1 1 3 2021/01/20
1 3 1 2021/01/30
1 negative 1 2021/01/20
1 positive 2 2021/01/28
3 negative 1 2021/02/04
2 positive 3 2021/02/20
5 positive 1 2021/01/28
logout
register akbar 1234 teacher
login akbar 1234
wrist-up2 4
wrist-up3 2
wrist-up4 2 1
wrist-up5 3
data clear
logout
exit
```

خروجی نمونه ۲

```
register successful
login successful
logout successful
login successful
wrist-up2
1 1
2 1
5 1
3 0
wrist-up3
1 1
2 1
wrist-up4
2021/01/28 1
wrist-up5
2021/01/28 3
2021/01/29 3
```

2021/01/30 3
logout successful

توضیحاتی در مورد این نمونه

در توضیح این نمونه هم می توان گفت که در درخواست wrist-up^۲ ما فقط افراد بیشترین تعداد تقلب positive رو می خواهیم که به وضوح، افراد با آیدی های یک و دو و پنج هر کدام ی کتقلب positive دارند و یغبه هیچی در نتیجه خروجی ای بخش به شکل

wrist-up2

1 1
2 1
5 1
3 0

می شود

در درخواست wrist-up^۳ همانطور که مشاهده می شود، در در امتحان ها با آیدی های یک و دو سه هر کدام یک تقلب positive وجود دارد و در نتیجه خروجی این بخش به شکل

wrist-up3

1 1
1 2

خواهد بود چون ما تنها ۲ امتحان را خواسته ایم.در درخواست wrist-up^۴ چون در امتحان با آیدی شماره ۲ تنها یک تقلب positive و آن هم در روز ۲۸ ام رخ داده است، در نتیجه خروجی به شکل زیر میشود

wrist-up4

2021/01/28 1

می شود

و در نهایت خروجی درخواست wrist-up5 را به شدت توجه کنید.

اول از همه فرد مشکوک، شخص با آیدی شماره یک است که در روز ۲۸ ام مشکوک شده است. حال دقت کنید که شخص شماره ی با شخص شماره ۲ در یک گروه تقلب بوده اند و در نتیجه این فرد هم مشکوک می شود به تقلب و در عین حال شخص شماره ۵ هم در همین روز ۲۸ ام مش وکبه تقلب شده است و چون هم گروهی تقلب او شخص شماره ۲ است، تاثیر خاصی ندارد چون شخص شماره ۲ قبل تر مشکوک شده بود. پس تا اینجا در بازه ی ۲۸ ام ژانویه تا پایان ۴ ام فوریه، در هر روز، ۳ نفر مشکوک بوده اند. حال در روز ۲۰ ام فوریه، شخص شماره ۲ با موفقیت تقلبی انجام داده است و چون هم با فرد شماره ۱ و هم با فرد شماره ۵ در گروه هست، آن ها هم مشکوک می شوند و در نتیجه در بازه ۲۰ ام فوریه تا ۲۷ ام فوریه باز هم سه شخص ۱ و ۲ و ۵ مشکوک هستند پس در مجموع ۱۶ روز وجود دارد که جمع مشکوک های آن برابر ۳ است و چون ما تنها ۳ روز را خواسته ایم در خروجی به ما نمایش بدهید و در صورت برابر بودن تعداد افراد دربرخی روز، روز قبل تر، بالاتر باید می آمد، خروجی در نهایت به شکل

wrist-up5

2021/01/28 3

2021/01/29 3

2021/01/30 3

می شود