



دانشکده مهندسی برق
دانشگاه صنعتی شریف

مزده‌ای دل که مسیحانفسی می‌آید
که زانفاس خوش بوی کسی می‌آید
از گم هجرم کن ناله و فریاد که دوش
زده ام فالی و فریادرسی می‌آید



درس برنامه سازی شے گرا

نیمسال دوم ۱۴۰۲-۱۴۰۱
دکتر وثوقی وحدت، دکتر هاشمی

بارم

تمرین دوم

- | | |
|-------|-----------------------------|
| ۱/۱۴۰ | ۱. روز جهانی هاتریس |
| ۵/۱۴۰ | ۲. خونه خاله کدوم و ره؟ |
| ۵/۱۳۰ | ۳. پول نقد نداریم |
| ۵/۱۴۰ | ۴. تکنولوژی در سرزمین عجایب |

مهلت تحويل:

دوشنبه ۷ فروردین ۱۴۰۲

توجه:

این تمرین در مجموع دارای ۵/۱ نمره می‌باشد؛ انمراه بارم اصلی تمرین دوم، ۵/۲ نمره امتیازی تمرین دوم و ۵/۵ نمره دیگر می‌تواند امتیازی تمرین اول را جبران کند!

روز جهانی ماتریس

ملکه سفید در روز جهانی ماتریس، مسابقه ای در سرزمین عجایب برگزار می کند. (چیه؟ ماتریس ها نمی تونن روز جهانی داشته باشند؟!) مسابقه به این صورتی که برنامه نویس های حرفه ای دور هم جمع می شن و کلاسی می سازن که به کمک اون میشه ماتریس ها رو ساخت و ذخیره کرد، و اعمال ساده ای مثل جمع، ضرب، دترمینان و ... رو روی اونا انجام داد. جایزه مسابقه هم همون طور که انتظارش رو داشتید، ۴۰ نمره ناقابل برای کسانیه که کلاس رو به درستی بسازن!

در این سوال مجاز به استفاده از کتاب خانه های آماده Java برای پیاده سازی ماتریس نیستید!

نمره کامل این سوال ۳۵ است. در صورتی که وارون ماتریس را به درستی پیاده سازی کنید، ۱۰ نمره امتیازی دریافت می کنید.

ورودی

در این سوال ۱۶ دستور ورودی داریم که در ادامه به توضیح آنها پرداخته شده است. همچنین خروجی هر دستور در بخش خروجی قابل مشاهده است.

۱) خروج از برنامه ▼

تا زمانی که دستور end وارد نشود برنامه به ورودی گرفتن ادامه مبدهد.

-end

۲) افزودن ماتریس ▼

با وارد کردن دستور زیر، کاربر می تواند یک ماتریس جدید به تعداد سطر و ستون numCols و numRows بسازد.

-addMatrix <matName> <numRows> <numCols>

همچنین تضمین می شود که matName تنها متشکل از یک حرف الفباست. دقت کنید که موقع تشکیل ماتریس، قبل از مقداردهی باید به طور پیش فرض مقدار تمامی عناصر ۰ باشد. تضمین می شود که مقادیر numCols و numRows معتبر هستند.

۳) مقداردهی ▼

در خط اول، کاربر عبارت زیر را وارد کرده

```
-set <matName>
```

و در ادامه آن مقادیر مدنظر برای ماتریس وارد می شوند. مثلا برای یک ماتریس ۳ در ۴

```
0 12 3 4.5
3 32.1 4 103
94 7 1.2 0
```

تضمین می شود که در هر سطر، تعداد ستون ها با هم برابر است. همچنین تضمین می شود که ابعاد ورودی های داده شده با ابعاد ذکر شده هنگام ساخت ماتریس یکی هستند.

۴) جمع ▼

با دستور زیر می توان ۲ ماتریس را با هم جمع کرد و در ماتریس result ذخیره کرد.

```
-add <mat1> <mat2> <result>
```

نکته: ماتریس result از قبل وجود نداشته و طی همین عملیات ایجاد می شود.

۵) تفریق ▼

با دستور زیر می توان ماتریس دوم را از ماتریس اول کم کرد. (mat1 - mat2)

```
-subtract <mat1> <mat2> <result>
```

مانند مورد ۴، حاصل در ماتریس جدید `result` ذخیره می شود.

۶) ضرب ▼

با این دستور می توان ماتریس اول را در ماتریس دوم ضرب کرد.

```
-multiply <mat1> <mat2> <result>
```

۷) جابجایی سطر ▼

با این دستور می توان جای دو سطر ماتریس مورد نظر را عوض کرد.

```
-swapRows <mat> <row1> <row2>
```

تضمین می شود که مقادیر `row1` و `row2` مثبت هستند.

۸) جابجایی ستون ▼

با این دستور می توان جای دو ستون ماتریس مورد نظر را عوض کرد.

```
-swapColumns <mat> <col1> <col2>
```

تضمین می شود که مقادیر `col1` و `col2` مثبت هستند.

۹) ترانهاده ▼

با این دستور، می توان یک ماتریس را ترانهاده کرد.

```
-transpose <mat>
```

۱۰) دترمینان ▼

با این دستور می توان دترمینان یک ماتریس مربعی را محاسبه کرد.

```
-determinant <mat>
```

۱۱) اثر ماتریس

با این دستور می توان اثر یک ماتریس مربعی را محاسبه کرد.

```
-trace <mat>
```

۱۲) کپی کردن ماتریس

این دستور مشابه دستور مقدار دهنی است با این تفاوت که به جای گرفتن مقادیر از کاربر، مقادیر یک ماتریس را در دیگری کپی می کند.

```
-copy <ref> <dest>
```

بعد از وارد کردن آن، مقادیر ماتریس ref در dest کپی می شود.

۱۳) وارون (امتیازی)

به کمک دستور زیر، می توان یک ماتریس مربعی وارون پذیر را وارون کرد.

```
-invert <mat>
```

۱۴) عناصر قطری

این دستور، مقدار تمامی عناصر روی قطر اصلی را نگه می دارد و مابقی را 0 می کند.

```
-diag <mat>
```

۱۵) ماتریس همانی

این دستور مقادیر یک ماتریس همانی به ابعاد n را به ماتریس مربعی ref به همان ابعاد می دهد.

```
-identity <n> <ref>
```

۱۶) چاپ ماتریس

در هر قسمت از برنامه، با وارد شدن این دستور، ماتریس مورد نظر به کاربر نمایش داده می شود.

```
-print <mat>
```

خروجی

خروجی متناظر با دستورات بالا در این قسمت آورده شده است. دقت کنید که در هر مورد در صورت روبرو شدن با اشتباهات متعدد، تنها پیغام مربوط به اولین خطای باید بررسی شود و مابقی خطای ها نباید بررسی شوند.

▼ ۱) خروج از برنامه

این دستور خروجی ندارد و تنها باعث اتمام برنامه می شود.

▼ ۲) افزودن ماتریس

در صورتی که این ماتریس از قبل موجود باشد، خطای زیر چاپ می شود.

Matrix already exists

و در غیر این صورت

Matrix created successfully

▼ ۳) مقداردهی

اگر ماتریس وارد شده وجود نداشت

Matrix doesn't exist

و در غیر این صورت

Values set successfully

(۴) جمع ▼

اگر ماتریس ۱ یا ۲ وجود نداشتند، خطای زیر چاپ می شود.

Input matrices don't exist

و اگر ابعاد آن ها با هم مطابقت نداشت

Invalid dimensions for matrix addition

اگر ماتریس result از قبل وجود داشت

Matrix already exists

و در غیر این صورت

matrices added successfully

(۵) تفریق ▼

اگر ماتریس ۱ یا ۲ وجود نداشتند، خطای زیر چاپ می شود.

Input matrices don't exist

و اگر ابعاد آن ها با هم مطابقت نداشت

Invalid dimensions for matrix subtraction

اگر ماتریس result از قبل وجود داشت

Matrix already exists

و در غیر این صورت

matrices subtracted successfully

۶) ضرب ▼

اگر ماتریس ۱ یا ۲ وجود نداشتند، خطای زیر چاپ می شود.

Input matrices don't exist

و اگر ابعاد آن ها با هم مطابقت نداشت

Invalid dimensions for matrix multiplication

اگر ماتریس result از قبل وجود داشت

Matrix already exists

و در غیر این صورت

matrices multiplied successfully

۷) جابجایی سطر ▼

اگر ماتریس مورد نظر وجود نداشت، پیغام زیر چاپ می شود.

Matrix doesn't exist

اگر مقادیر row1 و row2 خارج از محدوده ماتریس بودند

Invalid input rows

و در غیر اینصورت

Rows swapped successfully

۸) جابجایی ستون ▼

اگر ماتریس مورد نظر وجود نداشت، پیغام زیر چاپ می شود.

Matrix doesn't exist

اگر مقادیر row1 و row2 خارج از محدوده ماتریس بودند

Invalid input columns

و در غیر این صورت

Columns swapped successfully

۹) ترانهاده ▼

اگر ماتریس مورد نظر وجود نداشت، پیغام زیر چاپ می شود.

Matrix doesn't exist

در غیر این صورت

Matrix transposed successfully

۱۰) دترمینان ▼

اگر ماتریس مورد نظر مربعی نبود، خطای زیر چاپ می شود.

Matrix must be square

در غیر این صورت مقدار دترمینان آن ماتریس با دقت ۲ رقم بعد از اعشار چاپ می شود.

۱۱) اثر ماتریس ▼

اگر ماتریس مورد نظر مربعی نبود، خطای زیر چاپ می شود.

Matrix must be square

در غیر این صورت مقدار اثر آن ماتریس با دقت ۲ رقم بعد از اعشار چاپ می شود.

۱۲) کپی کردن ماتریس ▼

اگر ماتریس ref وجود نداشت، پیغام زیر چاپ می شود.

Reference matrix doesn't exist

اگر ماتریس dest وجود نداشت

Destination matrix doesn't exist

اگر ابعاد آن دو مطابقت نداشت

Invalid dimensions for instantiation

در غیر این صورت

Values set successfully

۱۳) وارون (امتیازی) ▼

اگر ماتریس مورد نظر وجود نداشت، پیغام زیر چاپ می شود.

Matrix doesn't exist

اگر ماتریس مربعی نبود، پیغام زیر چاپ می شود.

Matrix must be square

اگر ماتریس وارون پذیر نبود

Matrix is singular

در غیر این صورت، پیغام زیر چاپ می شود.

Matrix inverted successfully

۱۴) عناصر قطری ▼

اگر ماتریس مورد نظر وجود نداشت، پیغام زیر چاپ می شود.

Matrix doesn't exist

اگر ماتریس مربعی نبود، پیغام زیر چاپ می شود.

Matrix must be square

در غیر این صورت

Matrix diagonalized successfully

۱۵) ماتریس همانی ▼

اگر ماتریس ref وجود نداشت

Matrix doesn't exist

اگر ابعاد ذکر شده با ابعاد ref سازگار نبود

Invalid input dimensions

در غیر این صورت

Values set successfully

۱۶) چاپ ماتریس ▼

اگر ماتریس مورد نظر وجود نداشت، خطای زیر چاپ می شود.

Matrix doesn't exist

در غیر این صورت ماتریس مورد نظر به صورت زیر چاپ می شود.

Matrix name: <matName>

و در ادامه خطوط مقادیر آن ماتریس چاپ می شود. برای مثال برای یک ماتریس به ابعاد ۳ در ۴ داریم

```
0 12 3 4.5
3 32.1 4 103
94 7 1.2 0
```

مثال

خونه خاله کدوم وره؟

- محدودیت زمان: ۴ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

آلیس این بار از طریق یک آبینه وارد سرزمین عجایب شد. اون می خواست به مهمونی چای Hatter بره ولی هیچ ایده ای نداشت که از کدوم مسیر بره! اون که خیلی غمگین شده و ترسیده، از شما می خواهد که براش سیستم GPS را بسازید که به کمک اون بتوانه همه رفیقاش و مسیر های بین اون ها رو به صورت یه گراف ذخیره کنه. شما هم چون خیلی آلیس رو دوست دارید، نتونستید جلوش نه بیارید!

تعاریف و نکات

در این سوال، به نکات و تعاریفی ساده درمورد گراف ها و ارتباطشان با ماتریس ها نیاز دارید که در ادامه توضیح داده شده است. اگر به اندازه کافی بر مبحث گراف تسلط دارید، می توانید این قسمت را نخوانید و به سراغ بخش ورودی بروید.

- درجه یک راس در گراف: تعداد تمامی یال های متصل به یک راس را درجه آن راس می گویند.
- ضریب خوش بندی یک گراف: نسبت تعداد کل یال های گراف به تعداد یال های گراف کامل (گرافی که بین همه راس هایش یال وجود دارد) با همین تعداد راس را نسبت خوش بندی آن گراف می گویند.
- ماتریس مجاورت: ماتریسی مربعی ($n \times n$) است که در آن $a_{i,j}$ تعداد راس های گراف است. طبق قرارداد، درایه $a_{i,j}$ این ماتریس مقدار ۱ را دارد اگر و تنها اگر بین این دو راس در گراف مدنظر یالی وجود داشته باشد. در غیر این صورت مقدار آن ۰ است. همچنین مقادیر عناصر روی قطر اصلی آن ۰ هستند.

نکته: برای یافتن تعداد مسیر های به طول m بین دو راس i و j از یک گراف، کافی است که ماتریس مجاورت را m بار در خودش ضرب کنیم و از درایه i و j آن استفاده کنیم.

نکته: بد ترین حالت ممکن برای کوتاه ترین مسیر بین دو راس A و Z در یک گراف، برابر $n-1$ است که در آن n تعداد راس‌های گراف است.

وروדי

در این سوال ۱۲ دستور ورودی داریم که در ادامه آورده شده‌اند. همچنین خروجی متناظر هر دستور در بخش خروجی آمده است. همچنین در این سوال، حتماً از regex استفاده کنید چرا که در غیر این صورت پردازش ورودی بسیار دشوار خواهد شد!

۱) خروج از برنامه ▼

تا زمانی که دستور end وارد نشود برنامه به ورودی گرفتن ادامه می‌دهد.

```
end
```

۲) افزودن گراف ▼

برای ذخیره سازی نقشه‌ی شهری جدید در سرزمین عجایب، گرافی جدید به شیوه زیر ساخته می‌شود.

```
add graph <graphname>
```

که در آن graphname نام گراف مورد نظر می‌باشد. همچنین تضمین می‌شود که نام گراف، تنها مت Shank از یک حرف الفبا خواهد بود.

۳) حذف کردن گراف ▼

با دستور زیر گراف حذف می‌شود.

```
remove graph <graphname>
```

۴) افزودن راس ▼

هر خانه در سرزمین عجایب، معادل یک راس گراف خواهد بود. پس برای اضافه کردن آن، از دستور زیر استفاده می‌شود.

```
add node <nodename> to graph <graphname>
```

که در آن nodename نام راس مورد نظر است. باز هم تضمین می شود که نام راس تنها متشکل از ۱ حرف الفباست.

▼ ۵) حذف کردن راس

با دستور زیر راس را حذف میکنیم.

```
remove node <nodename> from graph <graphname>
```

▼ ۶) افزودن یال

همان طور که گفتیم، مسیر بین خانه ها برای آليس خیلی مهم است. پس می توان مسیر بین خانه ها را با یال های گراف مدل کرد. به کمک دستور زیر به گراف یال اضافه می کنیم.

```
add edge between nodes <node1> and <node2> in graph <graphname>
```

که node1 و node2 نام راس هاییست که می خواهیم بین آن ها یال ایجاد کنیم.

نکته: منطقا در این مدل، در گراف مدنظر طوقه یا ابرراس نمی توان داشت!

▼ ۷) حذف کردن یال

با دستور زیر یال را حذف میکنیم.

```
remove edge between nodes <node1> and <node2> in graph <graphname>
```

▼ ۸) درجه راس

با دستور زیر باید درجه یک راس از یک گراف چاپ میشود.

```
print degree of the node <nodename> in graph <graphname>
```

▼ ۹) ماتریس مجاورت

با دستور زیر ماتریس مجاورت گراف چاپ میشود.

```
print adjacency matrix of the graph <graphname>
```

نکته: ترتیب راس ها در سطر و ستون ها، مشابه ترتیب اضافه شدن راس ها به گراف است.

▼ ۱۰) تعداد مسیر

با دستور زیر تعداد مسیر های به طول m را بین دو راس از یک گراف را چاپ میکنیم.

```
print the number of pathes with <m> edges between nodes <node1> and <node2> in graph <
```

▼ ۱۱) کوتاه ترین مسیر

با دستور زیر طول کوتاه ترین مسیر بین دو راس چاپ میشود.

```
print the length of shortest path between nodes <node1> and <node2> in graph <graphname>
```

▼ ۱۲) ضریب خوش بندی

با دستور زیر ضریب خوش بندی گراف چاپ میشود.

```
print clustering coefficient of the graph <graphname>
```

خروجی

خروجی متناسب با دستورات بالا به تفکیک آورده شده است. دقت کنید که در هر مرحله درصورت وجود ورودی های اشتباه متعدد، تنها پیغام مربوط به اولین خطاب باید چاپ شود و بقیه نباید بررسی شوند.

▼ ۱) خروج از برنامه

این دستور خروجی ای ندارد و صرفا باعث پایان یافتن برنامه می شود.

▼ ۲) افزودن گراف

اگر گراف از قبل وجود داشت، خطای زیر چاپ می شود.

```
graph <graphname> already exists
```

در غیر این صورت

```
graph is added successfully
```

▼ ۳) حذف کردن گراف

اگر گرافی با این اسم وجود نداشت ، پیام زیر چاپ می شود.

```
graph <graphname> doesn't exist
```

در غیر این صورت، پیغام زیر چاپ می شود.

```
graph is removed successfully
```

▼ ۴) افزودن راس

اگر گرافی با این اسم وجود نداشت ، پیام زیر چاپ می شود.

```
graph <graphname> doesn't exist
```

اگر راسی به این اسم از قبل وجود داشته باشد، پیام زیر چاپ میشود.

```
node <nodename> already exists in graph <graphname>
```

در غیر این صورت

node is added successfully

۵) حذف کردن راس ▼

اگر گرافی با این اسم وجود نداشت ، پیام زیر چاپ می شود.

graph <graphname> doesn't exist

اگر راسی با این اسم در گراف وجود نداشت

node <nodename> doesn't exist in graph <graphname>

در غیر این صورت

node is removed successfully

۶) افزودن یال ▼

اگر گرافی با این اسم وجود نداشت ، پیام زیر چاپ می شود.

graph <graphname> doesn't exist

اگر هیچ کدام از راس ها وجود نداشتند، پیام زیر چاپ میشود.

nodes <node1> and <node2> don't exist in graph <graphname>

اگر یکی از راس ها وجود نداشت، پیام زیر چاپ میشود.

node <nodename> doesn't exist in graph <graphname>

اگر نام دو راس یکسان بود، پیام زیر چاپ میشود.

You can't add edge from a node to itself

اگر بین دو راس از قبل یال وجود داشته باشد، پیام زیر چاپ میشود.

edge already exists between nodes <node1> and <node2> in graph <graphname>

و اگر هیچ کدام از خطاهای بالا رخ نداده بود

edge is added successfully

▼ ۷) حذف کردن یال

اگر گرافی با این اسم وجود نداشت ، پیام زیر چاپ می شود.

graph <graphname> doesn't exist

اگر هیچ کدام از راسها وجود نداشتند، پیام زیر چاپ میشود.

nodes <node1> and <node2> don't exist in graph <graphname>

اگر یکی از راسها وجود نداشت، پیام زیر چاپ میشود.

node <nodename> doesn't exist in graph <graphname>

اگر بین دو راس یالی وجود نداشته باشد، پیام زیر چاپ میشود.

there is no edges between nodes <node1> and <node2> in graph <graphname>

و اگر هیچ کدام از خطاهای بالا رخ نداده بود

edge is removed successfully

۸) درجه راس ▼

اگر گرافی با این اسم وجود نداشت، پیام زیر چاپ می شود.

graph <graphname> doesn't exist

اگر راس مورد نظر وجود نداشت، پیام زیر چاپ میشود.

node <nodename> doesn't exist in graph <graphname>

در غیر این صورت درجه آن راس چاپ می شود.

۹) ماتریس مجاورت ▼

اگر گرافی با این اسم وجود نداشت پیام زیر چاپ میشود.

graph <graphname> doesn't exist

در این صورت، ماتریس مجاورت را مانند مثال زیر چاپ میکنیم.

```
0 1 0
1 0 1
0 1 0
```

۱۰) تعداد مسیر ▼

اگر گرافی با این اسم وجود نداشت، پیام زیر چاپ می شود.

graph <graphname> doesn't exist

اگر هیچ کدام از راس ها وجود نداشتند، پیام زیر چاپ میشود.

nodes <node1> and <node2> don't exist in graph <graphname>

اگر یکی از راس ها وجود نداشت، پیام زیر چاپ میشود.

node <nodename> doesn't exist in graph <graphname>

در غیر این صورت تعداد مسیر های به طول m چاپ میشود.

۱۱) کوتاه ترین مسیر ▼

اگر گرافی با این اسم وجود نداشت ، پیام زیر چاپ می شود.

graph <graphname> doesn't exist

اگر هیچ کدام از راس ها وجود نداشتند، پیام زیر چاپ میشود.

nodes <node1> and <node2> don't exist in graph <graphname>

اگر یکی از راس ها وجود نداشت، پیام زیر چاپ میشود.

node <nodename> doesn't exist in graph <graphname>

در غیر این صورت طول کوتاه ترین مسیر چاپ میشود.

۱۲) ضریب خوش بندی ▼

اگر گرافی به این نام وجود نداشت، پیام زیر چاپ میشود.

```
graph <graphname> doesn't exist
```

در غیر این صورت ضریب خوش بندی گراف چاپ میشود.

نکته: خروجی باید از جنس double باشد.

همچنین در هرجای برنامه، ممکن است کاربر دستورات بعضاً بی معنا و نامعتبر وارد کند که در این صورت باید پیغام زیر را چاپ کنید.

```
invalid command
```

در رابطه با ضرب ماتریس ها، می توانید از کلاس ساخته شده در سوال ۱ و یا حتی هرگونه کتاب خانه آماده زبان Java استفاده کنید.

مثال

نمونه ورودی

```
add graph q
add node a to graph q
add node b to graph q
add node c to graph q
add node d to graph q
add node e to graph q
add node f to graph q
add node g to graph q
add node h to graph q
add edge between nodes a and b in graph q
add edge between nodes a and c in graph q
add edge between nodes a and d in graph q
add edge between nodes b and c in graph q
add edge between nodes b and d in graph q
add edge between nodes b and e in graph q
```

پول نقد نداریم!

- محدودیت زمان: ۲ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

آلیس می خواست به گشت و گذار در سرزمین عجایب ادامه بده اما متوجه شد که تاکسی های اونجا پول نقد دریافت نمی کنن! از طرفی تنها سیستم بانکی موجود در اون سرزمین زیر نظر برادران دوقلو بود که در اثر جنگ و نزاع پیوسته بین آن دو، از بین رفته بود! آلیس که خیلی فوری نیاز به افتتاح یک حساب بانکی در سرزمین عجایب داشت، از شما می خواهد که مدیریت این بانک رو در دست گرفته و سیستم اون رو پیاده کنید. در عوض، اون به شما قول می ده که به دلیل نفوذی که بین تی ای های درس OOP داره، نمره سوال سوم تمرین رو برای شما کامل رد کنه!

همون طور که از نامش پیداست، این سیستم بانکی در سر زمین عجایب تعریف شده و هیچ جای اون با عقل و منطق جور در نمیاد! هر حساب کاربری، دارای ویژگی های زیره که جلوتر به توضیح اون ها پرداخته میشه.

accountName, accountID, password, balance, interest, transactions, transferLimit

وروڈی

در این سوال ۱۱۴ دستور وروڈی داریم که به توضیح هر کدام خواهیم پرداخت. خروجی های هر دستور نیز در بخش خروجی آمده است.

▼) خروج از برنامه

تا زمانی که دستور end- وارد نشود برنامه به وروڈی گرفتن ادامه میدهد.

-end

▼) افزودن کاربر

در ابتداء، هر شخص باید بتواند در این سیستم برای خود یک حساب بانکی بسازد که با دستور زیر انجام می‌شود.

```
-create <accountName> <password>
```

اما از آنجایی که نام هر فرد بخشن اساسی هویت او را تشکیل میدهد، در سرزمین عجایب باید هر accountName شامل نام و نام خانوادگی شخص باشد که توسط یک کاراکتر whitespace از هم مجزا شده‌اند!

گذروازه را هم که خودتان بهتر می‌دانید. باید حداقل از ۸ کاراکتر تشکیل شده باشد به طوری که این ۸ کاراکتر از میان حروف الفبای انگلیسی و ارقام ۰ تا ۹ قابل انتخاب هستند. در ضمن، مگر می‌توان گذروازه مناسبی بدون استفاده از ارقام ۰ تا ۹ ساخت؟!

حالات دیگری از این دستور وجود دارد که به ترتیب برای نسبت دادن مقادیر موجودی و سود اولیه به حساب استفاده می‌شوند.

```
-create <accountName> <password> <balance>
-create <accountName> <password> <balance> <interest>
```

در صورتی که هر کدام از این ۲ مقدار صراحتا ذکر نشوند، به موقع ساخت، حساب مقادیر ۰ به آنها تعلق می‌گیرد. عملکرد سود، اصلاً مطابق آنچه انتظار دارید نیست چرا که امپراتور زمان، گذر زمان را در سرزمین عجایب متوقف کرده است و خبری از سود سالانه نیست! در عوض، کارمندان بانک به ازای هر بار ورود و خروج یک کاربر، سود را بر روی موجودی او اعمال می‌کنند!

اما حواسitan باشد که با وجود اینکه این بانک در سرزمین عجایب قرار دارد، کاملاً موافق با ایدئولوژی‌های شایسته سالاری در جامعه است و هرگز پول مفت به کسی تعلق نمی‌گیرد! بدین معنا که حداکثر سود دریافتی از این بانک، ۲۵ درصد است!

▼) ورود به حساب کاربری

بعد از ساختن حساب، هر شخص گزینه ورود به سیستم را دارد که به کمک دستور زیر انجام می شود.

-login <accountName> <password>

۴) خروج از حساب کاربری ▼

با این دستور کاربر از حساب خود خارج میشود.

logout

۵) تغییر رمزعبور ▼

وی می تواند رمز خود را تغییر دهد که به کمک دستور زیر انجام می شود.

-changePassword <newPassword>

۶) مشاهده جزئیات حساب ▼

کاربر می تواند جزئیات مربوط به حساب خود را مشاهده کند که توسط دستور زیر انجام می شود.

-accountDetails

۷) برداشت وجه ▼

امکان برداشت وجه با دستور زیر وجود دارد.

-withdraw <amount>

۸) واریز وجه ▼

مشابه مورد ۷، به صورت زیر انجام می شود.

-deposit <amount>

▼ ۹) کارت به کارت

بانک سرزمین عجایب، همواره به دنبال ارائه آپشن های جدید به مشتری هاست. پس آپشن کارت به کارت را اخیرا به سیستم خود باید اضافه کنی... ببخشید! اضافه کرده است. این کار از طریق دستور زیر انجام می شود.

```
-transferMoney <destinationID> <amount>
```

در دستور فوق، منظور از destinationID همان شماره مشتری مدنظر است که می خواهیم وجه را برای او منتقل کنیم. (کارمندان این بانک به مشتری های خود احترام بسیار زیادی می گذارند چرا که در داخل بانک، به جای استفاده از نام آنها، آنها را با شماره شان صدا می زنند!) برای اولین مشتری، شماره ۱ به وی تعلق می گیرد و با باز شدن حساب های بیشتر، این مقدار به ازای هر مشتری یک واحد زیاد می شود!

همچنین، برای هر بار انتقال وجه، هر کاربر یک مقدار transferLimit دارد که می تواند فعال یا غیرفعال باشد. در صورت فعال بودن، در هر بار انتقال وجه، نمی توان بیشتر transferLimit کارت به کارت کرد. در غیر این صورت، محدودیتی وجود ندارد.

▼ ۱۰) سوابق کارت به کارت

آپشن دیگر این بانک، ذخیره سازی تراکنش های انجام شده از طریق کارت به کارت برای طرفین قرارداد است! یعنی به ازای هر کارت به کارت انجام شده، عضوی به transactions موجود در هر دو حساب بانکی اضافه خواهد شد. چرا که هر کاربر می تواند در ادامه، سوابق تراکنش های خود را چاپ کند که به صورت زیر انجام می شود.

```
-displayTransactions
```

▼ ۱۱) سقف کارت به کارت

کاربر می تواند با وارد کردن دستور زیر، از میزان سقف انتقال وجه خود مطلع شود.

```
-displayLimit
```

▼ ۱۲) تغییر سقف انتقال وجه

گاها پیش می آید که برادران دوقلو فضولیshan گل می کند و سقف انتقال وجه مشتری را دستکاری می کنند. این کار را برادر اول توسط دستور زیر انجام می دهد.

```
-setLimit <amount>
```

▼ ۱۳) غیرفعال کردن سقف

در هر آن، برادر دوم می تواند با دستور زیر، سقف را غیرفعال کند.

```
-disableLimit
```

▼ ۱۴) تغییر مقدار سود

متاسفانه، در سرزمین عجایب، همه چیز به شدت تحت تاثیر نویز و در فضای احتمالاتی است. به همین منظور، کارمندان بانک اجازه نمی دهند که مقدار سود هر کاربر ثابت بماند و با وارد کردن دستور زیر، آن را بازیچه خود قرار می دهند.

```
-setInterest <amount>
```

لازم به ذکر است که طبق قوانین، مقدار وارد شده باید همچنان بین ° و ۲۰ درصد باشد.

خروجی

خروجی هر دستور بالا به ترتیب نوشته شده است. لطفا به توضیحات هر دستور توجه فرمایید. همچنین دقت کنید که در صورت مواجه شدن با اشتباه های متعدد برای هر دستور، تنها پیغام خطای مربوط به اولین اشتباه را باید چاپ کنید و مابقی خطاهای را نباید بررسی کنید!

▼ ۱) خروج از برنامه

این دستور خروجی ندارد و تنها باعث می شود که برنامه پایان یابد.

▼ ۲) افزودن کاربر

از آنجایی که کارمندان این بازک بسیار دست و پا چلفتی هستند، ممکن است که دستور بالا را اشتباه وارد کنند. پس در هر صورت بروز هرگونه مغایرت، باید پیغام خطای مناسبی چاپ شود و خطاهای بعد از آن بررسی نشوند.

به ترتیب، اگر فرمت نام و نام خانوادگی شخص درست نبود (متشكل از حروف الفبای انگلیسی و جدا شده توسط ۱ کاراکتر whitespace)، پیغام زیر چاپ می‌شود.

Username format invalid!

در صورتی که فرمت گذرواژه مطابق آنچه گفته شد نبود (حداقل ۸ کاراکتر، دارای حداقل ۱ عدد از ۰ تا ۹، عدم وجود عدد در ابتدای عبارت)، پیغام زیر چاپ می‌شود.

Password format invalid!

اگر شخصی با این نام و نام خانوادگی از قبل ثبت نام کرده بود، پیغام زیر چاپ می‌شود.

Account name already taken!

حال در صورت ذکر شدن مقدار برای موجودی، اگر منفی بود، عبارت زیر چاپ می‌شود.

Invalid amount for balance!

در صورت ذکر شدن مقدار برای سود، اگر بین ۰ و ۲۰ نبود، عبارت زیر چاپ می‌شود.

Invalid amount for interest!

در صورت ساخته شدن حساب با موفقیت، پیغام زیر به کاربر نمایش داده خواهد شد:

Account created successfully.

۳) ورود به حساب کاربری ▼

همچنان اشتباهات ذکر شده در هنگام ساختن حساب (نامعتبر بودن فرمت نام و نام خانوادگی و گذرواژه)، محتمل هستند و باید در این قسمت نیز بررسی شوند. علاوه بر این دو، در صورت موجود نبودن نام کاربر، پیغام زیر نمایش داده خواهد شد.

Account name not found!

و در صورت اشتباه بودن گذرواژه

Incorrect password!

در صورت ورودی موفقیت آمیز، پیغام زیر نمایش داده خواهد شد.

Logged in successfully with ID <accountID>.

همان که قبل ذکر شد، متغیر accountID همان شماره مشتری است که به اولین مشتری ثبت نامی، شماره ا تعلق می گیرد و این مقدار به ازای هر مشتری زیاد می شود.

۴) خروج از حساب کاربری ▼

در صورت موفقیت آمیز بودن، پیغام زیر نمایش داده خواهد شد.

Logged out successfully.

۵) تغییر رمزعبور ▼

مشابه مورد ۲، فرمت گذرواژه ورودی باید بررسی شود و در صورت اشتباه بودن، خطای مربوطه چاپ شود. همچنین از آنجایی که کارمندان این بانک آن قدر بیکار نیستند که دستورات بیهوده تغییر رمز را پردازش

کنند، رمز عبور جدید باید با مقدار قبلى تفاوت داشته باشد! در صورت ورود مقدار تکراری، خطای زیر چاپ می شود.

Password already in use!

و در صورت تغییر گذرواژه پیغام زیر چاپ خواهد شد.

Password changed successfully.

۶) مشاهده جزئیات حساب ▼

اطلاعات به صورت زیر به وی نمایش داده خواهد شد.

Account name: <accountName>

Account ID: <accountID>

Current Balance: <balance>\$

به طوری که موجودی وی تا ۲ رقم بعد از اعشار چاپ خواهد شد.

۷) برداشت وجه ▼

مقدار وارد شده باید مثبت باشد. (مگر یادتان رفته که کارمندان بانک علاف نیستند!)

پس در صورت بروز اشتباه، باید پیغام زیر چاپ شود.

Can not withdraw non-positive amount!

همچنین اگر مقدار وارد شده از موجودی فرد بیشتر بود، باید پیغام زیر نمایش داده شود.

Withdraw amount exceeds balance!

در صورت موفقیت آمیز بودن، پیغام زیر نمایش داده می شود.

<amount>\$ withdrew successfully.

که در آن باید مقدار ذکر شده دقت ۲ رقم بعد از اعشار را داشته باشد.

(۸) واریز وجه ▼

مشابه مورد ۷، مقدار وارد شده اگر منفی بود

Can not deposit non-positive amount!

و در صورت موفقیت آمیز بودن

<amount>\$ deposited successfully.

(۹) کارت به کارت ▼

اگر مقدار وجه انتقالی بیشتر از سقف انتقال بود

Transfer amount exceeds limit!

اگر مقدار وجه انتقالی بیشتر از موجودی شخص بود

Transfer amount exceeds balance!

و اگر مقصد مدنظر وجود نداشت

Entered ID does not exist!

در نهایت اگر کارت به کارت با موفقیت انجام شد، باید دستور زیر چاپ شود.

Transferred <amount>\$ to <accountName> successfully.

که به جای اولین متغیر باید مقدار وجهه انتقالی با دقت ۲ رقم بعد از اعشار و به جای دومین متغیر نام و نام خانوادگی مقصد نوشته شود. بله درست دیدید! نام و نام خانوادگی! پس به نفع سیستم بانکی است که به کمک شماره مشتری بتوان نام و نام خانوادگی او را یافت!

▼ ۱۰) سوابق کارت به کارت

با وارد کردن این دستور، در ابتدا پیغام زیر نمایش داد خواهد شد.

Report of transactions:

در خطوط بعدی، گزارش تراکنش ها چاپ می شود. بدین صورت که اگر وجهی به حساب کاربر مدنظر وارد شده بود، به صورت زیر گزارش داده می شود.

Received <amount>\$ from <accountID>

و در صورتی که وجهی از حساب وی خارج شده بود.

Transferred <amount>\$ to <senderID>

▼ ۱۱) سقف کارت به کارت

در صورت فعل بودن سقف، عبارت زیر چاپ می شود.

No more than <amount>\$ each time!

و در صورت غیر فعل بودن

There is no limit to money transfer.

▼ ۱۲) تغییر سقف انتقال وجه

اگر مقدار وارد شده مثبت نبود، پیغام زیر چاپ می شود.

Can not set non-positive amount to limit!

و اما در صورت اجرای بدون نقص، پیغام زیر نمایش داده خواهد شد.

Limit (<amount>\$) set to each transaction successfully.

که دقیق نمایش تا ۲ رقم بعد از اعشار است.

▼ ۱۳) غیرفعال کردن سقف

پیغام زیر نمایش داده می شود.

Transaction limit disabled successfully.

▼ ۱۴) تغییر مقدار سود

اگر مقدار وارد شده بین ۰ و ۲۰ نبود

Invalid amount for interest!

و اگر مقدار آن به درستی تغییر کرد :

Interest changed to <amount>% successfully.

نکته بسیار مهم : کارمندان دست و پا چلفتی، دستورات بعضاً اشتباه و نامعتبر نیز وارد می کنند که در هر جای برنامه، در صورت رو برو شدن با این دستورات باید پیغام زیر را چاپ کنید.

Invalid command!