

۳ مدرسان: رامتین خسروی، محمدامین صادقی

۴ طراحان: جمعی از تی‌ای‌ها!

۵ موعد تحویل فاز ۱: سه‌شنبه ۱۹ اردیبهشت

۶ مقدمه

۷ هدف از این تمرین آشنایی شما با طراحی شیء‌گرا، برنامه‌نویسی تحت شبکه و آشنایی با کتابخانه‌ی کیوت^۱ (یا کیوتی)

۸ است که هر کدام در یک فاز مورد بررسی قرار می‌گیرد. بنابراین این تمرین ۳ فاز دارد و هر کدام در ادامه‌ی فاز قبل است.

۹ فاز اول این تمرین همانند اکثر تمرین‌های قبلی تحت محیط خط فرمان^۲ است و شما باید یک برنامه‌ی واحد برای این فاز

۱۰ تحویل بدهید. در این فاز شما باید ارتباط کلاس‌ها را به خوبی بدست بیاورید و آن را پیاده‌سازی کنید.

۱۱ فاز دوم آن نیز تحت محیط خط فرمان است با این تفاوت که برنامه‌ی شما به یک سرور و تعدادی کلاینت می‌شکند.

۱۲ فاز سوم این پروژه طراحی یک محیط گرافیکی مناسب برای کلاینت‌های برنامه به وسیله‌ی کتابخانه‌ی قدرتمند کیوت

۱۳ می‌باشد.

۱۴ توجه کنید که برخی از دستورات این فاز صرفاً جهت راحتی شما برای فازهای بعدی است و سعی شده‌است که دستورات

۱۵ به گونه‌ای باشد که تفکیک برنامه به کلاینت‌های مختلف و پیشبرد آن توسط مکانیزم سرکشی^۳ امکان‌پذیر باشد.

۱۶

۱۷

۱۸

۱۹

۲۰

۲۱

۲۲

۲۳

۲۴

^۱ [QT](#)

^۲ [CLI](#)

^۳ [Polling](#)



۲۶

۲۷

۲۸

۲۹

۳۰

۳۱

۳۲

۳۳

۳۴

۳۵

۳۶

۳۷

۳۸

۳۹

۴۰

سه سال بعد از مرگ دارت ویدر^۴، از طرف یکی از بستگان دور شاهزاده لیا اورگانا^۵ ارث قابل توجهی به او و هان سولو^۶ رسید. لیا و هان - که خسته از جنگ‌های طولانی سعی می‌کردند دوران بازنشستگی خود را بگذرانند - تصمیم گرفتند از این پول استفاده‌ی مناسبی بکنند و یک سامانه‌ی جابه‌جایی میان‌سیاره‌ای راه بیندازند. از آن جا که هان تجربیات فراوانی از سفرهایش با شاهین هزاره داشت، قرار شد مدیریت را او برعهده بگیرد. آن‌ها اسم شرکت خود را “تاکسی‌سپاره” گذاشتند. اما از آنجایی که سولو هنوز به خدمت مقدس سربازی نرفته‌است و کارت پایان خدمت ندارد به شکل قانونی نمی‌تواند به کهکشان‌های دیگر برود. بنابراین تصمیم می‌گیرد که شرکت خود را دانش‌بنیان کند و خدمت خود را در همین شرکت بگذرانند.

سولو تصمیم گرفت از یک دوست قدیمی - شما! - بخواهد برای سامانه‌شان یک سیستم کامپیوتری طراحی کنید تا بتواند شرکت را دانش‌بنیان! کند و همچنین پول بیشتری دربیاورد. بعد از مشورت با لیا، ویژگی‌هایی برای این سیستم کامپیوتری مشخص شد.

کاربران سیستم

در این سیستم سه نوع کاربر وجود دارد: مسافر، راننده و مدیر سیستم

هر کاربر درخواست‌هایش را به وسیله‌ی دستورهای در سیستم وارد می‌کند. همچنین هر کاربر دارای یک نام کاربری یکتا می‌باشد. مدیر سیستم با نام کاربری “admin” شناخته می‌شود.

⁴ [Darth Vader](#)

⁵ [Princess Leia](#)

⁶ [Han Solo](#)

۴۱ قالب دستورها

۴۲ به طور کلی قالب دستورهایی که هر کاربر می‌تواند وارد کند به شکل زیر است:

```
<username> <command> <param1> <param2> ...
```

```
<result>
```

- ۴۳ ----- ۴۴
- ۴۵
- ۴۶ - کاربر برای اجرای همه‌ی دستورها (به جز ثبت‌نام) لازم است در سیستم لاگین کرده باشد.
- ۴۷ - برنامه‌ی شما باید نام دستور و پارامترها و قالب آن‌ها را بررسی نماید و در صورت خطا پیغام مناسب نمایش دهد.
- ۴۸ - عبارات هر قالب دستور به وسیله‌ی یک کاراکتر فاصله از هم جدا می‌شوند. همچنین در خود عبارات از کاراکتر فاصله استفاده نخواهد شد (منظور از عبارات کلماتی هستند که در هر دستور جایگزین <user> یا <command> یا <param1> یا ... می‌شوند).
- ۴۹
- ۵۰
- ۵۱ - در قسمت خروجی صرفاً مواردی که لازم است بیان شود مشخص شده‌است. برنامه‌ی شما باید آن‌ها را در قالب مناسب نمایش دهد (این فاز به صورت دستی تحویل گرفته می‌شود).
- ۵۲

۵۳ ثبت‌نام

- ۵۴ هر یک از کاربران سیستم (به جز مدیر سیستم) برای استفاده باید ابتدا در سیستم ثبت‌نام کنند. رانندگان علاوه بر اطلاعات خود باید مشخصات سفینه‌ی خود را نیز وارد کنند.
- ۵۵

۵۶ ۱. ثبت‌نام مسافر

۵۷ برای ثبت‌نام، مسافر کافی است دستور زیر را وارد نماید:

```
<passenger_username> register_passenger <password> <phone_number>
```

- ۵۹ - هنگام ثبت‌نام مسافر یکتا بودن نام کاربری آن بین تمام کاربران باید بررسی شود.
- ۶۰ - هنگام ثبت‌نام مسافر باید یکتا بودن شماره‌ی تلفن او بین مسافران قبلی بررسی شود.

۶۱ ۲. ثبت‌نام راننده

۶۲ برای ثبت‌نام، راننده ابتدا دستور زیر را وارد می‌کند:

```
<driver_username> register_driver <password> <spaceship_number>
```

```
<spaceship_model> <production_year> <spaceship_color> [VIP]
```

- ۶۵ - هنگام ثبت‌نام راننده یکتا بودن نام کاربری آن بین تمام کاربران باید بررسی شود.
- ۶۶ - هنگام ثبت‌نام راننده باید یکتا بودن شماره‌ی سفینه‌ی او در بین رانندگان قبلی بررسی شود.
- ۶۷ - مدل سفینه‌ی راننده باید در بین مدل‌های تعریف‌شده در سیستم باشد. برنامه‌ی شما باید مدل‌های مجاز را در ابتدای اجرا از فایل "vehicles.cfg" در کنار برنامه بخواند.
- ۶۸
- ۶۹ - یک راننده می‌تواند از نوع VIP باشد. در این صورت انتهای درخواست ثبت‌نام از کلمه‌ی VIP استفاده می‌کند.
- ۷۰ در صورت VIP بودن فقط مسافرانی که در خواست سفر VIP دارند را می‌تواند جابجا کند و در غیر اینصورت فقط سفرهای عادی را می‌تواند بپذیرد.
- ۷۱
- ۷۲

۷۳ پس از آن مدیر سیستم باید به شکل زیر ثبت نام راننده را تایید یا رد کند:

```
admin accept_registration <driver_username>
admin reject_registration <driver_username>
```

- ۷۶ - قبل از نهایی شدن ثبت نام راننده فرد دیگری امکان ثبت نام با نام او را ندارد.
- ۷۷ - در صورت عدم پذیرش ثبت نام توسط مدیر امکان ثبت نام با نام رد شده وجود دارد.
- ۷۸ - در صورتی که راننده هنوز تایید نشده باشد هنگام احراز هویت (که در ادامه توضیح داده می شود) پیام “در حال بررسی ثبت نام” به او نمایش داده می شود.

۸۰ مدیر سیستم می تواند درخواست های ثبت نام را به شکل زیر مشاهده کند:

```
admin show_registration_requests
```

```
-----
[<username> <spaceship_number> <spaceship_model> <production_date>
<spaceship_color> [VIP]\n]*
```

۸۵ کد تخفیف برای ثبت نام

- ۸۶ مسافران سیستم می توانند دوستان خود را به سیستم دعوت کنند و به این ترتیب از تخفیف سیستم برخوردار شوند. هر
- ۸۷ کاربر می تواند از سیستم درخواست کد تخفیف کند و سیستم، کد تخفیف متناظر او را نمایش می دهد. سپس او می تواند
- ۸۸ کد تخفیف خود را به پنج نفر بدهد تا اعتبار آن ها و خودش ده واحد افزایش یابد. پس از استفاده پنج نفر از کد تخفیف
- ۸۹ این کاربر، کد تخفیف او نامعتبر می شود.
- ۹۰ مثال: اگر چهار نفر از کد تخفیف کاربر A همزمان استفاده کنند، اعتبار کاربر A چهار واحد افزایش می یابد و اعتبار
- ۹۱ اولیه ی چهار کاربر دیگر نیز ده واحد می شود و تنها یک کاربر دیگر می تواند از کد تخفیف کاربر A استفاده کند.
- ۹۲ نکته: اعتباری که برای مسافر از طریق کد تخفیف بدست می آید به اعتبار سیستم تاکسیاره اضافه نمی شود.
- ۹۳ نحوه ی دریافت کد تخفیف به شکل زیر است:

```
<passenger_username> get_discount_code
-----
<discount_code> <available_count>
```

- ۹۷ - خروجی <available_count> تعداد بارهایی است که کد تخفیف مربوطه قابل استفاده است.

۹۸ نحوه ی استفاده از کد تخفیف هنگام ثبت نام نیز به شکل زیر است:

```
<passenger_username> register_passenger <password> <phone_number>
<discount_code>
```

۱۰۱ ورود

- ۱۰۲ هر کاربر برای اجرای دستورات خود ابتدا لازم است که با دستور زیر وارد سیستم شود:

```
<username> login <password>
```

۱۰۴

۱۰۵

- رمز عبور مدیر سیستم s3cret است.

۱۰۶

- همزمان چند نفر می‌توانند در سیستم لاگین کنند (برای تمایز کاربر هر دستور، طبق قالب معرفی شده، ابتدای هر

۱۰۷

دستور نام کاربری آن باید بیان شود).

۱۰۸

- در صورتی که راننده هنوز تایید نشده باشد هنگام احراز هویت پیامی با مضمون "در حال بررسی ثبت نام" به او

۱۰۹

نمایش داده می‌شود.

۱۱۰

- در صورتی که کاربر قبل از ورود به سیستم دستوری را وارد نماید، سیستم به او پیغام خطا می‌دهد و دستور او

۱۱۱

اجرا نخواهد شد.

۱۱۲

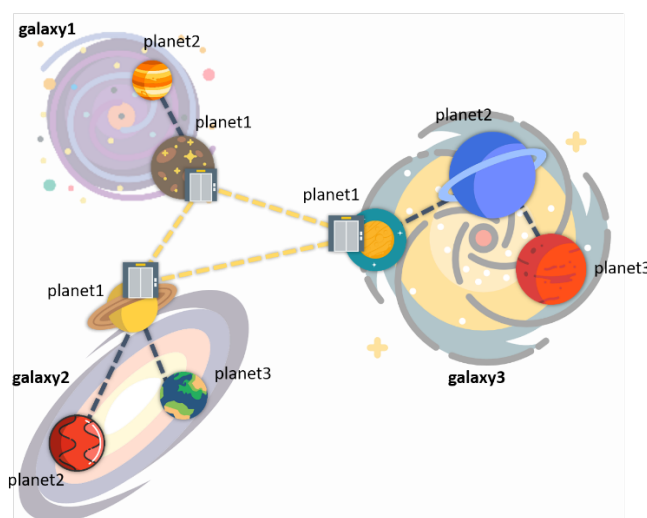
برای خروج نیز باید دستور زیر را وارد کند:

۱۱۳

`<username> logout`

۱۱۴

توپولوژی فضایی



۱۱۵

شکل ۱

۱۱۶

۱۱۷

در این سیستم تعدادی کهکشان وجود دارد و هر کهکشان شامل تعدادی سیاره است. بین برخی سیارات داخل کهکشان

۱۱۸

خطوط فضایی وجود دارد و حرکت بین سیاره‌های داخل یک کهکشان فقط از این خطوط امکان پذیر است. هر کهکشان

۱۱۹

دارای یک گیت ورود و خروج است که برای سفر از یک کهکشان به کهکشان دیگر لازم است از این گیت به گیت

۱۲۰

کهکشان دیگر رفت. سفر بین این دو گیت با سرعت نور و با هزینه‌ی صفر انجام می‌پذیرد. همچنین از گیت یک کهکشان

۱۲۱

می‌توان به گیت‌های همه‌ی کهکشان‌ها سفر کرد.

۱۲۲

هر آدرس شامل یک کهکشان و یک سیاره می‌شود. یک آدرس معتبر است اگر کهکشان آن موجود باشد و شامل سیاره‌ی

۱۲۳

مذکور باشد. آدرس شامل کاراکتر فاصله نیست و اجزای آن با کاراکتر کاما جدا می‌شوند. فرمت هر آدرس به صورت زیر

۱۲۴

است:

۱۲۵

`<galaxy>,<planet>`

۱۲۶

لیست کهکشان‌ها و سیاره‌هایشان و ارتباط بین سیاره‌ها در فایل "topology.cfg" کنار برنامه به فرمت زیر وجود دارد:

<تعداد کهکشان‌ها>	۱۲۷
<تعداد سیاره‌های کهکشان اول> <نام کهکشان اول>	۱۲۸
<تعداد سیاره‌های متصل به سیاره اول> <نام سیاره اول>	۱۲۹
<نام اولین سیاره متصل به سیاره اول>	۱۳۰
<نام دومین سیاره متصل به سیاره اول>	۱۳۱
<نام سومین سیاره متصل به سیاره اول>	۱۳۲
...	۱۳۳
<تعداد سیاره‌های متصل به سیاره دوم> <نام سیاره دوم>	۱۳۴
<نام اولین سیاره متصل به سیاره دوم>	۱۳۵
...	۱۳۶
...	۱۳۷
<تعداد سیاره‌های کهکشان دوم> <نام کهکشان دوم>	۱۳۸
<تعداد سیاره‌های متصل به سیاره اول> <نام سیاره اول>	۱۳۹
<نام اولین سیاره متصل به سیاره اول>	۱۴۰
...	۱۴۱
<تعداد سیاره‌های متصل به سیاره دوم> <نام سیاره دوم>	۱۴۲
<نام اولین سیاره متصل به سیاره دوم>	۱۴۳
...	۱۴۴
...	۱۴۵

- ۱۴۶ - فاصله‌های ابتدای هر خط صرفاً برای راحتی در خواندن نوشته شده‌اند و در فایل توپولوژی وجود ندارند.
- ۱۴۷ - نام کهکشان‌ها یکتاست و نام سیاره‌های یک کهکشان داخل آن کهکشان یکتاست.
- ۱۴۸ - اولین سیاره‌ی هر کهکشان را گیت آن کهکشان در نظر بگیرید.
- ۱۴۹ - برای شکل ۱ این فایل در کنار صورت پروژه قرار داده شده‌است.
- ۱۵۰ - به عنوان مثال توپولوژی زیر شامل دو کهکشان که هر کدام دو سیاره دارند است:

2	۱۵۱
LeftGalaxy 2	۱۵۲
PlanetA 1	۱۵۳
PlanetB	۱۵۴
PlanetB 1	۱۵۵
PlanetA	۱۵۶
RightGalaxy 2	۱۵۷
PlanetC 1	۱۵۸
PlanetD	۱۵۹
PlanetD 1	۱۶۰
PlanetC	۱۶۱

فاصله‌ی دو مکان اگر هر دو در یک کهکشان باشند برابر تعداد خطوط هوایی است که باید طی شود تا از یکی به دیگری

رسید. در فایل توپولوژی تضمین می‌شود که گراف ارتباطی یک کهکشان درخت است، بنابراین بین دو سیاره داخل یک

کهکشان تنها یک مسیر وجود دارد. برای مثال در شکل زیر فاصله‌ی سیاره‌ی یک تا دو برابر ۴ است و تنها یک مسیر بین

این دو سیاره است.



در صورتی که دو سیاره در دو کهکشان متفاوت باشند فاصله‌ی آن‌ها برابر فاصله‌ی اولی تا گیت خروجی کهکشان مبدأ به

علاوه‌ی فاصله‌ی گیت ورودی کهکشان مقصد تا سیاره‌ی مقصد است (فاصله‌ی بین دو گیت دو کهکشان با سرعت نور

طی می‌شود و صفر در نظر گرفته می‌شود).

۱۷۲ اعتبار افراد و اعتبار سیستم تاکسیاره

مسافر برای پرداخت هزینه‌ی سفرهای خود از روش پرداخت اعتباری استفاده می‌کند، بنابراین هر مسافر مقداری اعتبار

نزد سیستم تاکسیاره دارد. هنگام انجام سفر، هزینه‌ی سفر از اعتبار مسافر کاسته خواهد شد. همچنین مسافر می‌تواند

اعتبار خود را در سیستم افزایش دهد. پرداخت حقوق راننده‌های سیستم نیز از طریق افزایش اعتبار آنها خواهد بود. علاوه

بر این دو مورد، مقدار دارایی کل سیستم تاکسیاره به صورت اعتبار نگه‌داری می‌شود. مقدار پولی که مسافر برای هر سفر

پرداخت می‌کند باعث افزایش اعتبار سیستم تاکسیاره خواهد شد. همچنین حقوق راننده‌ها نیز از اعتبار سیستم تاکسیاره

پرداخت می‌شود. جزئیات نحوه‌ی افزایش اعتبار مسافر و پرداخت حقوق راننده در ادامه توضیح داده خواهد شد.

۱۷۹ وضعیت راننده

هر راننده می‌تواند در سه حالت باشد:

۱. در دسترس

۲. خارج از دسترس

۳. در حال سفر

۱۸۴

هر راننده‌ی خارج از دسترس با اعلام آمادگی برای پذیرش مسافر وارد حالت در دسترس می‌شود. همچنین هر راننده‌ی در

۱۸۵

دسترس می‌تواند اعلام کند که دیگر مسافر نمی‌پذیرد و وارد حالت خارج از دسترس شود. راننده‌ی در دسترس با پذیرش

۱۸۶

مسافر به حالت در حال سفر می‌رود و پس از اتمام آن به حالت در دسترس می‌رود. آدرس راننده پس از اتمام سفر به مکانی

۱۸۷

که در آن سفر تمام شده است تغییر می‌کند.

۱۸۸

```
<driver_username> set_status available <address>
```

۱۸۹

```
<driver_username> set_status unavailable
```

۱۹۰

– توجه کنید که در حالت در دسترس باید امکان وجود آدرس ورودی بررسی شود.

۱۹۱

– راننده می‌تواند از حالت در دسترس در یک آدرس به حالت در دسترس در یک آدرس دیگر برود.

۱۹۲

درخواست سفر

۱۹۳

هر مسافر می‌تواند به سیستم درخواست سفر بدهد. سفر می‌تواند دارای چند مقصد باشد، یعنی باید مسیر سفر به ترتیب

۱۹۴

مقصد اول، مقصد دوم، ... تا مقصد نهایی را شامل شود.

۱۹۵

مسافر می‌تواند هزینه‌ی یک سفر و مسافت آن را با دستور زیر مشاهده کند:

۱۹۶

```
<passenger_username> estimate_trip [VIP] <source_address>
```

۱۹۷

```
<destination0_address> <destination2_address> ... <final_destination_address>
```

۱۹۸

```
-----
```

۱۹۹

```
<trip_cost> <trip_distance>
```

۲۰۰

– محاسبه‌ی هزینه در ادامه توضیح داده می‌شود.

۲۰۱

– مسافت سفر برابر مجموع فاصله‌ی مبدا (مکان اولیه‌ی مسافر) تا مقصد اول، مقصد اول تا مقصد دوم، ... تا

۲۰۲

مقصد نهایی است.

۲۰۳

دستور درخواست سفر به شکل زیر است:

۲۰۴

```
<passenger_username> request_trip [VIP] <source_address> <destination0_address>
```

۲۰۵

```
<destination2_address> ... <final_destination_address>
```

۲۰۶

```
-----
```

۲۰۷

```
<trip_cost> <trip_distance>
```

۲۰۸

– هر مسافر در یک لحظه تنها می‌تواند یک سفر درخواست کند و در صورتی که سفر قبلی آن به اتمام نرسیده یا لغو

۲۰۹

نشده نمی‌تواند درخواست سفر دیگری بدهد.

۲۱۰

– در صورتی که اعتبار مسافر برای این سفر کافی نباشد پیغام مناسبی نمایش داده می‌شود. توجه کنید که اعتبار هر

۲۱۱

مسافر تا ده واحد می‌تواند منفی شود.

۲۱۲

– در صورتی که طبق الگوریتم راننده‌ای برای درخواست بالا یافت نشد پیام مناسب باید نمایش داده شود.

۲۱۳

مسافر می‌تواند تا زمانی که راننده‌ای درخواست او را تایید نکرده است درخواست خود را لغو کند:

۲۱۴

```
<passenger_username> cancel_trip_request
```

۲۱۵

اگر مسافر سفر را لغو نکرده باشد سیستم طبق الگوریتمی که در ادامه توضیح داده می‌شود راننده‌های مناسب این سفر را

۲۱۶

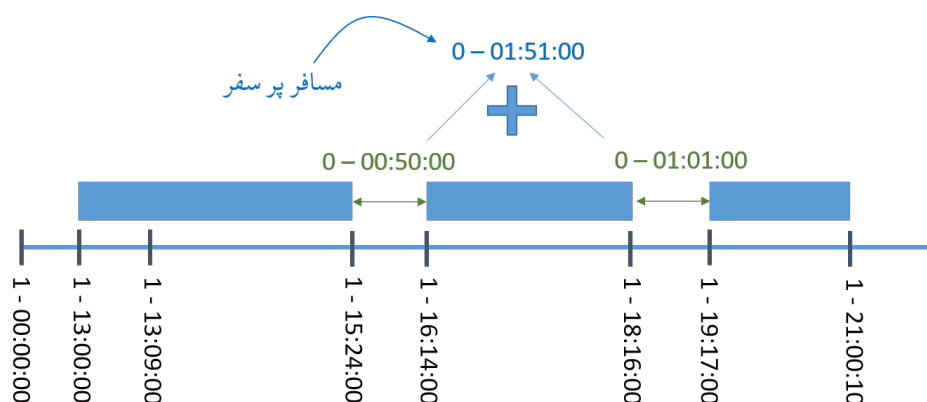
پیدا می‌کند. راننده‌ها می‌توانند درخواست سفرهایی که به آن‌ها ارسال شده است را با دستور زیر مشاهده کنند:

<code><driver_username> show_trip_requests</code>	۲۱۷
-----	۲۱۸
<code>[<passenger_username> <source_address> <destination1_address></code>	۲۱۹
<code><destination2_address> ... <final_destination_address> <trip_cost></code>	۲۲۰
<code><trip_distance>]*</code>	۲۲۱
- در صورتی که راننده VIP باشد تنها سفرهای VIP به او ارسال می‌شود و در غیر این صورت به رانندگان عادی درخواست فرستاده می‌شود.	۲۲۲
- الگوریتم انتخاب راننده‌های مناسب در ادامه توضیح داده می‌شود.	۲۲۳
- راننده به شکل زیر می‌تواند یک درخواست سفر را بپذیرد:	۲۲۴
<code><driver_username> accept_trip_request <passenger_username></code>	۲۲۵
- پس از تایید درخواست، وضعیت راننده به در حال سفر تغییر می‌کند.	۲۲۶
- راننده نمی‌تواند تا اتمام سفر وضعیت خود را تغییر دهد.	۲۲۷
- پس از تایید درخواست، راننده به سمت مسافر حرکت می‌کند.	۲۲۸
- مسافر می‌تواند وضعیت درخواست خود را به شکل زیر مشاهده کند:	۲۲۹
<code><passenger_username> trip_status</code>	۲۳۰
-----	۲۳۱
<code>accepted <driver_username> <driver_address> <spaceship_model> <spaceship_color></code>	۲۳۲
<code>OR</code>	۲۳۳
<code>waiting</code>	۲۳۴
- آدرس راننده در قسمت نتایج بالا آدرس اولیه‌ی راننده وقتی درخواست را تایید کرده‌است می‌باشد.	۲۳۵
- هنگامی که راننده به مسافر می‌رسد دستور زیر را اجرا می‌کند:	۲۳۶
<code><driver_username> arrived</code>	۲۳۷
- در انتهای سفر نیز راننده با دستور زیر خاتمه‌ی سفر را اعلام می‌کند:	۲۳۸
<code><driver_username> end_trip</code>	۲۳۹
- پس از این دستور وضعیت راننده به در دسترس در آدرس مقصد نهایی سفر تغییر می‌کند.	۲۴۰
- پس از اتمام سفر هزینه‌ی مربوطه از اعتبار مسافر کم شده و به اعتبار سیستم اضافه می‌شود.	۲۴۱
- پس از پایان سفر مسافر باید به راننده به شکل زیر از یک تا ده امتیاز بدهد:	۲۴۲
<code><passenger_username> rate_driver <rate_number></code>	۲۴۳
- مسافر حتماً باید در انتهای سفر به راننده امتیاز بدهد و تا وقتی امتیاز ندهد امکان درخواست سفر دیگری ندارد.	۲۴۴
- همانطور که گفته شد rate_number عددی بین ۱ تا ۱۰ است.	۲۴۵
	۲۴۶
	۲۴۷
	۲۴۸

افزایش اعتبار	۲۴۹
مسافر هر زمان که بخواهد می تواند اکانت خود را شارژ کند. هر کاربر می تواند تا ده واحد به سیستم بدهکار شود. اگر	۲۵۰
بدهی کاربر از ده واحد بیشتر شود هنگام درخواست سفر سیستم خطای عدم موجودی می دهد.	۲۵۱
مسافر با دستور زیر می تواند حساب خود را شارژ کند:	۲۵۲
<code><passenger_username> charge_account <amount></code>	۲۵۳
مسافر با دستور زیر نیز می تواند از اعتبار خود مطلع شود:	۲۵۴
<code><passenger_username> get_credit</code>	۲۵۵
-----	۲۵۶
<code><credit></code>	۲۵۷
زمان	۲۵۸
برای راحتی تست برنامه در زمان اجرا برنامه ی شما باید قابلیت دریافت زمان از مدیر سیستم را داشته باشد. مدیر سیستم	۲۵۹
طبق دستور زیر زمان را تعیین می کند:	۲۶۰
<code>admin set_time <date_time></code>	۲۶۱
که date_time به فرمت زیر است:	۲۶۲
<code><date_time> ≡ <day: [1-31]>-<hour: [0-23]>:<minute: [0-59]>:<second: [0-59]></code>	۲۶۳
- مدیر سیستم همواره زمان را به صورت صعودی تغییر می دهد. یعنی امکان ندارد زمان در فرمت بالا (روز،	۲۶۴
ساعت، دقیقه و ثانیه) به عقب برگردد.	۲۶۵
- زمان ابتدایی سیستم برابر 00:00:00 و روز اول می باشد.	۲۶۶
- برای مثال 14:23:13 یک date_time است.	۲۶۷
- دقت داشته باشید که زمان به صورت خودکار جلو نمی رود. زمان تنها در صورتی تغییر می کند که مدیر سیستم	۲۶۸
دستور set_time را وارد نماید.	۲۶۹
الگوریتم پیدا کردن راننده	۲۷۰
بعد از درخواست سفر، سیستم حداکثر چهار راننده از رانندگان در دسترس حاضر در کهکشان مبدأ را بر اساس فاصله از	۲۷۱
سیاره ی مسافر و امتیاز راننده انتخاب می کند. هر چه فاصله کمتر باشد و امتیاز بیشتر باشد احتمال انتخاب راننده بیشتر	۲۷۲
است (میزان تاثیر هر یک بر عهده ی خود شماست). توجه نمایید که در صورت VIP بودن سفر الگوریتم بالا فقط برای	۲۷۳
رانندگان VIP تکرار می شود و در غیر اینصورت فقط برای رانندگان عادی انجام می شود.	۲۷۴
محاسبه ی هزینه ی سفر	۲۷۵
هزینه ی سفر برابر با مجموع فاصله ی مبدأ تا مقصد اول، مقصد اول تا مقصد دوم، ... تا مقصد نهایی است. در صورتی	۲۷۶
که سفر VIP باشد این هزینه دوبرابر می شود.	۲۷۷
نکته: در ادامه مفهوم مسافر پرسفر مطرح می شود که در هزینه ی سفر تاثیر گذار است.	۲۷۸

۲۷۹ مسافر پرسفر!

- ۲۸۰ بعد از مدتی شرکت دانش بنیان تاکسیاره تصمیم می گیرد برای افزایش سفرها مفهومی تحت عنوان “مسافر پرسفر” را
- ۲۸۱ تعریف کند. یک مسافر می تواند در وضعیت پرسفر قرار بگیرد یا از آن خارج شود. تمام هزینه های مسافرهای پرسفر با
- ۲۸۲ ضریب یک دوم محاسبه می شود.
- ۲۸۳ یک مسافر در حالت پرسفر قرار می گیرد اگر حداقل سه سفر داشته باشد و مجموع فاصله ی بین سه سفر اخیرش کمتر از
- ۲۸۴ ۱ روز باشد.



- ۲۸۵
- ۲۸۶ یک مسافر که پرسفر است از این وضعیت خارج می شود اگر فاصله ی زمانی آن تا آخرین سفرش بیشتر از دو روز باشد.
- ۲۸۷ اگر هر دو شرط بالا برای یک نفر برقرار بود هنگام ورود به وضعیت پرسفر بودن شرط اولی اولویت دارد و هنگامی که
- ۲۸۸ داخل این وضعیت است شرط دوم اولویت دارد.
- ۲۸۹ وقتی یک مسافر در وضعیت پرسفر است هزینه ی سفرهایش ضریب 0.5 (مقدار صحیح آن) می شود.

۲۹۰ پرداخت حقوق رانندگان

- ۲۹۱ در انتهای هر روز سیستم ۸۰ درصد از هزینه ی سفرهایی که در آن روز به اتمام رسیده است را به حساب راننده ی مربوطه
- ۲۹۲ می ریزد. از آنجایی که زمان در این سیستم گسسته است و از مدیر سیستم دریافت می شود می توان این واریز مبلغ را هنگام
- ۲۹۳ تغییر زمان انجام داد. برای مثال اگر زمان سیستم 13:01:02 روز 1 باشد با تغییر آن به 01:01:01 روز 3 باید ۸۰ درصد
- ۲۹۴ از هزینه های سفرهایی که تا زمان 13:01:02 روز 1 به پایان رسیده است، به حساب راننده های مربوطه ریخته شود (توجه
- ۲۹۵ کنید که به واسطه ی پرش از روز دوم در تغییر زمان عملاً در روز دوم هیچ سفری انجام نشده است). تا قبل از تسویه مبالغ
- ۲۹۶ دریافت شده جزء اعتبار سیستم هستند و پس از تسویه عملاً از اعتبار سیستم کم شده و به حساب راننده ریخته می شوند.
- ۲۹۷ توجه کنید از آنجایی که ممکن است به واسطه ی تخفیف، مسافر هزینه ای پرداخت نکرده باشد امکان دارد اعتبار سیستم
- ۲۹۸ پس از پرداخت هزینه به راننده منفی شود.

۲۹۹ بازرس

- ۳۰۰ بعد از چند ماه، آژانس های مسافرتی قدیمی که درآمدشان کاهش یافته بود سامانه ی جابه جایی میان سیاره ای تاکسیاره را به
- ۳۰۱ تقلب و قاچاق متهم کردند. قرار شد سنا یک بازرس معتمد بفرستد تا از قانونی بودن فعالیت های سامانه ی جابه جایی
- ۳۰۲ مطمئن شود.

بازرس می‌تواند از مدیر سیستم (admin) بخواهد که تعداد سفرها و درآمد خالصی را که یک راننده در یک بازه‌ی زمانی خاص از مشتریان کسب کرده، امتیازاتی که یک راننده در یک بازه‌ی زمانی خاص به‌دست آورده و موجودی حساب یک مسافر خاص را نمایش دهد. همچنین بازرس باید اطلاعاتی همانند اعتبار کل سیستم، تعداد سفرهای کل سیستم، تعداد راننده‌ها، تعداد مسافران در سیستم (که در بازه‌ی مربوطه یا به طور کلی ثبت‌نام کرده‌اند) را به سنا بفرستد.

مدیر سیستم برای بازرس مربوطه می‌تواند دستورات را به شکل زیر وارد نماید:

۳۰۳
۳۰۴
۳۰۵
۳۰۶
۳۰۷
۳۰۸
۳۰۹
۳۱۰
۳۱۱
۳۱۲
۳۱۳
۳۱۴
۳۱۵
۳۱۶
۳۱۷
۳۱۸
۳۱۹
۳۲۰
۳۲۱
۳۲۲
۳۲۳
۳۲۴
۳۲۵
۳۲۶
۳۲۷
۳۲۸
۳۲۹
۳۳۰
۳۳۱
۳۳۲
۳۳۳

```
admin driver_report <driver_username> <from_date_time> <to_date_time>
-----
<trips_count> <total_income> <total_rate>
```

```
admin passenger_report <passenger_username>
-----
<credit>
```

```
admin system_report
OR
admin system_report <from_date_time> <to_date_time>
-----
<system_credit> <trips_count> <drivers_count> <passengers_count>
```

نحوه‌ی تحویل	۳۳۴
شما باید فایل‌های h و .cpp و Makefile خود را در یک پوشه با عنوان A7-SID قرار داده و آن را با فرمت zip	۳۳۵
آرشیو کنید و در نهایت فایل‌ی با نام A7-SID.zip را در سایت درس آپلود کنید. (SID پنج رقم آخر شماره‌ی دانشجویی	۳۳۶
شماست. به عنوان مثال اگر شماره‌ی دانشجویی شما 810195422 است، نام فایل شما باید A7-95422.zip باشد.)	۳۳۷
تحویل این تمرین به صورت حضوری است و در هنگام تحویل باید به تمام قسمت‌های کد خود مسلط باشید.	۳۳۸
دقت کنید	۳۳۹
• هدف اصلی این تمرین آشنایی با برنامه‌نویسی شیء‌گرا است. سعی کنید ابتدا طراحی مناسبی برای کلاس‌های خود بیابید و سپس کدزدن را آغاز کنید.	۳۴۰ ۳۴۱
• به علت زیاد بودن تعداد دستورات، به طراحی مناسب برای مدیریت آن‌ها حتماً فکر کنید. مقداری از نمره‌ی شما متعلق به طراحی شما برای مدیریت دستورات است.	۳۴۲ ۳۴۳
• رعایت تمامی نکات مطرح‌شده در کلاس درس همانند:	۳۴۴
– عدم استفاده از متغیرهای جهانی ⁷	۳۴۵
– عدم استفاده از goto	۳۴۶
– استفاده از یک رویه مشخص و ثابت برای نامگذاری متغیرها، ساختارها، ثابت‌ها، توابع و فایل‌ها	۳۴۷
(استفاده از Underscore به عنوان جداکننده یا قاعده‌ی CamelCase یا ...)	۳۴۸
– نامگذاری مناسب و بامعنی متغیرها، ساختارها، ثابت‌ها، توابع و فایل‌ها	۳۴۹
– ...	۳۵۰
اجباری است.	۳۵۱
• پروژه‌ی شما باید حتماً شامل Makefile باشد.	۳۵۲
• برنامه‌ی شما باید در سیستم عامل لینوکس نوشته و با کامپایلر g++ کامپایل شود.	۳۵۳
• به فرمت و نام فایل‌های خود دقت کنید. در صورتی که هر یک از موارد گفته شده رعایت نشود، مقدار قابل توجهی از نمره‌ی شما کسر می‌گردد.	۳۵۴ ۳۵۵
• در صورت کشف تقلب در کل و یا قسمتی از تمرین، برای هر دو طرف نمره‌ی 100- منظور خواهد شد.	۳۵۶

⁷ Global