



برنامه سازی پیشرفته

تمرین شماره ۷ - فاز ۲

مدرس: رامتین خسروی

طراحان: زهرا یزدانی، طاهای فخریان، پرنیان فاضل

ادیب رضایی، سروش میرزاسروری، امین ستایش

مهلت تحویل: دوشنبه ۱۳ دی، ساعت ۲۳:۵۵

Smart UTRAVEL!

مقدمه

در فاز اول شما به پیاده سازی منطق برنامه و استفاده از رابط خط فرمان^۱ برای ارتباط با آن پرداختید. در این فاز نیز تمامی تعاملات با کاربر و سیستم همچنان از طریق خط فرمان انجام خواهد شد و رابط کاربری گرافیکی^۲ مختص فاز آخر خواهد بود. هدف از این فاز این است که امکاناتی جهت هوشمند کردن برنامه و گرفتن گزارش به آن اضافه می کنیم.

در این فاز پیاده سازی قابلیت هایی نظیر پیشنهاد دادن بهترین مسیر از نظر قیمت، مسافت، زمان انتظار و ... با حداکثر یک بار توقف در مسیر و گزارش دادن روی داده ها می پردازید. همچنین در این فاز به پیاده سازی یک گزارش از اطلاعات کلی پروازها و بلیت های خریداری شده می پردازید.

^۱ Command Line Interface (CLI)

^۲ Graphical User Interface (GUI)

شرح تمرین

در این فاز به پیاده‌سازی یک روش پیشنهاد دادن پروازها به افراد بر اساس قیمت میپردازیم و چند دستور جدید به برنامه اضافه می‌کنیم تا بتوانیم از قابلیت‌های اضافه شده استفاده کنیم. همچنین در بخش گزارش اطلاعات کلی از پروازها یا خریدها را محاسبه خواهیم کرد.

تمام نکاتی که برای دستورات در بخش قبل ذکر شده بود در این فاز نیز برقرار است و بایستی در برنامه به درستی بررسی شوند. توضیح پاسخ‌ها و خطاها مجدداً در انتهای صورت پروژه ذکر شده است که انتظار می‌رود به درستی بررسی شوند و پیغام مناسب را چاپ کنید.

خروجی پروژه شما مانند فاز قبل به صورت خودکار آزموده می‌شود، لذا خروجی شما باید دقیقاً همانند خروجی خواسته‌شده باشد. در غیر این صورت نمره‌ی بخش آزمون را از دست خواهید داد.

● پاسخ دستورات

به ازای هر دستوری که اقدام به اجرای آن می‌کنیم، پاسخی از سمت سیستم دریافت می‌کنیم. این پاسخ می‌تواند اطلاعاتی باشد که از سیستم خواسته شده است مانند اطلاعات یک پرواز یا لیست تمام پروازها. اما گونه‌های دیگری از پاسخ نیز وجود دارد.

○ پاسخ درخواست موفقیت‌آمیز

اگر دستوری که کاربر وارد می‌کند (قسمت پس از POST یا GET یا DELETE)، به درستی انجام پذیرد ولی اطلاعاتی را برنگرداند. این پاسخ برای کاربر نمایش داده می‌شود.

خروجی

OK

○ پاسخ خالی بودن

اگر دسترسی که کاربر وارد می کند برای نمایش لیستی باشد که در آن لیست هیچ داده ای وجود ندارد، این خروجی نمایش داده می شود. در ادامه مواقعی که نیاز به نمایش این پاسخ وجود دارد توضیح داده شده است.

خروجی
EMPTY

○ پاسخ عدم وجود

اگر دسترسی که کاربر وارد می کند (قسمت پس از POST یا GET یا DELETE)، در لیست دستورهای تعریف شده نباشد یا کاربر با وارد کردن شناسه ای ناموجود در سیستم قصد دسترسی به موجودیتی را داشته باشد، این پاسخ در خروجی چاپ می شود.

خروجی
Not Found

○ پاسخ درخواست اشتباه

اگر اولین قسمت خط ورودی کاربر یکی از سه مقدار POST، GET، DELETE نباشد یا این دستور اطلاعات کافی برای اجرا را در خود نداشته باشد، سیستم خروجی زیر را چاپ می کند.

خروجی
Bad Request

○ پاسخ عدم دسترسی

اگر کاربری به سامانه وارد نشده باشد و از دستورات غیر از ثبت نام یا ورود استفاده کند، سیستم باید خروجی زیر را به کاربر نشان دهد.

خروجی
Permission Denied

● مشاهده پروازهای غیر مستقیم

کاربر می‌تواند با وارد کردن مبدا و مقصد سفر خود با دستور زیر، لیست پروازهای غیر مستقیم با ۱ توقف را از مبدا به مقصد مورد نظر خود را مشاهده کند. در خروجی ابتدا اطلاعات پرواز اول، در خط دوم اطلاعات پرواز دوم و در خط سوم اطلاعاتی درباره طول زمان توقف و کل هزینه سفر که مجموع هزینه پرواز اول و دوم می‌باشد، نشان داده می‌شود. پروازهای غیر مستقیم را به ترتیب هزینه کل و به صورت صعودی چاپ کنید. هر سفر غیر مستقیم را با استفاده از ۱۰ کاراکتر خط فاصله (-) از هم جدا کنید. دقت کنید که زمان‌های داده شده برای هر پرواز در فایل اطلاعات ورودی به زمان محلی همان شهر است، بنابراین برای به دست آوردن طول زمان توقف شما لازم نیست زمان‌های حرکت و نشست پروازها را تبدیل کنید و اختلاف زمان حرکت پرواز دوم و نشست پرواز اول در یک شهر، طول زمان توقف را به شما می‌دهد. در اینجا شما فقط پروازهایی را به عنوان پرواز غیر مستقیم نشان می‌دهید که حداکثر ۱۵ ساعت زمان توقف داشته باشند. طول زمان پرواز را به فرمت h m نشان دهید. به جدول زیر توجه کنید.

ورودی

```
GET connecting_flights ? from <city> to <city>
```

خروجی

```
-----  
Flight 1: <flight_id> <airline> <origin> <destination> <departure_date>  
<departure_time> <arrival_date> <arrival_time> <available_seats> <cost>  
Flight 2: <flight_id> <airline> <origin> <destination> <departure_date>  
<departure_time> <arrival_date> <arrival_time> <available_seats> <cost>  
* Connection duration: <connection_duration>, Total cost: <total_cost>  
-----
```

| Not Found | Bad Request | Permission Denied

ورودی نمونه

```
GET connecting_flights ? from Tehran to London
```

خروجی نمونه

```
Flight 1: 2 Turkish Tehran Istanbul 24 02:05 24 05:05 25 112.00  
Flight 2: 5 Turkish Istanbul London 24 08:20 24 09:25 34 150.00  
* Connection duration: 3h 15m, Total cost: 262.00  
-----  
Flight 1: 11 Emirates Tehran Dubai 21 11:05 21 13:40 7 109.00  
Flight 2: 3 Emirates Dubai London 21 16:05 21 20:00 34 158.00  
* Connection duration: 2h 25m, Total cost: 267.00  
-----
```

● مشاهده به صرفه‌ترین پرواز

کاربر می‌تواند با وارد کردن مبدا، مقصد و تاریخ حرکت پرواز مورد نظر خود، کم‌هزینه‌ترین پرواز موجود را ببیند. در این قسمت شما باید کم‌هزینه‌ترین پرواز را از بین پروازهای مستقیم و غیر مستقیم با یک توقف پیدا کنید. در صورتی که کم‌هزینه‌ترین پرواز، پرواز مستقیم بود، اطلاعات آن پرواز را در یک خط چاپ کنید و اگر پرواز غیر مستقیم بود، اطلاعات پرواز اول را در خط اول و اطلاعات پرواز دوم را در خط دوم چاپ کنید. در خط آخر خروجی، هزینه نهایی سفر را به ازای یک مسافر نشان دهید. هزینه نهایی برای پرواز

مستقیم، هزینه پرواز و برای پرواز غیر مستقیم مجموع هزینه پرواز اول و پرواز دوم است. دقت داشته باشید در صورتی که هیچ پروازی بین دو شهر مشخص شده موجود نباشد Not Found نشان داده می‌شود. به جدول زیر توجه کنید.

ورودی
GET cheapest_flight ? from <city> to <city> departure_date <departure_date>
خروجی
<flight_id> <airline> <origin> <destination> <departure_date> <departure_time> <arrival_date> <arrival_time> <available_seats> <cost> Total cost: <total_cost> Not Found Bad Request Permission Denied
ورودی نمونه اول
GET cheapest_flight ? from Tehran to London departure_date 24
خروجی نمونه اول
2 Turkish Tehran Istanbul 24 02:05 24 05:05 25 112.00 5 Turkish Istanbul London 24 08:20 24 09:25 34 150.00 Total cost: 267.00
ورودی نمونه دوم
GET cheapest_flight ? from Mashhad to Ahvaz departure_date 9
خروجی نمونه دوم
1 IranAir Mashhad Ahvaz 9 15:00 9 17:00 100 34.00 Total cost: 34.00

● مشاهده‌ی گزارش

در این بخش می‌خواهیم در یک گزارش کلی برخی از اطلاعات را راجع به پروازها و بلیت‌ها به کاربر نمایش دهیم.

در ادامه توضیحات هر کدام از بخش‌های گزارش و در آخر نحوه‌ی چاپ شدن گزارش ذکر شده است.

● `average_flight_cost`: این بخش در گزارش نشان‌دهنده‌ی میانگین قیمت تمام پروازها است و به صورت اعشاری نمایش داده می‌شود.

● `min_flight_cost`: نشان‌دهنده‌ی کمترین قیمت بین تمام پروازها است.

● `max_flight_cost`: نشان‌دهنده‌ی بیشترین قیمت بین تمام پروازها است.

● `most_popular_destination`: نشان‌دهنده‌ی مقصدی است که بیشترین خرید برای آن انجام

شده است. به این صورت محاسبه می‌شود که بین بلیت‌های خریداری شده در بین تمامی کاربرها چه مقصدی بیشتر خریداری شده است. برای محاسبه‌ی این موضوع باید تعداد خرید را نیز در نظر گرفت، به این صورت که برای مثال اگر فردی در خرید بلیت فیلد `quantity` را ۴ ثبت کرده است باید در نظر گرفت که ۴ بار این مقصد را خریده است.

● `top_airlines`: در این بخش بایستی نام ۳ شرکت هواپیمایی که بیشترین فروش را داشته‌اند نمایش

داده شود. توجه داشته باشید که مانند بخش قبل در این بخش نیز باید `quantity` در محاسبه

بیشترین تعداد فروش لحاظ شود. نام این شرکت‌های هواپیمایی باید به ترتیب و با یک فاصله چاپ شود.

در ادامه یک نمونه ورودی و خروجی برای گزارش ذکر شده است:

ورودی

GET `overall_report`

خروجی
<pre>average_flight_cost: <average_flight_cost> min_flight_cost: <min_flight_cost> max_flight_cost: <max_flight_cost> most_popular_destination: <most_popular_destination> top_airlines: <airline1> <airline2> <airline3> Bad Request Permission Denied</pre>
ورودی نمونه
GET overall_report
خروجی نمونه
<pre>average_flight_cost: 156.53 min_flight_cost: 25.00 max_flight_cost: 2000.00 most_popular_destination: paris top_airlines: singapore_airways emirates pegasus</pre>

نحوه‌ی تحویل و نکات پایانی

- تمام فایل‌های خود را در قالب یک پرونده‌ی زیپ با نام A7-<SID>.zip در صفحه‌ی Elearn درس بارگذاری کنید که

SID شماره‌ی دانشجویی شماست؛ برای مثال اگر شماره‌ی دانشجویی شما ۸۱۰۱۹۹۹۹۹ است، نام پرونده‌ی شما باید

A7-810199999.zip باشد.

○ برای مثال، نمونه فایل مورد قبول در زیر آمده است:

A7-810199999.zip

└─ main.cpp

└─ makefile

... _

- از آن جایی که در فازهای بعدی شما باید رابط کاربری برنامه‌ی خود را از خط فرمان به روش‌هایی دیگر تغییر دهید بهتر است تا طراحی برنامه‌ی شما طوری باشد که کمترین وابستگی میان منطق برنامه و رابط کاربری آن وجود داشته باشد.
- توجه داشته باشید که حالت‌های خاصی که در صورت پروژه ذکر نشده است در تست‌های خودکار نخواهد بود و هر طور خواستید می‌توانید آن‌ها را مدیریت کنید.
- **دقت کنید** که پرونده زیپ آپلودی شما باید پس از Unzip شدن شامل پرونده‌های پروژه شما (از جمله Makefile) باشد و از زیپ کردن پوشه‌ای که داخل آن فایل‌های پروژه‌تان قرار دارد خودداری فرمایید.
- برنامه‌ی شما باید در سیستم عامل لینوکس و با مترجم ++g با استاندارد ++11c ترجمه و در زمان معقول برای ورودی‌های آزمون اجرا شود.
- **دقت کنید** که پروژه شما باید Multi-file باشد و Makefile داشته باشد. همین‌طور در Makefile خود مشخص کنید که از استاندارد ++11c استفاده می‌کنید.
- درستی برنامه‌ی شما از طریق **آزمون‌های خودکار** سنجیده می‌شود؛ بنابراین پیشنهاد می‌شود با استفاده از ابزارهایی مانند diff خروجی برنامه خود را با خروجی‌هایی که در اختیارتان قرار داده شده است مطابقت دهید.
- **دقت کنید** که نام پرونده‌ی اجرایی شما باید **utavel.out** باشد و کنار Makefile شما ساخته شود.
- دقت کنید که تمام اعداد غیر صحیح که در خروجی چاپ می‌شوند باید **دقت 2 رقم اعشار بدون گرد کردن** داشته باشند. (مثلا عدد 2.1 باید 2.10 و عدد 4.1234 باید 4.12 نمایش داده شود).
- طراحی درست، رعایت سبک برنامه نویسی درست و تمیز بودن کد برنامه‌ی شما در نمره‌ی تمرین تأثیر زیادی دارد.

- هدف این تمرین یادگیری شماسست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.

- سوالات خود را تا حد ممکن در فروم درس مطرح کنید تا سایر دانشجویان نیز از پاسخ آن‌ها بهره‌مند شوند. در صورتی که قصد مطرح کردن سوال خاص‌تری داشتید، از طریق ایمیل با طراحان این فاز پروژه ارتباط برقرار کنید.