



UTRAVEL!

مقدمه

این پروژه به جمع بندی آموخته‌های شما در درس می‌پردازد. انتظار می‌رود مهارت‌هایی را که در تمرین‌های پیشین و سایر بخش‌های درس آموخته‌اید، در پیاده‌سازی این پروژه به کار گیرید.

در نهایت شما باید سایت رزرو هواپیمایی را طراحی کنید، که دارای امکاناتی مانند رزرو، ایجاد و مشاهده پروازها خواهد بود.

ابتدا اطلاعات پروازها از فایل خوانده خواهند شد و این پروازها لیست کلی پروازهای موجود را تشکیل خواهد داد. کاربران مختلف می‌توانند پروازها را رزرو کنند، بلیت‌ها را مشاهده کنند، و یا لیست پروازها را طبق خواسته‌ی خودشان فیلتر کرده و مشاهده کنند.

در فاز اول این پروژه شما منطق برنامه‌ی این سایت را پیاده‌سازی می‌کنید که با استفاده از رابط کاربری خط فرمان¹ می‌توان از آن استفاده کرد.

نکته قابل توجه در این پروژه این است که بهتر است پروژه به صورت قسمت به قسمت پیاده‌سازی و تست شود. به طوری که ابتدا یک ساختار کلی از پروژه پیاده‌سازی شود و سپس دستورات مختلف به آن اضافه گردد.

¹ CLI (Command Line Interface)

شرح تمرین

در این فاز، منطق برنامه در قالب تعدادی دستور در ادامه توضیح داده شده است. روند استفاده از برنامه به این شکل است که کاربر در برنامه شما با استفاده از رابط خط فرمان، دستوری همراه با آرگومان‌های لازم برای اجرای آن در ورودی استاندارد وارد می‌کند. به عنوان مثال برای گرفتن یک لیست از اطلاعات موجودیت‌ها از دستور GET و در صورت گرفتن یک عنصر خاص از موجودیت مورد نظر بعد از دستور، id موجودیتی که مایل به گرفتن اطلاعات آن است را نیز وارد می‌کند تا دستور مورد نظرش اجرا شود.

همینطور در نظر داشته باشید که این اطلاعات یعنی آرگومان‌های هر دستور پس از علامت ؟ در دستور می‌آیند و همینطور ترتیب خاصی برای آن‌ها وجود ندارد به این معنا که لزومی ندارد آرگومان‌ها به همان ترتیبی که در توضیحات هر دستور گفته شده، وارد شوند. برای مثال می‌توانید در نظر بگیرید که رمز عبور پس از شناسه کاربر برای ورود، وارد شود. بعضی از دستورات نیز آرگومان ورودی ندارند و تنها خود دستور ورودی داده می‌شود.

نکته دیگری که دستورات وجود دارد این است که قبل از وارد شدن دستور، عبارت GET و POST و DELETE وارد می‌شود که به این شکل دستورات به نحوی از هم جدا می‌شوند به این شکل که دستوراتی که بیشتر برای گرفتن اطلاعات از سیستم استفاده می‌شوند، از GET و دستوراتی که بیشتر برای فرستادن اطلاعات به سیستم استفاده می‌شوند، از POST استفاده می‌کنند، همچنین دستوراتی که قصد حذف یک منبع را دارند از DELETE استفاده می‌کنند. این نام‌گذاری دستورها در فازهای آتی پروژه که برنامه خود را روی وب عرضه خواهید کرد معنای خاص پیدا خواهند کرد.

همینطور دقت داشته باشید که تمامی دستورها پس از اجرا شدن دارای خروجی مشخص هستند که منحصراً ذکر می‌شود.

دقت داشته باشید با توجه به تمامی نکات ذکر شده اعم از عبارت قبل از دستورات و یا آرگومان‌های داده شده به دستورات، اگر در دستورات وارد شده کاربر، خطایی وجود داشته باشد باید با توجه به توضیحاتی که همراه با هر دستور آمده است، خطای آن را خروجی دهید. به عنوان مثالی از خطاهایی که ممکن است رخ دهد، می‌توانید وارد کردن دستور خاصی از طرف کاربری که وارد سیستم نشده را نیز در نظر بگیرید.

خروجی پروژه شما به صورت خودکار آزموده می‌شود، لذا خروجی شما باید دقیقاً همانند خروجی خواسته شده 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 دستوری 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 دستوری 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 ی 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

اطلاعات پرواز

فایل اطلاعات پروازها در اختیارتان قرار گرفته است که شامل تعدادی پرواز است که توضیح اطلاعات هر پرواز در ادامه آمده است.

هر پرواز شامل اطلاعات زیر است:

- airline نام شرکت هواپیمایی است.
- origin مبدا پرواز است.
- destination مقصد پرواز است.
- departure_date تاریخ حرکت که عددی در بازه ی ۱ تا ۳۰ است که نشان دهنده ی روز حرکت پرواز است.

- `departure_time` زمان شروع پرواز است که به فرمت hh:mm داده می‌شود.
 - `arrival_date` تاریخ نشستن که عددی در بازه‌ی ۱ تا ۳۰ است که نشان‌دهنده‌ی روز نشستن پرواز است.
 - `arrival_time` زمان نشستن پرواز است که به فرمت hh:mm داده می‌شود.
 - `seats` ظرفیت کل پرواز است.
 - `cost` قیمت پرواز به ازای یک نفر در درجه^۲ اقتصادی^۳ است، این عدد به صورت اعشاری در فایل داده شده است. دقت داشته باشید که در طول پروژه همه‌ی اعداد غیرصحیح با دقت دو رقم اعشار و بدون گردکردن چاپ می‌شوند.
- * توجه شود که شما باید در هنگام خواندن اطلاعات پروازها از فایل CSV، برای هر پرواز یک شناسه یکتا در نظر بگیرید. در فایل CSV اطلاعات هر پرواز در یک خط جداگانه آمده است. شناسه‌هایی که به این پروازها اختصاص می‌دهید را از عدد ۱ شروع کنید. یعنی به اولین پروازی که در فایل آمده، شناسه ۱ و به دومین پرواز شناسه ۲ و ... را اختصاص دهید.
- * ۲۵٪ ظرفیت کل پرواز مربوط درجه تجاری و بقیه ظرفیت مربوط به درجه اقتصادی هستند.
- * همانطور که گفته شد، قیمتی که برای هر پرواز در فایل CSV ثبت شده، قیمت آن پرواز در درجه اقتصادی است. قیمت همان پرواز در درجه تجاری را ۲۰۵ برابر قیمت پرواز در کلاس اقتصادی در نظر بگیرید.
- * `arrival_time` و `departure_time` به زمان محلی هستند و نیازی به تبدیل زمان‌ها نیست.

● وارد کردن اطلاعات پروازها

در ابتدای شروع برنامه، باید اطلاعات پروازها از یک فایل CSV که نام آن به عنوان آرگومان خط فرمان به برنامه‌ی شما داده می‌شود، خوانده شده و در حافظه‌ی برنامه ذخیره شود. دستور آغاز برنامه به شکل زیر خواهد بود.

توجه کنید نام فایل اجرایی برنامه شما باید `uttravel.out` باشد.

^۲ Class

^۳ Economy

ورودی

```
./uttravel.out <csv_file_relative_address>
```

ورودی نمونه

```
./uttravel.out folder1/folder2/uttravel.csv
```

دستورات

● احراز هویت

○ ثبت نام

با این دستور افراد می‌توانند در سامانه به عنوان کاربر ثبت نام کنند که در این صورت پس از وارد کردن این دستور وارد سیستم نیز خواهند شد. نام کاربری هر کاربر یک رشته یکتا در سیستم است. اگر نام کاربری در سیستم موجود باشد درخواست معتبر نیست و خروجی Bad Request چاپ می‌شود. تضمین می‌شود اطلاعات ورودی، درون خودشان کاراکتر فاصله ندارند.

ورودی

```
POST signup ? username <username> password <password>
```

خروجی

```
OK | Bad Request
```

○ ورود

اگر کاربری قبلاً در سامانه ثبت نام کرده باشد، پیش از استفاده از امکانات آن باید وارد سیستم شود. توجه کنید که کاربران بلافاصله بعد از ثبت نام، نیازی به دستور ورود ندارند و پس از ثبت

نام، وارد سیستم نیز می‌شوند. در صورت درست نبودن اطلاعاتی که کاربر وارد کرده است پیام Bad Request در خروجی چاپ شود.

ورودی	خروجی
POST login ? username <username> password <password>	OK Bad Request

○ خروج

اگر شخصی قبلاً در سیستم وارد شده بود با وارد کردن این دستور از سیستم خارج می‌شود و می‌تواند با یک حساب کاربری دیگر وارد سامانه شود. اگر کاربری قبل از ورود به سیستم این دستور را وارد کرد، پیام Permission Denied باید خروجی داده شود.

ورودی	خروجی
POST logout	OK Permission Denied

● افزایش اعتبار کیف پول

کاربر می‌تواند به مقداری که وارد می‌کند (عدد اعشاری بزرگتر از صفر)، حساب مالی خود را شارژ کند. اگر مقدار داده شده کوچکتر یا مساوی صفر بود باید خطای Bad Request داده شود، همینطور در صورتی که کاربر وارد حساب کاربری خود نشده بود با وارد کردن این دستور پیغام Permission Denied خواهد گرفت.

ورودی	خروجی
POST wallet ? amount <amount>	OK Bad Request Permission Denied

• نمایش پروازها

کاربر می‌تواند با دستور زیر لیست پروازهایی که حداقل یک ظرفیت خالی دارند را به همراه اطلاعات خلاصه‌شده‌ی آن‌ها مشاهده کند. اطلاعات هر پرواز در یک خط نمایش داده می‌شود.

دقت داشته باشید در صورتی که کاربر وارد حساب کاربری خود نشده بود پیام `Permission Denied` و همین‌طور در صورتی که لیست پروازها خالی بود عبارت `Empty` را خروجی دهید.

برای این که در آزمون خودکار برنامه شما مشکلی ایجاد نشود، می‌توانید فرض کنید ترتیب چاپ شدن پروازها همواره به ترتیب شناسه‌ی آن‌ها خواهد بود.

در بخش‌های بعدی فیلترها تعریف خواهند شد که اگر کاربر فیلترهایی را اضافه کرده باشد، این لیست باید طبق آن فیلترها، ابتدا پالایش شود و سپس نتیجه نمایش داده شود. در صورتی که کاربر هیچ فیلتری ثبت نکرده باشد، لیست تمامی پروازهایی که حداقل یک ظرفیت خالی دارند، نمایش داده خواهد شد.

ورودی
GET flights
خروجی
<flight_id> <airline> <origin> <destination> <departure_date> <departure_time> <arrival_date> <arrival_time> <available_seats> <cost> Empty Permission Denied
ورودی نمونه
GET flights
خروجی نمونه
1 IranAir Mashhad Ahvaz 9 15:00 9 17:00 100 70.00 2 Pegasus Istanbul London 30 10:45 30 11:50 25 390.25

● نمایش مشخصات یک پرواز

کاربر می‌تواند با وارد کردن شناسه‌ی یک پرواز، اطلاعات کامل آن را دریافت کند. همینطور در صورتی که شناسه ورودی کاربر موجود نبود و یا به عبارتی قابل قبول نبود نیز خروجی Not Found دهید. دقت داشته باشید برای حالتی که کاربر وارد حساب کاربری خود نشده باشد نیز همانند حالات گذشته خروجی Permission Denied دهید.

ورودی
GET flights ? id <id>
خروجی
<flight_id> <airline> <origin> <destination> <departure_date> <departure_time> <arrival_date> <arrival_time> <available_seats> <cost> Not Found Bad Request Permission Denied
ورودی نمونه
GET flights ? id 2
خروجی نمونه
2 Pegasus Istanbul London 30 10:45 30 11:50 25 390.25

● اضافه کردن فیلتر

هر کاربر می‌تواند تعدادی فیلتر برای خودش اضافه کند. فیلترها باعث می‌شوند لیست پروازهایی که با دستور نمایش پروازها دیده خواهد شد، پالایش شوند. انتظار می‌رود این فیلترها را به صورت کلاس پیاده‌سازی کنید و برای اعمال آنها از چندریختی⁴ استفاده کنید. با اضافه کردن فیلتر جدید با نوع تکراری فیلتر قبلی حذف شده و فیلتر جدیدتر روی لیست اعمال میشود.

⁴ polymorphism

دقت داشته باشید فیلترهای اعمال شده کاربر با خروج از سیستم (logout) از بین می‌روند و با ورود دوباره کاربر به سیستم لیست پروازها بدون اعمال هیچ گونه فیلتری نمایش داده خواهند شد.

○ فیلتر مبدا و مقصد پرواز

این فیلتر فقط پروازهایی که مبدا و مقصدشان در دستور مشخص شده است را تایید می‌کند.

ورودی	خروجی
POST filters ? from <city> to <city>	OK Bad Request
ورودی نمونه	خروجی نمونه
POST filters ? from Tehran to Shiraz	OK

○ فیلتر بازه‌ی هزینه‌ی پرواز

این فیلتر، فقط پروازهایی که هزینه‌ی پروازشان در بازه‌ی خاصی است را تایید می‌کند. اگر هم max_price و min_price داده شده بود، پروازها با قیمتی بین min_price و max_price، در صورتی که فقط min_price داده شده بود، تمامی پروازها با قیمت بیشتر یا مساوی min_price و اگر فقط max_price داده شده بود، تمامی پروازها با قیمت کمتر یا مساوی max_price تایید می‌شوند.

اگر دستور شامل هیچ یک از ۳ حالت بالا نبود، اعداد داده شده کمتر از صفر بودند، یا عدد پایان بازه از عدد شروع بازه کمتر بود باید خطای Bad Request نمایش داده شود.

ورودی	خروجی
POST filters ? min_price <min_price> max_price <max_price>	OK Bad Request

ورودی نمونه	خروجی نمونه
POST filters ? min_price 255.20 max_price 300.90	OK

○ فیلتر شرکت هواپیمایی پرواز

این فیلتر فقط پروازهایی که متعلق به یک شرکت هواپیمایی خاص هستند را تایید می کند.

ورودی	خروجی
POST filters ? airline <airline>	OK Bad Request
ورودی نمونه	خروجی نمونه
POST filters ? airline Mahan	OK

○ فیلتر بازه زمان حرکت پرواز

این فیلتر فقط پروازهایی که در یک تاریخ خاص و بازه‌ی زمانی خاصی هستند را تایید می کند. همانطور که گفته شد، بازه‌ی زمانی تاریخ پروازها یک عدد بین ۱ تا ۳۰ خواهد بود. در صورتی که زمان min_departure_time در دستور داده نشود، پروازهایی که ساعت حرکتشان از ساعت ۱۲ بامداد همان تاریخ تا انتهای بازه‌ی زمانی داده شده در دستور است، در صورتی که max_departure_time در دستور داده نشود، پروازهایی که ساعت حرکتشان از ابتدای بازه‌ی داده شده تا ساعت ۲۴ تاریخ داده شده در دستور است و اگر هم min_departure_time و max_departure_time داده نشود، تمامی پروازهای موجود در تاریخ داده شده، تایید می شوند. توجه کنید که تاریخ (departure_date) باید حتما در دستور داده شود و در غیر این صورت خطای Bad Request باید چاپ شود.

ورودی	خروجی
-------	-------

POST filters ? departure_date <departure_date> min_departure_time <min_departure_time> max_departure_time <max_departure_time>	OK Bad Request
ورودی نمونه	خروجی نمونه
POST filters ? departure_date 23 min_departure_time 10:00 max_departure_time 18:30	OK

○ حذف تمام فیلترها

کاربر می‌تواند تمام فیلترهایی که اضافه شده است را حذف کند. خروجی برای زمانی که فیلتری وجود داشته یا هیچ فیلتری وجود نداشته فرقی ندارد و در هر دو OK چاپ می‌شود و تنها در صورتی که دستور اشتباه باشد خطا داده می‌شود.

ورودی	خروجی
DELETE filters	OK Bad Request
ورودی نمونه	خروجی نمونه
DELETE filters	OK

● خرید بلیت هواپیما

کاربر می‌تواند با دادن شناسه‌ی یک پرواز، تعداد مسافران، کلاس پروازی و نوع بلیتی که می‌خواهد، بلیت مورد نظر را خرید کند. کلاس پروازی را یکی از ۲ درجه‌ی اقتصادی یا تجاری در نظر بگیرید (economy یا business). همانطور که گفته شد اگر کاربر قصد خرید بلیت از درجه تجاری را داشت، قیمت آن 2.5 برابر قیمت همان بلیت از درجه اقتصادی است. نوع بلیت نشان دهنده‌ی سیستمی یا چارتری بودن آن است (refundable یا nonrefundable). در صورتی که بلیت از نوع سیستمی باشد، کاربر مجاز به لغو کردن بلیت است و اگر آن بلیت را لغو کند، نصف هزینه بلیت به کاربر بازگردانده می‌شود. اگر بلیت را چارتری

خرید کند، نمی‌تواند آن بلیت را لغو کند. طبیعتاً پس از خرید بلیت، هزینه‌ی آن بلیت به تعداد خریداری شده، از اعتبار کاربر کاسته خواهد شد. در صورتی که خرید بلیت موفقیت‌آمیز باشد، شناسه‌ی آن بلیت نمایش داده می‌شود. شناسه بلیت‌های یک کاربر یکتا است. همچنین توجه کنید که هیچ دو کاربری بلیت با شناسه‌ی یکسان نباید داشته باشند. برای اختصاص شناسه بلیت به کاربران، از عدد ۱ شروع کنید. یعنی کاربری که اولین بلیت را خرید کرد، بلیت با شناسه‌ی ۱ خواهد داشت و کاربری که دومین بلیت را خرید کرد، بلیت با شناسه‌ی ۲ خواهد داشت و

به جز خطاهای عادی که شرایط وقوع آن قبلاً گفته شد، در این بخش خروجی‌های دیگری نیز متصور است.

- اگر اعتبار کاربر برای خرید آن بلیت کافی نباشد باید خطای Bad Request چاپ شود.
- اگر پرواز به تعداد خواسته شده ظرفیت نداشت، باید خطای Bad Request چاپ شود.

ورودی
<code>POST tickets ? flight <flight_id> quantity <quantity> class <class> type <type></code>
خروجی
<code><ticket_id> Bad Request Not Found</code>
ورودی نمونه
<code>POST tickets ? flight 15 quantity 2 class economy type refundable</code>
خروجی نمونه
5

● مشاهده‌ی بلیت‌های خریداری شده توسط کاربر

کاربر می‌تواند بلیت‌های خریداری شده خود را مشاهده کند. اطلاعات هر بلیت در یک خط چاپ می‌شوند. در صورتی که هنوز بلیتی خریداری نکرده پیام Empty در خروجی نمایش داده می‌شود.

ورودی
GET <code>tickets</code>
خروجی
<code><ticket_id> <flight_id> <airline> <quantity> <origin> <destination></code> <code><departure_date> <departure_time> <arrival_date> <arrival_time> <class> <type></code> <code><cost> Empty Bad Request</code>
ورودی نمونه
GET <code>tickets</code>
خروجی نمونه
<pre>3 70 Turkish 1 Istanbul Brussels 11 11:45 11 13:20 business refundable 209.00 9 64 Qatar 4 Copenhagen Doha 12 16:35 13 00:35 economy nonrefundable 1240.50</pre>

● مشاهده‌ی یک بلیت از بلیت‌های خریداری شده توسط کاربر

کاربر می‌تواند با دادن شناسه یکی از بلیت‌هایی که قبلاً خرید کرده است، آن بلیت را مشاهده کند. در صورتی که بلیت با آن شناسه در لیست بلیت‌های آن کاربر نبود یا قبلاً آن بلیت توسط کاربر لغو شده بود، خطای Not Found چاپ شود.

ورودی
GET <code>tickets ? id <id></code>
خروجی
<code><ticket_id> <flight_id> <airline> <quantity> <origin> <destination></code> <code><departure_date> <departure_time> <arrival_date> <arrival_time> <class> <type></code> <code><cost> Bad Request Not Found</code>

ورودی نمونه
GET tickets ? id 9
خروجی نمونه
9 64 Qatar 4 Copenhagen Doha 12 16:35 13 00:35 economy nonrefundable 1240.25

● لغو یک بلیت

کاربر می‌تواند در صورتی که بلیت خود را سیستمی خرید کرده باشد، با وارد کردن شناسه‌ی بلیتی که قبلاً خریده، آن بلیت را کنسل کند. در صورت موفقیت، نصف هزینه‌ی پرداخت شده به او باز می‌گردد. مشخص است که پس از لغو یک بلیت، ظرفیت آن پرواز یک واحد زیاد شده و بقیه می‌توانند آن بلیت را خریداری کنند. همچنین دقت کنید همانطور که گفته شد اگر کاربر بلیت خود را از نوع چارتر خرید کرده باشد مجاز به لغو آن بلیت نخواهد بود.

پس از لغو کردن یک بلیت، دیگر نباید آن بلیت در لیست بلیت‌های کاربر نمایش داده شود. اگر شناسه‌ی وارد شده وجود نداشته باشد یا قبلاً لغو شده باشد باید خطای Not Found داده شود. اگر کاربر قصد لغو بلیت از نوع چارتری را داشت خطای Bad Request داده شود.

ورودی
DELETE tickets ? id <ticket_id>
خروجی
OK Not Found Bad Request
ورودی نمونه
DELETE tickets ? id 3
خروجی نمونه
OK

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  
|— ...
```

- با توجه به حجم نسبتاً زیاد این فاز از تمرین توصیه می‌شود قبل از پیاده‌سازی کد طراحی اولیه‌ای برای منطق برنامه و روندهای آن مثل ثبت‌نام، ساخت لیست پروازها و ... انجام دهید و پس از این طراحی شروع به پیاده‌سازی آن کنید. از آن جایی که در فازهای بعدی شما باید رابط کاربری برنامه‌ی خود را از command line 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.
- **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 c++11 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 c++11 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 **uttravel.out** باشد.
- دقت کنید که تمام اعداد غیر صحیح که در خروجی چاپ می‌شوند باید **دقت 2 رقم اعشار بدون گردکردن** داشته باشند. (مثلاً 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 شماست. 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.
- توجه داشته باشید که حالت‌های خاصی که در صورت پروژه ذکر نشده است در تست‌های خودکار نخواهد بود و می‌توانید به هر شکلی که مد نظر دارید آن‌ها را مدیریت کنید.