



UTunes!

مقدمه



این پروژه به جمع‌بندی 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 پخش موسیقی را طراحی کنید که دارای امکاناتی مانند اضافه کردن آهنگ‌ها به علاقه‌مندی‌ها و ساخت لیست پخش اختصاصی خواهد بود.

در فاز اول این پروژه شما منطبق برنام‌ی این سایت را پیاده‌سازی می‌کنید که با استفاده از رابط کاربری خط فرمان¹ می‌توان از آن استفاده کرد. ابتدا اطلاعات آهنگ‌ها از فایل خوانده خواهند شد و سپس کاربران مختلف می‌توانند پس از ثبت نام یا ورود، لیست آهنگ‌ها را طبق خواسته‌ی خودشان فیلتر کرده و مشاهده کنند و از باقی قابلیت‌های سیستم استفاده کنند.

¹CLI (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، دستوراتی همراه با آرگومان‌های لازم برای اجرای آن‌ها در جریان ورودی استاندارد² وارد می‌کند.

همین‌طور در نظر داشته باشید که این اطلاعات، یعنی آرگومان‌های هر دستور، پس از علامت ؟ در دستور می‌آیند. بعضی از دستورات نیز آرگومانی برای دادن به دستور نیاز ندارند که در آن‌صورت تنها دستور مربوط به آن‌ها وارد می‌شود و ؟ در دستور وجود نخواهد داشت.

فرمت کلی دستورات به شکل زیر است:

```
POST/GET/DELETE <command> ? <options>
```

نکته دیگری که در دستورات وجود دارد این است که قبل از وارد شدن دستور، عبارت GET، POST یا DELETE وارد می‌شود که به این صورت دستورات به نحوی از هم جدا و طبقه‌بندی می‌شوند. به این شکل که دستوراتی که بیشتر برای گرفتن اطلاعات از سیستم است، از GET، دستوراتی که بیشتر برای فرستادن اطلاعات به سیستم است، از POST و دستوراتی که برای حذف است از DELETE استفاده می‌کنند. 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 داده‌شده 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. همچنین توضیح کلی این موارد در بخش خطاها در آخر صورت پروژه آمده است.

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.



² Standard Input Stream

• اطلاعات آهنگها

فایل اطلاعات آهنگها در اختیارتان قرار گرفته است که شامل تعدادی قطعه است که توضیح اطلاعات هر آهنگ در ادامه آمده است.

هر آهنگ شامل اطلاعات زیر است:

1. id شناسه یکتای آهنگ است که برای دسترسی به آن استفاده می شود.
2. title نام آهنگ است.
3. artist نام خالق اثر است.
4. release_year سال انتشار است.
5. link لینک مستقیم آهنگ است که در فاز آخر پروژه برای پخش آهنگها استفاده خواهد شد.

دستورها

• وارد کردن اطلاعات آهنگها

در ابتدای شروع برنامه، باید اطلاعات آهنگها از یک فایل csv که نام آن به عنوان آرگومان خط فرمان به برنامه می شما داده می شود، خوانده شده و در حافظه ی برنامه ذخیره شود. دستور آغاز برنامه به شکل زیر خواهد بود.

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

ورودی

```
./utunes.out <csv_file_relative_address>
```

ورودی نمونه

```
./utunes.out folder1/folder2/utunes.csv
```

• احراز هویت

○ ثبت نام

با این دستور افراد می‌توانند در سامانه ثبت نام کنند که در این صورت پس از وارد کردن این دستور وارد سیستم نیز خواهند شد. ایمیل و نام کاربری هر کاربر یک رشته یکتا در سیستم است. اگر ایمیل یا 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 ? email <email> username <username> password <password>
خروجی
OK Bad Request

○ ورود

اگر فردی 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 ? email <email> 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

• نمایش آهنگ‌ها

کاربر می‌تواند با دستور زیر لیست آهنگ‌ها را به همراه اطلاعات خلاصه‌شده‌ی آن‌ها مشاهده کند. در هر خط اطلاعات یک آهنگ 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 آهنگ‌ها خالی بود پیام `Empty` را خروجی دهید. برای پیش نیامدن مشکل در تست‌های اتوماتیک ترتیب چاپ شدن آهنگ‌ها همواره به ترتیب `id` آهنگ 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 آهنگ‌ها نمایش داده خواهد شد.

ورودی
<code>GET songs</code>
خروجی
<code><id> <title> <artist> <release_year> Empty Permission Denied</code>
ورودی نمونه
<code>GET songs</code>
خروجی نمونه
<code>1 Perfect Ed Sheeran 2017 2 Morghe Sahar Mohammadreza Shajarian 1990 ...</code>

• نمایش مشخصات یک آهنگ

کاربر می‌تواند با وارد کردن شناسه‌ی یک آهنگ، اطلاعات کامل آن را دریافت کند. همین‌طور در صورتی که شناسه ورودی کاربر موجود نبود و یا به عبارتی valid 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.

num_of_likes تعداد کاربرانی است که این آهنگ را لایک کرده‌اند.

num_of_comments تعداد کامنت‌هایی است که روی این آهنگ گذاشته شده است.

num_of_playlists تعداد لیست پخش‌هایی است که شامل این آهنگ هستند.

ورودی	خروجی
GET songs ? id <id>	<id> <title> <artist> <release_year> #likes: <num_of_likes> #comments: <num_of_comments> #playlists: <num_of_playlists> Not Found Permission Denied
ورودی نمونه	خروجی نمونه
GET songs ? id 1	1 Perfect Ed Sheeran 2017 #likes: 123 #comments: 0 #playlist: 2

• پسندیدن آهنگ‌ها

○ پسندیدن یک آهنگ

کاربر می‌تواند با وارد کردن id یک آهنگ آن را به آهنگ‌هایی که می‌پسندد اضافه کند. در صورتی که کاربر، آهنگ را قبلاً پسندیده بود Bad Request داده می‌شود. در صورتی که آهنگ موجود نباشد نیز Not Found داده می‌شود.

ورودی	خروجی
POST likes ? id <id>	OK Bad Request Permission Denied
ورودی نمونه	خروجی نمونه
POST likes ? id 2	OK

○ نمایش آهنگ‌های پسندیده شده

کاربر می‌تواند با این دستور اطلاعات آهنگ‌هایی که پسندیده را مشاهده کند. هر آهنگ در یک خط نمایش داده می‌شود و ترتیب چاپ آنها id آنها است. در صورتی که کاربر، آهنگ پسندیده‌شده‌ای نداشته باشد Empty چاپ می‌شود.

ورودی	خروجی
GET likes	<id> <title> <artist> <release_year> Empty Permission Denied
ورودی نمونه	خروجی نمونه
GET likes	2 Morghe Sahar Mohammadreza Shajarian 1990 3 Shape Of My Heart Sting 1993 ...

○ حذف یک آهنگ از لیست پسندیده‌ها

کاربر می‌تواند با وارد کردن id آهنگی که می‌خواهد، آن را از لیست آهنگ‌هایی که پسندیده حذف کند. اگر آهنگ وارد شده قبلاً پسندیده نشده بود، Bad Request داده می‌شود.

ورودی	خروجی
DELETE likes ? id <id>	OK Bad Request Permission Denied
ورودی نمونه	خروجی نمونه
DELETE likes ? id 3	OK

• لیست پخش آهنگ‌ها

○ ایجاد یک لیست پخش

کاربر می‌تواند با وارد کردن یک نام و مشخص کردن عمومی یا خصوصی بودن لیست، برای خود لیست پخش با آن نام بسازد. در خروجی، شناسه لیست پخش جدید چاپ می‌شود. شناسه‌ها توسط برنامه ایجاد می‌شوند و از 1 شروع و یکی یکی اضافه خواهند شد. دقت کنید این شناسه بین لیست پخش‌های تمام کاربران باید یکتا باشد.

ورودی	خروجی
POST <code>playlists ? name <name> privacy <public/private></code>	<code><id> </code> <code>Bad Request </code> <code>Permission Denied</code>
ورودی نمونه	خروجی نمونه
POST <code>playlists ? name my_shiny_playlist privacy public</code>	4

○ نمایش فهرست لیست پخش‌های یک کاربر

کاربر می‌تواند با این دستور فهرستی از تمام لیست پخش‌های خودش یا دیگران را مشاهده کند. ترتیب چاپ، شناسه لیست پخش است. اگر شناسه کاربری که وارد شده، برای کاربر دیگری باشد، تنها لیست پخش‌هایی از او که به صورت عمومی هستند نمایش داده خواهد شد. اگر شناسه وارد شده، شناسه خود کاربر بود، تمام لیست پخش‌هایش نمایش داده خواهد شد. اگر موردی برای نمایش نبود، Empty چاپ می‌شود.

ورودی	خروجی
GET <code>playlists ? username <username></code>	<code><id> <name> <public/private></code> ... <code> Bad Request Empty Permission Denied</code>
ورودی نمونه	خروجی نمونه
GET <code>playlists ? username my_username</code>	3 my_first_playlist private 4 my_shiny_playlist public

○ اضافه کردن یک آهنگ به لیست پخش

کاربر می‌تواند با وارد کردن شناسه یک آهنگ و شناسه یک لیست پخش که متعلق به خودش است، آن آهنگ را به آن لیست پخش اضافه کند. اگر لیست پخش متعلق به این کاربر نباشد، خطای Permission Denied داده می‌شود.

ورودی	خروجی
POST <code>playlists_songs ? playlist_id <playlist_id> song_id <song_id></code>	OK Bad Request Permission Denied
ورودی نمونه	خروجی نمونه
POST <code>playlists_songs ? playlist_id 4 song_id 3</code>	OK

○ نمایش لیست آهنگ‌های داخل یک لیست پخش

کاربر می‌تواند با وارد کردن شناسه یک لیست پخش، اطلاعات آهنگ‌هایی که در آن گذاشته را مشاهده کند. اگر لیست پخش برای کاربر دیگری باشد و به صورت خصوصی ساخته شده باشد، خطای Permission Denied داده خواهد شد. هر آهنگ در یک خط نمایش داده می‌شود و ترتیب چاپ آنها شناسه آن‌ها است. در صورتی که لیست خالی باشد Empty چاپ می‌شود.

ورودی	خروجی
GET <code>playlists_songs ? playlist_id <playlist_id></code>	<id> <title> <artist> <release_year> Empty Permission Denied
ورودی نمونه	خروجی نمونه
GET <code>playlists_songs ? playlist_id 4</code>	2 Morghe Sahar Mohammadreza Shajarian 1990 3 Shape Of My Heart Sting 1993 ...

اطلاعات ورودی همگی در یک خط هستند و به علت محدودیت عرض صفحه به خط بعد رفته‌اند.

○ حذف یک آهنگ از لیست پخش

کاربر می‌تواند با وارد کردن شناسه یک لیست پخش خود و id آهنگی که می‌خواهد، آن را از لیست پخش خود حذف کند. اگر آهنگ وارد شده در لیست پخش نبود، Bad Request داده می‌شود. اگر لیست پخش متعلق به این کاربر نبود Permission Denied داده می‌شود.

ورودی	خروجی
DELETE playlists_songs ? playlist_id <playlist_id> song_id <song_id>	OK Bad Request Permission Denied
ورودی نمونه	خروجی نمونه
DELETE playlists_songs ? playlist_id 4 song_id 2	OK

● نمایش کاربران دیگر

کاربر می‌تواند لیستی از نام کاربری بقیه کاربرانی که در این سامانه ثبت نام کرده‌اند را ببیند. ترتیب چاپ نام‌های کاربری به ترتیب الفبایی³ است. اگر هیچ کاربر دیگری جز خود کاربر در سامانه ثبت‌نام نکرده بود Empty چاپ می‌شود. دقت کنید که در هر صورت نام خود کاربر نباید چاپ شود.

ورودی
GET users
خروجی
<username1> <username2> ... Bad Request Empty Permission Denied
ورودی نمونه
GET users
خروجی نمونه
Ali Amin Armin Bardia Mohsen Saeed

³ Lexicographical Order

• اضافه کردن فیلتر

هر کاربر می تواند تعدادی فیلتر برای خودش اضافه کند. فیلترها باعث می شوند لیست آهنگ هایی که با دستور نمایش آهنگ ها دیده خواهد شد پالایش شوند. انتظار می رود این فیلترها را به صورت کلاس پیاده سازی کنید و برای ذخیره سازی و اعمال آنها از چندریختی⁴ استفاده کنید. برای آشنایی با نحوه ی تعریف فیلترهای مختلف به صورت کلاس، می توانید از این [لینک](#)⁵ استفاده کنید. البته بخش چندریختی را که برای ذخیره فیلترهای مختلف در یک لیست لازم است، خودتان به این ساختار اضافه کنید.

توجه کنید که اگر یک نوع فیلتر، برای بار دوم با موفقیت اضافه شود، اطلاعات فیلتر دوم جایگزین فیلتر قبلی خواهند شد.

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

○ فیلتر نام خالق اثر

این فیلتر فقط آهنگ هایی که artist آنها در دستور مشخص شده است را تایید می کند.

ورودی	خروجی
POST filters ? artist <artist>	OK Bad Request Permission Denied
ورودی نمونه	خروجی نمونه
POST filters ? artist Homayoun Shajarian	OK

○ فیلتر بازه ی تاریخ انتشار آهنگ

این فیلتر فقط آهنگ هایی که تاریخ انتشارشان در محدوده ی مشخص شده است را تایید می کند. (توجه داشته باشید فیلتر به صورت بسته عمل می کند یعنی هم ابتدا و هم انتهای بازه را شامل می شود) اگر دستور به شکل زیر نبود یا عدد پایانی از عدد شروع کمتر بود باید خطای Bad Request داده شود. برای فیلتر کردن آهنگ های یک سال خاص، عدد شروع و پایان برابر خواهند بود.

ورودی	خروجی
POST filters ? min_year <min_year> max_year <max_year>	OK Bad Request Permission Denied
ورودی نمونه	خروجی نمونه
POST filters ? min_year 2015 max_year 2020	OK

⁴ polymorphism

⁵ https://www.tutorialspoint.com/design_pattern/filter_pattern.htm

○ فیلتر تعداد لایک

این فیلتر، فقط آهنگ‌هایی که تعداد کسانی که لایکش کرده‌اند در بازه‌ی خاصی است را تایید می‌کند. (دقت کنید این فیلتر نیز همانند فیلتر قبلی به صورت بسته عمل می‌کند). اگر اعداد داده شده کمتر از صفر بودند، یا عدد پایانی از عدد شروع کمتر بود باید خطای Bad Request داده شود.

ورودی	خروجی
POST filters ? min_like <min_like> max_like <max_like>	OK Bad Request Permission Denied
ورودی نمونه	خروجی نمونه
POST filters ? min_like 10 max_like 15	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 Permission Denied
ورودی نمونه	خروجی نمونه
DELETE filters	OK

• اضافه کردن نظر به آهنگ

کاربر با دادن شناسه‌ی یک آهنگ می‌تواند روی آن نظر اضافه کند. comment یک رشته‌ی متنی است و اگر کاربری چندین بار با استفاده از این دستور کامنت بگذارد، همه‌ی کامنت‌ها ذخیره خواهند شد. در این فاز می‌توانید فرض کنید که کامنت، شامل کاراکتر فاصله نیست. ضمناً در ورودی یک عدد که واحد آن ثانیه است نیز گرفته می‌شود تا بتوان روی یک زمان خاص از آهنگ کامنت گذاشت. اگر شناسه آهنگ داده شده وجود نداشت، Not Found نمایش داده می‌شود.

ورودی	خروجی
POST comments ? song_id <song_id> time <time> comment <comment>	OK Bad Request Not Found Permission Denied
ورودی نمونه	خروجی نمونه
POST comments ? song_id 1 time 25 comment Great_lyrics	OK

• مشاهده‌ی نظرات روی یک آهنگ

کاربران می‌توانند لیست نظرات روی یک آهنگ را با دستور زیر مشاهده کنند. هر نظر در یک خط نمایش داده می‌شود و شامل زمانی از آهنگ که کامنت گذاشته شده، نام کاربری و متن نظر او است. ترتیب چاپ براساس زمانی از آهنگ که کامنت رویش گذاشته شده است و اگر برابر بود، بر اساس نام کاربری خواهد بود. اگر هیچ کامنتی گذاشته نشده بود Empty چاپ می‌شود. اگر شناسه آهنگ داده شده وجود نداشت، Not Found نمایش داده می‌شود.

ورودی	خروجی
GET comments ? song_id <song_id>	<time1> <username1>: <comment1> <time2> <username2>: <comment2> ... Bad Request Not Found Empty Permission Denied
ورودی نمونه	خروجی نمونه
GET comments ? song_id 1	25 mohsen: Great_lyrics 25 Saeed: Highly_recommended 36 Amin: PERFECT

○ خطای عدم وجود

اگر دسترسی که کاربر وارد می‌کند (قسمت پس از POST یا GET یا DELETE)، در لیست دستورهای تعریف‌شده نباشد یا کاربر با وارد کردن شناسه‌ی ناموجود در سیستم قصد دسترسی به موجودیتی را داشته باشد، این خطا در خروجی چاپ می‌شود.

خروجی
Not Found

○ خطای درخواست اشتباه

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

خروجی
Bad Request

○ خطای 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

○ هش⁶ کردن گذرواژه

همان‌طور که می‌دانید گذرواژه رشته‌ای است که افراد تمایل دارند از دسترسی سایرین حفظ شود. به این دلیل است که هنگام ذخیره‌سازی گذرواژه آن را به صورت خام در حافظه نگه‌داری نمی‌کنند. هش‌کردن روشی است که توسط آن می‌توان یک رشته را طی فرآیندی غیرقابل بازگشت به رشته‌ی دیگری تبدیل کرد؛ یعنی هش هر رشته یک رشته‌ی یکتا است و با هر رشته‌ی دیگر تفاوت دارد و همچنین از طریق هش یک رشته نمی‌توان به آن رشته دسترسی پیدا کرد. در این قسمت باید گذرواژه‌های افراد را به صورت هش‌شده در حافظه نگه‌داری کنیم تا از دسترسی به گذرواژه خام پرهیز شود. هش‌کردن الگوریتم‌های متفاوتی دارد که می‌توانید در اینترنت درباره‌ی آن‌ها جستجو کنید. دقت کنید که اگر نیاز داشتید، تنها از کتابخانه‌های استاندارد استفاده کنید تا در تست خودکار مشکلی پیش نیاید.

⁶ Hash

نحوه‌ی تحویل و نکات

- 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 ۸۱۰۱۹۸۹۹۹ است، نام پرونده‌ی شما باید A7-810198999.zip باشد.

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

A7-810198999.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 ثبت نام و ساخت لیست پخش و آهنگ‌ها 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 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.
- برنامه 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 **utunes.out** باشد.
- 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.