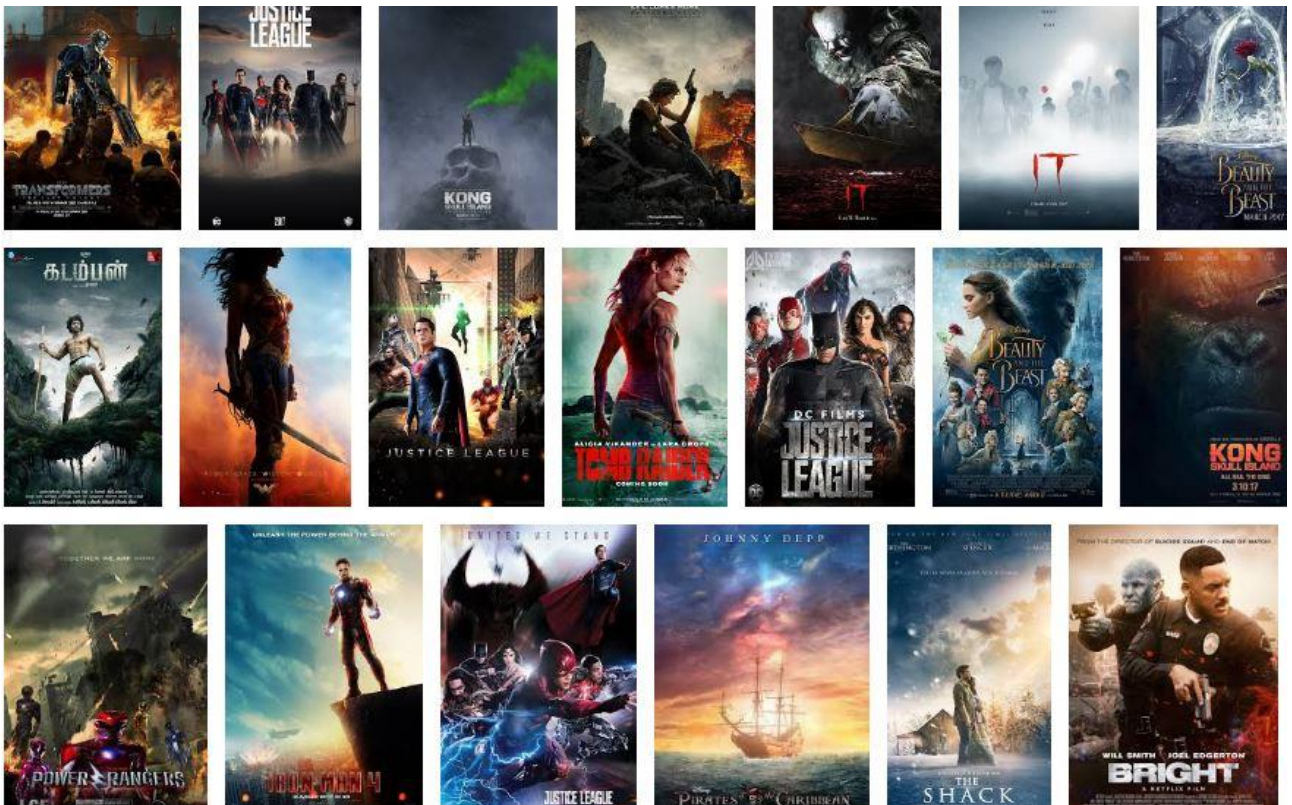


بنام پروردگار بزرگ



## تمرین اول

درس: برنامه سازی پیشرفته  
پیاده سازی کلاس فیلم و روابط مربوط به آن



## مقدمه

در این تمرین قصد داریم با کلاسی به نام کلاس فیلم کار کنیم! که صفات کلاس: نام فیلم، نمره‌ی منتقدین (صرفاً یک عدد)، سال تولید، نام کارگردان، زمان فیلم (صرفاً به دقیقه)، کشور سازنده و شناسه‌ی عددی (آیدی) هستند. در این تمرین باید رابطی را پیاده‌سازی کنید که با گرفتن یک خط دستور، عمل مورد نظر کاربر را انجام می‌دهد (در خط فرمان یا `command` line).

دستورهایی که باید برنامه آن‌ها را شناسایی کند:

**نکته:** در تابع `main` یک آرایه پویا از جنس کلاس فیلم برای ذخیره‌سازی فیلم‌ها تعریف کنید.

<code>add &lt;movie_name&gt; &lt;score&gt; &lt;data&gt; &lt;time&gt; &lt;country&gt;</code>	با این دستور، یک شی از کلاس فیلم با مشخصات وارد شده ساخته می‌شود و به آرایه‌ی پویا اضافه می‌شود.
<code>add SpiderMan3 10 2014 120 Iran</code>	مثال:
<code>remove &lt;movie_name&gt;</code>	با این دستور، فیلمی که نام آن وارد شده است از آرایه حذف می‌شود.
<code>remove SpiderMan3</code>	مثال:
<code>sort-by score</code>	با این دستور تمام فیلم‌ها براساس نمره‌ی منتقدین مرتب می‌شوند.
<code>sort-by name</code>	با این دستور تمام فیلم‌ها براساس نام فیلم مرتب می‌شوند.
<code>show &lt;movie_name&gt;</code>	با این دستور صرفاً فیلمی که نام آن وارد شده است نمایش داده می‌شود.
<code>show SpiderMan3</code>	مثال:
<code>show-all</code>	با این دستور تمام اشیاء تولید شده چاپ می‌شوند.
<code>average-score</code>	با این دستور میانگین نمره‌ی منتقدین تمام فیلم‌ها چاپ می‌شود.
<code>help</code>	با این کامند تمامی دستورهای برنامه با راهنمایی برای کاربر چاپ می‌شود.

### تابع تولید عدد تصادفی:

برای هر شی از کلاس فیلم که ساخته می شود باید یک عدد واقعا تصادفی غیرتکراری توسط این تابع ساخته شود و مقدار تولید شده به صفت شناسه ی عددی مربوط به شی (فیلم) ساخته شده مقداردهی می شود.  
**نکته:** برای تولید از random engine استفاده کنید.

### تابع Command Separator:

این تابع دستور وارد شده ی کاربر را کلمه به کلمه (با توجه با فاصله های بین آنها) جدا می کند.

### دستور add:

این دستور یک شی با مشخصاتی که کاربر آن را وارد می کند می سازد و آن را در آرایه ای پویا که اشیا را در آن نگه می داریم می ریزیم.  
**راهنمایی:** شی فیلمی که کاربر قصد اضافه کردن و آرایه ی پویایی که اشیا در آن ذخیره می شوند ورودی های این تابع هستند.

### دستور remove:

این دستور با توجه نام وارد شده توسط کاربر یک شی را از آرایه حذف می کند.  
**توجه:** برای تخصیص حافظه از عملگر new و برای آزاد کردن حافظه تخصیص داده شده از delete استفاده کنید.

### دستور sort-by:

این دستور فیلم ها را مرتب می کند. آرگومان دوم نوع معیار مرتب سازی را بیان می کند. اگر name باشد، بر اساس نام فیلم و اگر score باشد، بر اساس نمره ی منتقدین مرتب می گردد.  
**راهنمایی:** برای پیاده سازی این توابع از bubble sort استفاده کنید.

### دستور show:

در پیاده سازی این دستور حتما از کتابخانه iomanip استفاده کنید. این دستور تمام صفات فیلم را در یک خط نمایش می دهد.  
توجه: باید تمامی اشیا به صورت مرتب و زیر هم قرار بگیرند.

### دستور show-all:

این دستور همه ی فیلم ها را با استفاده از دستور show نمایش می دهد.

### دستور average-score:

این دستور میانگین نمره ی منتقدین به تمامی فیلم ها را به نمایش می گزارد.

## دستور help:

این دستور تمامی دستورها را به همراه راهنمایی‌شان چاپ می‌کند.

## امتیازی: دستور score-details:

این دستور، نمره‌هایی که منتقدین به این فیلم داده‌اند را به صورت آرایه‌ای دو بعدی نگه می‌دارد. این امتیازها، به صورت مجموعه‌ای از منتقدین است که هر منتقد می‌تواند چند امتیاز (مثلاً درباره‌ی بخش‌های گوناگون آن فیلم) به آن داده باشد. شیوه‌ی استفاده از این دستور یکی از دو شکل زیر است:

```
score-details <movie_name> set <critics_count>  
<score_count> <scores...>  
score-details <movie_name> get <critic_number>  
<score_number>
```

دستور اول امتیازهای منتقدین را ذخیره می‌کند. دستور دوم نیز امتیاز خواسته‌شده را برمی‌گرداند. نمونه:

```
score-details SpiderMan3 set 3 2 100 85 80 73 60 70  
score-details SpiderMan3 get 2 2
```

در نمونه‌ی بالا، به برنامه گفته می‌شود که ۳ منتقد هست که هر یک ۲ امتیاز داده‌اند. سپس، در دستور بعدی، دومین امتیاز منتقد دوم را درخواست می‌کنیم (که ۷۳ است).

**توجه:** برای پیاده‌سازی این دستور حتما باید از آرایه‌های پویا استفاده نمایید.

## نکات قابل توجه در پیاده‌سازی تمرین:

- در این برنامه موظف به استفاده از کتابخانه‌ی `iomanip`، اشاره‌گرها (Pointers)، تولید اعداد تصادفی (Random Number Generation) و آرایه‌های پویا (Dynamic Arrays) هستید. در صورت عدم استفاده از مباحث فوق بخشی از نمره را از دست خواهید داد.
- در هر بخشی از برنامه که داده‌ای پیدا نشود (مانند هنگام حذف یک فیلم)، برنامه باید خطا را تشخیص داده و پیام مناسبی را در خروجی چاپ کند.
- برنامه در دریافت دستورها نباید به فاصله و بزرگی و کوچکی حروف حساس باشند؛ مثلاً دستور

`ReMove`      `SpidErMan3`

نیز برنامه باید درست کار کند.

- تمرین را با زبان سی‌پلاس‌پلاس پیاده‌سازی کنید.
- کدنویسی خود را به صورت ماژول‌بندی شده انجام دهید.
- تمرین‌ها علاوه بر بارگزاری در سامانه‌ی درس‌افزار، ممکن است به صورت آنلاین نیز تحویل گرفته شود.
- استفاده از هرگونه کانتیرها (مانند وکتور) مجاز نیست!
- تهیه‌ی گزارش کار (خوب و قابل فهم) الزامی است.
- استفاده از سی‌میک (CMake) امتیازی است و ۱۰ درصد نمره‌ی اضافی دارد.

**مهم:** مهلت تحویل تمرین ۱۰ روز است و به ازای هر روز دیرکرد ۲۰ درصد نمره کسر می‌شود.