



مقدمه

هدف از این تمرین آشنایی شما با طراحی بالا به پایین^۱ یک مسئله است. با توجه به حجم پروژه لازم است که قبل از شروع پیاده‌سازی زمانی را به طراحی اختصاص دهید. در غیر این صورت در هنگام پیاده‌سازی با مشکل مواجه می‌شوید. بنابراین ابتدا به چگونگی شکستن این مسئله به مسائل کوچکتر و پخش کردن مسئولیت‌ها میان قسمت‌های مختلف برنامه فکر کنید.

برای آشنایی بیشتر شما با این نوع طراحی یک ویدیو آپلود خواهد شد.

نمایش برنامه فیلم‌های سینماها

در این تمرین قرار است با بررسی برنامه هفتگی سانس‌های مختلف سینماها، برنامه‌ی پخش یک فیلم مشخص در روزهای هفته را به صورت دو جدول مختلف نشان دهید. برای این منظور اطلاعات مربوط به سانس‌های فیلم‌ها توسط فایلی به شما داده می‌شود.

فرمت فایل ورودی

فرمت فایل ورودی CSV^۲ است. در این نوع فایل داده‌ها توسط کاما (,) از یکدیگر جدا می‌شوند. برنامه‌ی شما باید این فایل را بخواند و اطلاعات آن را به شیوه‌ی مناسب ذخیره کند. خط اول این فایل حاوی عنوان ستون‌ها است. ترتیب این ستون‌ها ثابت نیست و عنوان آن‌ها به شکل زیر است:

توضیح	مثال	ستون
اسم سینما	Azadi	CinemaName
اسم فیلم	Inception	MovieName

^۱ Top Down Design

^۲ Comma Separated Values

Day	Monday	روز پخش فیلم که یکی از روزهای هفته به شکل زیر است: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday
StartingTime	08:00	ساعت شروع به شکل hh:mm
FinishingTime	16:30	ساعت پایان به شکل hh:mm
Price	30	قیمت فیلم (عدد طبیعی)
Hall	4	شماره سالن (عدد طبیعی)

برای مثال این فایل می‌تواند به هر دو شکل زیر باشد:

```
CinemaName,Day,Price,Hall,FinishingTime,StartingTime,MovieName
Ghods,Sunday,10,1,22:30,20:00,Inception
Enghelab,Friday,25,10,10:30,09:00,The Terminal
```

```
Price,Day,MovieName,StartingTime,Hall,CinemaName,FinishingTime
10,Sunday,Inception,20:00,1,Ghods,22:30
25,Friday,The Terminal,09:00,10,Enghelab,10:30
```

خواندن درخواست‌های کاربر و نمایش جدول

این برنامه با استفاده از تعدادی دستور که از ورودی استاندارد³ می‌گیرد، عملیات مورد انتظار را انجام می‌دهد. به طور کلی ۲ دستور شناخته شده برای برنامه وجود دارد که در ادامه نحوه عملکرد آن‌ها را مشخص کرده‌ایم. اگر کاربر دستوری به غیر از این ۲ تا وارد کند، برنامه باید آن را نادیده گرفته و دستور بعدی را بخواند.

۱. نمایش نام تمام فیلم‌ها

```
GET ALL MOVIES
```

با وارد شدن این دستور تمام فیلم‌های موجود در تمام سینماها (نام هر فیلم در یک سطر) به ترتیب الفبا در خروجی استاندارد چاپ شوند.

³ stdin

۲. نمایش برنامه هفتگی پخش یک فیلم

برنامه با خواندن دستور زیر، باید دو جدول برنامه هفتگی پخش برای فیلمی مشخص خروجی دهد. جدول اول در ترمینال نمایش داده می‌شود. جدول دوم یک خروجی html است که در ادامه به جزئیات هرکدام پرداخته‌ایم.

```
GET SCHEDULE <MOVIE NAME>
```

منظور از MOVIE NAME همان نام فیلم است که در فایل CSV آمده بوده که نمادهای < و > در واقعیت به برنامه داده نمی‌شوند. برای مثال اگر در فایل CSV فیلم The Pianist وجود داشته باشد دستور زیر یک دستور مجاز است:

```
GET SCHEDULE The Pianist
```

در صورتی که نام فیلم مورد نظر وجود نداشت، برنامه این دستور را در نظر نمی‌گیرد و از آن عبور می‌کند.

جدول در ترمینال:

	08:00	10:00	12:00	14:00	16:00	18:00	20:00	22:00	00:00
Saturday					Esteghlal	Koorosh		Esteghlal	
Sunday		Felestin				Charsoo			
Monday		Azadl	Alborz			Paytakht	Arg		
Tuesday				Arash				Felestin	
Wednesday	Felestin		Charsoo	Eram					
Thursday	Javan		Koorosh	Mellat		Payam			
Friday		Ghods		Hafez	Arg	Charsoo			

تصویر بالا نمونه‌ای از جدولی است که قرار است در ترمینال نمایش دهید.

قواعد نوشتن جدول:

- سطر اول جدول همیشه شکل ثابتی دارد و مستقل از نام فیلم است. ساعت‌های ۰۸:۰۰، ۱۰:۰۰، ۱۲:۰۰، ۱۴:۰۰، ۱۶:۰۰، ۱۸:۰۰، ۲۰:۰۰، ۲۲:۰۰ و ۰۰:۰۰ به ترتیب با فاصله‌های ۱۵ اسپیس چاپ می‌شوند. می‌توانید به فایل Braveheart.txt مراجعه کنید تا این نکات را بررسی کنید. ستون متناظر با ساعت ۰۸:۰۰ ستون ۱۱ (شماره‌گذاری ستون‌ها را از ۱ در نظر گرفتیم) است.
- روزهای هفته به ترتیب

Saturday - Sunday - Monday - Tuesday - Wednesday - Thursday - Friday

هستند که در ابتدای سطرهای ۳، ۵، ۷، ... و ۱۵ نوشته می‌شوند (شروع شماره‌گذاری سطرها را در اینجا از ۱ گرفتیم)

● در هر روزی سینماهایی وجود دارند که در ساعات مختلفی فیلم مورد درخواست کاربر را پخش می‌کنند که این ساعت‌ها می‌توانند تداخل داشته باشند. شما باید به ازای هر روز بیشترین تعداد سانس دو به دو بدون تداخل (بازه‌های دو به دو جدا از هم) را که می‌توانید نمایش دهید. اگر شرایطی پیش آمد که چندین حالت برای انتخاب بازه‌ها به طوری که تعداد آن‌ها بیشینه باشد داشتید، باید اولویت را به ترتیب (۱) بازه‌ای که زمان آغاز زودتری دارد (۲) بازه‌ای که قیمت فیلم متناظر با آن کمتر است و (۳) بازه‌ای که نام سینمای متناظر با آن از نظر الفبایی کوچک‌تر است در نظر بگیرید.

(راهنمایی: ابتدا تابعی بنویسید که برداری^۴ دریافت کند و اعضای آن را با اولویت (۱) ساعت شروع یا پایان (۲) قیمت آن فیلم در سینما (۳) نام آن سینما مرتب کند. سپس شروع به انتخاب بازه‌ها می‌کنید. به این صورت که روی بازه‌های مرتب شده از اول تا آخر حرکت کرده و هر دفعه به بازه‌ای رسیدید که با آخرین بازه انتخاب شده تداخل نداشت، آن را انتخاب کنید. برای آشنایی با حالت‌های کلی‌تر این سوال (مثل بازه‌های با طول نابرابر) و اثبات الگوریتم گفته شده به [این](#) صفحه مراجعه کنید)

برای مثال در فایل CSV داده شده برای فیلم Braveheart در روز Friday، هفت بازه وجود دارد که مقادیر زیر هستند (شماره‌گذاری سطرها صرفاً برای بیان بهتر مثال است و شماره‌ها جزو فایل CSV نیستند):

- ۱) Hafez,Braveheart,Friday,13:00,15:00,20,4
- ۲) Ghods,Braveheart,Friday,09:30,11:30,15,6
- ۳) Ghods,Braveheart,Friday,11:00,13:00,25,1
- ۴) Arg,Braveheart,Friday,15:00,17:00,15,8
- ۵) Charsoo,Braveheart,Friday,17:00,19:00,20,2
- ۶) Felestin,Braveheart,Friday,10:30,12:30,15,1
- ۷) Mellat,Braveheart,Friday,17:00,19:00,20,2

اگر الگوریتم گفته شده را روی این بازه‌ها اجرا کنیم در ابتدا بازه‌های ۲، ۱ و ۴ انتخاب می‌شوند. بازه‌های ۵ و ۷ ساعت شروع یکسانی دارند. قیمت بلیط نیز برای هر دو آن‌ها یکسان (۲۰) است. بنابراین اولویت آخر که همان نام سینما است بررسی می‌شود که $\text{Charsoo} < \text{Mellat}$ بنابراین بازه ۵ انتخاب می‌شود.

● به ازای هر سانس موجود در یک سینما که قصد نمایش آن را دارید، بایستی یک مستطیل به کمک کاراکترهای + و - و + تشکیل دهید که درون آن نام سینمایی که آن فیلم را نشان می‌دهد را به همراه تعدادی کاراکتر فاصله^۵ بنویسید.

^۴ vector

^۵ space

○ به ازای هر نیم ساعت باید ۵ کاراکتر در نظر بگیرید

■ اگر مدت زمان فیلمی hh:mm (اختلاف ساعت پایان و شروع) باشد، تعداد کاراکترهایی که دقیقا بین دو کاراکتر | (بدون احتساب خود | ها) باید قرار گیرد از فرمول زیر بدست می‌آید:

$$1 - \left(\frac{hh:mm}{00:30} \times 5 \right)$$

■ اگر فیلمی در زمان hh:mm آغاز شود شماره ستونی که کاراکتر | مربوط به آغاز آن

فیلم نوشته می‌شود از رابطه زیر به دست می‌آید :

$$10 + \left(\frac{hh:mm - 08:00}{00:30} \times 5 \right)$$

(شماره گذاری ستون‌ها را از ۰ در نظر گرفتیم)

○ در سطر بالا و پایین هر کاراکتر | و در همان ستون، باید علامت + نوشته شده باشد. بین

علامت‌های + که فیلمی در آن بازه وجود دارد هم باید علامت‌های - نوشته شده باشد.

○ اگر با قاعده بالا نتیجه داد که یک خانه هم باید علامت - باشد و هم علامت +، علامت +

بگذارید (این حالت موقعی رخ می‌دهد که زمان آغاز/پایان فیلمی در میان بازه پخش آن فیلم

در روز قبلی یا بعدی آن باشد. در تصویری که بالاتر ارائه دادیم، تلاقی سینماهای قدس در روز

جمعه و جوان در روز پنجشنبه این موضوع را نشان می‌دهد).

● یک نمونه فایل CSV به شما داده شده است. خروجی جدول به ازای دستور

GET SCHEDULE Braveheart

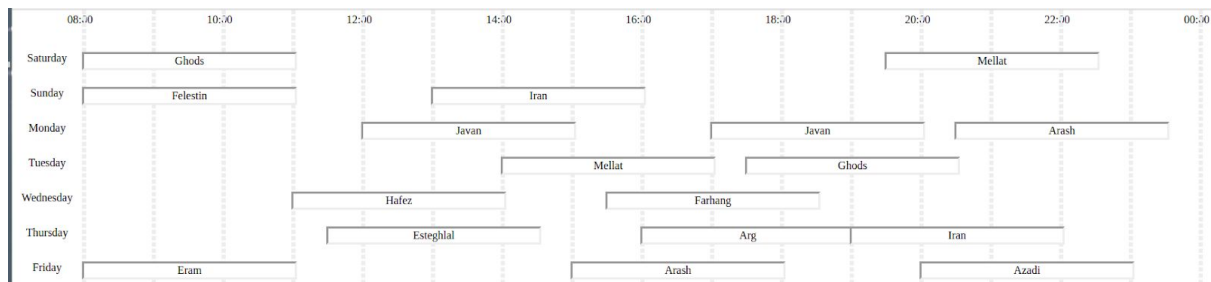
روی همان فایل CSV، یک فایل به نام Braveheart.txt است که به شما داده است تا بتوانید با

خروجی کد خود diff بگیرید و تعداد فاصله‌های لازم و ... خود را با توجه به آن تنظیم کنید.

جدول با فرمت HTML

تمام دانشی که از HTML برای پیاده‌سازی این بخش نیاز دارید آشنایی با مفهوم تگ است که برای این موضوع می‌توانید به این [لینک](#) مراجعه کنید.

برنامه باید یک فایل با پسوند html. به نام اسم فیلم، خروجی دهد که در آن تگ‌های لازم برای تولید جدولی مشابه تصویر زیر را بنویسد. ساختار سطر و ستون‌های این جدول کاملا مشابه جدولی است که در بخش قبل پیاده‌سازی کردید. تصویر زیر نمونه‌ای از این نوع جدول است.



فایل Whiplash.html که در کنار صورت پروژه به شما داده شده است، نمونه‌ای است از آنچه کد شما می‌بایست خروجی دهد. همچنین یک فایل به نام style.css نیز کنار صورت پروژه به شما داده است که لازم است این فایل را دقیقاً در همان محلی که فایل با پسوند html. شما قرار دارد، قرار دهید تا نمایش زیباتری از جدولی که تولید می‌کنید داشته باشید. شما کاری با محتوای این فایل ندارید و صرفاً آن را به طور ثابت در همان پوشه پروژه خود که فایل خروجی html در آن ساخته می‌شود، قرار می‌دهید.

اگر فایل Whiplash.html را با مرورگر خود باز کنید و سپس CTRL + U را فشار دهید، تگ‌های این فایل را مشاهده می‌کنید. همچنین اگر این فایل را با ادیتور دلخواه خود باز کنید، تگ‌ها را مشاهده می‌کنید.

در فایل Whiplash.html که به شما داده شده است، تعدادی تگ که نشان دهنده خط‌های عمودی جدول (vertical-lines)، ساعت‌های روز (time-box) و روزهای هفته (day-box) هستند نوشته شده‌اند و شما از آن‌ها به صورت آماده استفاده می‌کنید. یعنی کافی است صرفاً آن‌ها را در فایل خروجی که تولید می‌کنید بنویسید.

دسته آخری از تگ‌ها که در فایل Whiplash.html قرار دارند و برنامه شما وظیفه تولید آن‌ها را از روی داده‌های ورودی دارد به شکل زیر هستند:

```
<div class="record-box" style="width: 300px; left: 100px; top:360px; ">Eram</div>
```

مشابه جدولی که در بخش قبلی (۲) دیدید، باید با همان الگوریتم و اولویت‌ها، سانس‌هایی (بازه‌هایی) که تصمیم می‌گیرید به ازای هر روز نمایش دهید را انتخاب کنید. به ازای هر سانس که برای نمایش انتخاب می‌کنید، باید تگی مشابه بالا در فایل خروجی html. خود بنویسید که مقادیر آن از روی روابط زیر محاسبه می‌شوند:

$$width = 50 \times \frac{hh:mm}{00:30}$$

که hh:mm مدت زمان فیلم (اختلاف ساعت پایان و شروع) است

$$left = 100 + 50 \times \frac{hh:mm}{00:30}$$

که hh:mm زمان آغاز فیلم است

$$top = 60 + dayNum \times 50$$

که dayNum شماره روز است. روزها از صفر (۰) شماره‌گذاری می‌شوند. یعنی شنبه روز صفرم، یکشنبه روز یکم، ... و جمعه روز ششم است.

در مثال بالا، نام Eram برای سینمای متناظر با آن سانس (بازه) نوشته شده است که در برنامه، شما باید نام سینمای متناسب با آن سانس را بنویسد.
فایل Whiplash.html که به شما داده شده است، در واقع همان خروجی است که باید به ازای دستور

```
GET SCHEDULE Whiplash
```

روی CSV داده شده تولید کنید.

نحوه دریافت فایل ورودی

آدرس فایل ورودی در ابتدا توسط آرگومان‌های خط فرمان به برنامه داده می‌شود. برای آشنایی با آرگومان‌های خط فرمان به پیوست مراجعه کنید.
برای مثال اگر نام فایل اجرایی شما a.out باشد، برنامه با دستور زیر اجرا خواهد شد:

```
./a.out ./schedule.csv
```

نحوه تحویل

- کد خود را در قالب یک فایل با نام A3-SID.cpp در صفحه eLearn درس بارگذاری کنید که SID شماره دانشجویی شماست؛ برای مثال اگر شماره‌ی دانشجویی شما ۸۱۰۱۹۸۹۹۹ باشد، نام پرونده شما باید A3-810198999.cpp باشد که شامل کد شما است.
- 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.

- تمیزی کد، ذخیره کردن اطلاعات در ساختارهای مناسب، شکستن مرحله به مرحله مسئله و طراحی مناسب، در کنار تولید خروجی دقیق و درست، بخش مهمی از نمره شما را تعیین خواهد کرد.
- درستی برنامه شما از طریق آزمون‌های خودکار سنجیده می‌شود؛ بنابراین پیشنهاد می‌شود که با استفاده از ابزارهایی مانند diff خروجی برنامه خود را با خروجی‌هایی که در اختیارتان قرار داده شده است مطابقت دهید.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.

نکات پایانی

- در این تمرین اجازه استفاده از شیء‌گرایی را ندارید.
- تضمین می‌شود که نام فایل از طریق آرگومان‌های خط فرمان به شما داده می‌شود و همچنین این فایل به فرمت گفته شده وجود دارد.
- استفاده از کتابخانه‌های موجود برای تجزیه فایل CSV قابل قبول نیست.

پیوست

آرگومان‌های خط فرمان:

آرگومان‌های خط فرمان آرگومان‌هایی هستند که سیستم‌عامل در زمان اجرای برنامه آن‌ها را به برنامه انتقال می‌دهد. برنامه می‌تواند آن‌ها را نادیده بگیرد و یا از آن‌ها استفاده کند.

برای استفاده از این آرگومان‌ها، تابع main باید به صورت زیر نوشته شود:

```
int main(int argc, char* argv[ ])
```

دو آرگومان تابع را می‌توان برای دسترسی به آرگومان‌های خط فرمان استفاده کرد:

● argc

عدد صحیح؛ تعداد آرگومان‌های خط فرمان داده شده به برنامه
این مقدار حداقل برابر با یک است؛ زیرا دستور اجرای برنامه (نام پرونده اجرایی) حتماً در زمان اجرای برنامه مورد استفاده قرار می‌گیرد و همواره به عنوان آرگومان‌های خط فرمان شماره صفر به برنامه داده می‌شود.

● argv

آرایه‌ای از رشته‌های مدل زبان C؛ آرگومان‌های خط فرمان داده شده به برنامه

به عنوان یک مثال ساده برنامه زیر را در نظر بگیرید:

```
#include <iostream>

int main(int argc, char *argv[])
{
    std::cout << "There are " << argc << " arguments:" << std::endl;

    // Loop through each argument and print its number and value
    for (int count=0; count < argc; ++count)
        std::cout << count << " " << argv[count] << std::endl;

    return 0;
}
```

اگر برنامه به شکل

```
./a.out myFile.txt 100
```

اجرا شود، خروجی زیر تولید می‌شود:

There are 3 arguments:

0 ./a.out

1 myFile.txt

2 100

برای آشنایی بیشتر با نحوه کار آرگومان‌های خط فرمان می‌توانید به این [لینک](#) مراجعه کنید.