



### مقدمه

هدف از این تمرین آشنایی شما با طراحی بالا به پایین<sup>۱</sup> 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 می‌توانید به ویدیویی که در بخش محتوای دستیاران آموزشی در صفحه درس قرار گرفته مراجعه کنید.

### برنامه هفتگی درسی

هرساله برای شروع سال تحصیلی، برنامه هفتگی به دانش آموزان ارائه می‌شود. در این مدرسه تعدادی معلم وجود دارد که هریک می‌تواند چندین درس مشخص را ارائه دهد؛ همچنین این معلم‌ها روزهای مشخصی در مدرسه حضور دارند و امکان حضور همه روزه را ندارند. هر یک از دروس باید در دو روز ارائه شود و مدت زمان ارائه هر درس در هر روز ۱.۵ ساعت می‌باشد (هر درس جمعا ۳ ساعت در هفته). دقت کنید زمان شروع و پایان ارائه درس در دو روز یکسان است. با توجه به میزان تمرکز دانش آموزان در ساعات مختلف روز، هر درس باید تنها در بازه مشخصی از روز تدریس شود. برنامه شما با الگوریتمی که توضیح داده خواهد شد، زمان‌بندی مناسب برای ارائه دروس پایه دهم این مدرسه که شامل دو کلاس است را انجام می‌دهد.

اطلاعات لازم درباره‌ی دروس و معلم‌ها به عنوان اطلاعات ورودی به شما داده می‌شود.

### معلم

برای هر معلم نام او، دروسی که می‌تواند ارائه دهد و همچنین روزهایی که آزاد است تا به مدرسه بیاید به عنوان ورودی داده می‌شود.

<sup>۱</sup> Top-Down Design

## درس

برای هر درس نام آن، دو روز هفته که باید در آن روزها ارائه شود و بازه زمانی در روز که امکان برگزاری دارد در ورودی داده می‌شود.

## نحوه بدست آوردن زمان بندی مناسب

ساعت کاری مدرسه از شنبه تا چهارشنبه از ساعت 7:30 تا 13 می‌باشد؛ هر روز حداکثر سه درس در بازه‌های 1.5 ساعته برگزار می‌شود و بین هر دو درس یک زنگ تفریح نیم ساعته وجود دارد. دقت کنید ممکن است در یک بازه زمانی درسی برای ارائه یافت نشود، در این صورت زنگ تفریح‌ها همچنان برقرار است. (ساعات کلاس‌ها شامل 7:30 تا 9، 9:30 تا 11 و 11:30 تا 13 است؛ همچنین زنگ تفریح‌ها ساعت 9 تا 9:30 و 11 تا 11:30 است).

برنامه زمان بندی به این صورت کار می‌کند که از شنبه ساعت 7:30 برای کلاس اول شروع می‌کنیم و درسی که قابلیت ارائه در آن زمان دارد را به همراه معلم متناسب با آن انتخاب می‌کنیم. در صورتی که بیشتر از یک معلم برای درس و زمان مشخص شده پیدا شد، معلمی را انتخاب می‌کنیم که تعداد روزهای کمتری در مدرسه حضور دارد؛ همچنین اگر تعداد روزهای یکسانی را حضور داشتند، معلمی که نام او از لحاظ ترتیب الفبایی کوچکتر باشد، در اولویت است. در نهایت اگر بیش از یک درس به همراه معلم متناسب با آن برای یک بازه زمانی پیدا شد، درسی را انتخاب می‌کنیم که نامش از لحاظ ترتیب الفبایی کوچکتر باشد. ابتدا کل هفته را برای کلاس اول برنامه‌ریزی کنید و سپس به سراغ کلاس دوم بروید. توجه داشته باشید که اگر یک معلم برای یک کلاس انتخاب شود، بازه زمانی برگزاری کلاس از زمان آزاد او حذف می‌شود و در آن زمان نمی‌تواند برای کلاس دیگر انتخاب شود.

## قالب اطلاعات ورودی

اطلاعات معلم‌ها و دروس در ورودی استاندارد<sup>2</sup> به برنامه داده می‌شود. اطلاعات ورودی شامل دو بخش است که بخش اول مربوط به معلم‌ها و بخش دوم مربوط به درس‌ها است. در خط اول تعداد معلم‌ها (n) داده می‌شود و در n خط بعدی اطلاعات هر معلم با قالب زیر وارد می‌شود:

<teacher_name> <number_of_free_days> <free_days> <number_of_courses> <courses>
--

<sup>2</sup> Standard Input

دقت کنید free\_days نام روزهایی از هفته است که معلم آزاد است، همچنین courses دروسی هستند که معلم قادر به تدریس آنها است و این اطلاعات با فاصله<sup>3</sup> از هم جدا شده‌اند.

بعد از اتمام اطلاعات معلم‌ها، تعداد دروس (k) داده می‌شود. در k خط بعدی نیز اطلاعات هر درس با قالب زیر وارد می‌شود:

```
<course_name> <day1> <day2> <start_time> <end_time>
```

start\_time و end\_time به فرمت hh:mm هستند و نشان‌دهنده‌ی شروع و پایان بازه زمانی مشخصی از روز است که درس می‌تواند در آن ارائه شود. همچنین day1 و day2 روزهایی از هفته هستند که درس باید ارائه شود.

## قالب خروجی

برای هر درس ابتدا در یک خط نام آن درس و دو خط بعدی به ترتیب برنامه درسی کلاس اول و کلاس دوم مطابق قالب زیر چاپ می‌شود. دقت کنید که درس‌ها به ترتیب حروف الفبا در خروجی چاپ می‌شوند.

```
<course1_name>
<teacher_name>: <start_time> <end_time>
<teacher_name>: <start_time> <end_time>
<course2_name>
<teacher_name>: <start_time> <end_time>
<teacher_name>: <start_time> <end_time>
...
```

start\_time و end\_time به فرمت hh:mm هستند و نشان‌دهنده‌ی شروع و پایان بازه زمانی مشخصی از روز است که درس در آن زمان ارائه می‌شود. در صورتی که امکان ارائه یک درس برای یک کلاس وجود نداشته، به جای کل خط مربوط به آن کلاس، عبارت Not Found چاپ می‌شود.

---

<sup>3</sup> Space

## ورودی و خروجی نمونه

توجه کنید که برای اطمینان کامل از عملکرد برنامه‌تان لازم است خودتان آزمون‌های بیشتری طراحی کنید.  
نمونه ۱:

ورودی	خروجی
2 Hamid 3 Saturday Monday Wednesday 2 Math Physics Amin 3 Saturday Sunday Monday 2 Math Science 3 Math Saturday Monday 07:30 10:00 Physics Saturday Wednesday 08:00 11:00 Science Monday Sunday 07:30 13:00	Math Amin: 07:30 09:00 Hamid: 07:30 09:00 Physics Hamid: 09:30 11:00 Not Found Science Amin: 09:30 11:00 Amin: 11:30 13:00

زمانبندی را از شنبه ساعت 07:30 برای کلاس اول شروع می‌کنیم. در این زمان تنها Math می‌تواند ارائه شود و دو استاد Hamid و Amin می‌توانند آن را در شنبه و دوشنبه ارائه دهند و تعداد روزهای آزاد هر دو یکسان است ولی Amin از نظر الفبایی کوچکتر است و این استاد را برای کلاس اول انتخاب می‌کنیم. سپس به سراغ زمان 9:30 تا 11:00 شنبه برای کلاس اول می‌رویم، در این زمان تنها درسی که می‌تواند ارائه شود Physics است و استاد آن Hamid خواهد بود زیرا در این زمان و در روزهای شنبه و چهارشنبه آزاد است، بعد از آن زمان 11:30 تا 13:00 شنبه را بررسی می‌کنیم که درسی برای آن یافت نمی‌شود و همین کار را ادامه می‌دهیم تا به زمانی برسیم که بتوان برای آن درس و معلمی اختصاص داد؛ این زمان 09:30 تا 11:00 روز یکشنبه است که درس Science در آن زمان ارائه می‌شود و معلم آن Amin است. حال به سراغ کلاس دوم می‌رویم و زمانبندی را از 07:30 شنبه آغاز می‌کنیم که در این زمان تنها درسی که می‌تواند ارائه شود Math است و چون این زمان برای Amin پر شده است، Hamid را به عنوان معلم انتخاب می‌کنیم. زمان 09:30 تا 11:00 روز شنبه برای Hamid پر شده است پس نمی‌توان زمانی برای ارائه درس Physics برای کلاس دوم پیدا کرد و عبارت Not Found چاپ می‌شود؛ زمان بعدی که می‌توان درسی برای کلاس دوم اختصاص داد 11:30 تا 13:00 روز یکشنبه است که به درس Science با تدریس Amin اختصاص می‌یابد.

## 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 شماره دانشجویی شما ۸۱۰۱۰۱۰۰۰ باشد، نام پرونده شما باید A3-810101000.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 ورودی های آزمون اجرا شود.
- در این تمرین اجازه استفاده از شیءگرایی و makefile را ندارید.
- 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 آزمون های خودکار سنجیده می شود؛ بنابراین پیشنهاد می شود که با استفاده از ابزارهایی مانند diff خروجی برنامه خود را با خروجی هایی 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.