



### 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.

### سیستم نوبت‌دهی بیماران

امروزه بیمارستان‌ها از شلوغ‌ترین مراکز شهرهای بزرگ شناخته می‌شوند، که صدها نفر به آن‌ها مراجعه می‌کنند. هر بیمارستان برای نوبت‌دهی بیماران خود از مکانیزمی خاص استفاده می‌کند. به عنوان مثال، در بیمارستان گردو، به دلیل تغییرات هفتگی برنامه پزشکان بیمارستان، نوبت‌دهی به این صورت انجام می‌شود که بیماران در طول هفته به بیمارستان مراجعه می‌کنند تا برای هفته بعد نوبت بگیرند. در پایان هفته، سیستم پیامکی بیمارستان نوبت هر بیمار را به او اطلاع می‌دهد. در این تمرین شما باید با توجه به زمان مراجعه بیماران و اطلاعات پزشکان بیمارستان، سیستم نوبت‌دهی این بیمارستان را پیاده‌سازی کنید.

---

<sup>1</sup> Top-Down Design

## اطلاعات ورودی

- اطلاعات اولیه مربوط به بیماران و پزشکان به صورت سه فایل CSV به شما داده می‌شود. در فایل‌های CSV یا Comma-separated Values هر سطر دارای تعدادی داده است که با کاراکتر کاما (,) از هم جدا شده‌اند. در سطر اول این فایل‌ها، عنوان هر داده در سطرهای پیش رو با کاما از هم جدا شده‌اند. در هر فایل روزهای هفته به صورت کوتاه شده (Sat, Sun, Mon, Tue, Wed, Thu, Fri) نوشته می‌شوند. ساعات مربوط به هر قسمت در فایل‌ها نیز به صورت عدد صحیحی بین 0 تا 23 خواهد بود.
- **فایل مربوط به بیماران (patients.csv):** در این فایل در هر خط نام بیمار، مشکل بیمار و زمان مراجعه (به صورت weekday-hour) به فرمت زیر به شما داده می‌شود:

Name, Medical Issue, Appointment Time
MobinaDehghan, High blood pressure, Mon-8

- **فایل مربوط به پزشکان (doctors.csv):** در این فایل در هر خط نام پزشک، تخصص او، هزینه ویزیت، میانگین زمان انتظار (برحسب دقیقه)، مدت زمان ویزیت هر بیمار (برحسب دقیقه) و در انتها، روزهای حضور پزشک در بیمارستان به همراه ساعت ورود و خروج نیز داده می‌شود. زمان حضور پزشک در بیمارستان در هر روز به فرمت weekday-arrival-departure است که روزها با کاراکتر (\$) از هم جدا می‌شوند و مدت زمان حضور پزشک در بیمارستان نسبت به زمان ویزیت هر بیمار تقسیم‌پذیر است. تضمین می‌شود که پزشک حداقل یک روز در بیمارستان حضور داشته باشد و همچنین هر روز هفته بیش از یک بار عنوان نشود.

Name, Specialty, Visit Fee, Visit Duration, Wait Time Average, Days Available
MohsenMoghadam, Cardiology, 200, 20, 30, Sat-7-10\$Mon-8-13\$Wed-15-20
HadiZarrin, Orthopedics, 250, 15, 45, Sun-6-11

- **فایل مربوط به بیماری‌ها (diseases.csv):** در این فایل به ازای هر تخصص، بیماری‌هایی که باید به پزشک دارای آن تخصص ارجاع داده شوند به شما داده می‌شود. بیماری‌ها با کاراکتر (\$) از هم جدا می‌شوند. تضمین می‌شود که برای هر تخصص حداقل یک متخصص در لیست پزشکان موجود باشد و تخصص همه پزشکان در این فایل عنوان شوند؛ همچنین در بین بیماری‌های هیچ دو تخصصی مورد مشترکی نباشد و بیماری مراجعه‌کنندگان نیز از بین بیماری‌های این فایل باشد.

Specialty, Diseases
Endocrinology, Diabetes\$Thyroid\$Obesity

توجه داشته باشید که در هر سه فایل، هر داده ممکن است حاوی کاراکترهایی مانند فاصله (space) باشد. وظیفه شما تشخیص هر داده بر اساس کاراکترهای جداکننده مشخص شده است و بقیه کاراکترها نقش جداکننده ایفا نمی‌کنند.

## نحوه نوبت‌دهی

هر بیمار فقط می‌تواند توسط پزشک متخصص مربوط به بیماری‌اش ویزیت شود (این اطلاعات از طریق فایل diseases.csv به شما داده شده است).

بیماران به ترتیب زمان مراجعه، نوبت‌دهی می‌شوند و اولین نوبت خالی را دریافت می‌کنند؛ اگر زمان مراجعه چند بیمار با یکدیگر یکسان باشد نوبت‌دهی بر اساس اولویت حروف الفبا انجام می‌شود.

در صورت وجود چند پزشک واجد شرایط (داشتن تخصص لازم و اولین نوبت خالی)، پزشک با کمترین هزینه ویزیت انتخاب می‌شود. اگر هزینه چند پزشک برابر بود، پزشک با کمترین زمان انتظار انتخاب می‌شود و اگر زمان انتظار چند پزشک با یکدیگر برابر بود پزشکان بر اساس اولویت حروف الفبا اولویت‌بندی می‌شوند.

## قالب خروجی

کد شما باید برای هر بیمار نام او، نام پزشک، روز نوبت (به صورت کوتاه شده)، شماره نوبت، زمان نوبت و هزینه ویزیت را نمایش دهد. اگر زمان خالی در آن هفته وجود نداشت، عبارت: "No free time" نمایش داده می‌شود. توجه کنید که ترتیب نمایش بیماران در خروجی باید بر اساس حروف الفبا باشد.

```
Name: <patient_name>
Doctor: <doctor_name>
Visit: <visit_day> <visit_number> <visit_time>
Charge: <visit_charge>
-----
Name: <patient_name>
Doctor: <doctor_name>
Visit: <visit_day> <visit_number> <visit_time>
Charge: <visit_charge>
-----
Name: <patient_name>
No free time
```

پس از نمایش اطلاعات هر بیمار، در خط بعدی 10 عدد کاراکتر dash (-) چاپ شده و سپس اطلاعات بیمار بعدی نمایش داده می‌شود (به جز بیمار آخر).

- <visit\_number> نشان‌دهنده شماره نوبت بیمار بین همه بیماران آن **دکتر (در روز نوبت)** می‌باشد (با شروع از 1).
- <visit\_time> نشان‌دهنده زمان ویزیت به فرمت hh:mm است.
  - **hh**: عددی دو رقمی که نمایانگر ساعت مورد نظر است. این عدد می‌تواند بین 00 تا 23 باشد.
  - **mm**: عددی دو رقمی که نمایانگر دقیقه مورد نظر است. این عدد می‌تواند بین 00 تا 59 باشد.

## نمونه خروجی

```
Name: KasraAmiri
Doctor: MohsenDehghan
Visit: Mon 2 09:20
Charge: 200
-----
Name: HanieOsati
Doctor: FatemeNajafi
Visit: Sat 11 15:00
Charge: 150
-----
Name: MahdiMoghimi
No free time
```

## نکات و نحوهٔ تحویل

- کد خود را در قالب یک فایل با نام A3-SID.cpp در صفحهٔ eLearn درس بارگذاری کنید که SID شمارهٔ دانشجویی شماست؛ برای مثال اگر شمارهٔ دانشجویی شما ۸۱۰۱۰۱۰۰۰ باشد، نام پروندهٔ شما باید A3-810101000.cpp باشد که شامل کد شما است.
- برنامهٔ شما باید در سیستم عامل لینوکس و با مترجم g++ با استاندارد c++20 ترجمه و در زمان معقول برای ورودی‌های آزمون اجرا شود.
- در این تمرین اجازه استفاده از شیءگرایی و makefile را ندارید.
- درستی برنامهٔ 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.

## نمرات

- تمیزی کد
  - رعایت کردن نام‌گذاری صحیح و انسجام<sup>2</sup>
  - عدم وجود کد تکراری
  - رعایت دندانه‌گذاری<sup>3</sup>
  - عدم استفاده از متغیرهای گلوبال
  - استفاده صحیح از متغیرهای ثابت به جای Magic Value-ها
- درستی کد
  - آزمون‌های خودکار
- طراحی
  - شکستن مناسب و مرحله به مرحله مسئله
  - ذخیره اطلاعات در ساختارهای مناسب
  - ساختاردهی کد در قالب توابع کوتاه که فقط یک کار را انجام می‌دهند

دقت کنید که موارد ذکر شده لزوماً کل نمره شما را تشکیل نمی‌دهند و ممکن است با تغییراتی همراه باشند.

---

<sup>2</sup> Consistency

<sup>3</sup> Indentation