

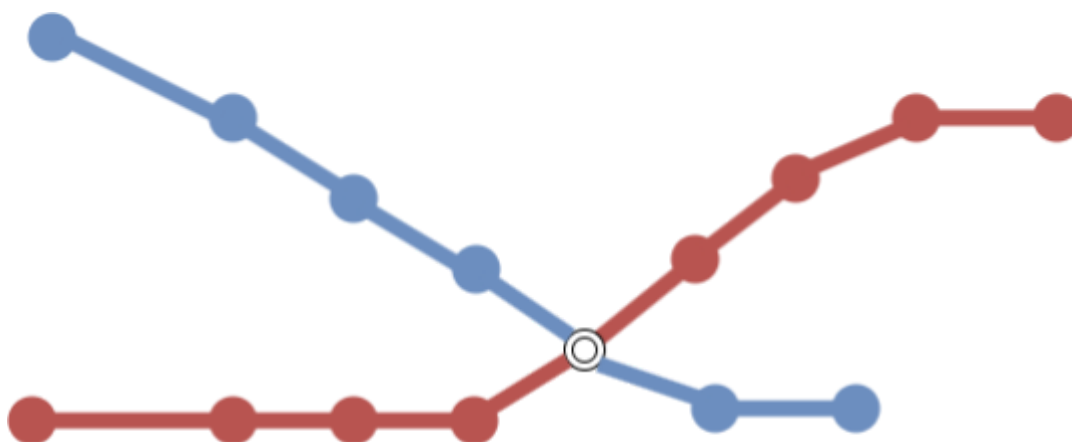


## مقدمه

هدف از این تمرین آشنایی شما با رشته‌ها و وکتورها است. در این تمرین شما یک برنامه‌ی مسیریاب را طراحی می‌کنید که مسافران را به خطوط درست در تقاطع مترو هدایت می‌کند. علاوه بر اجرای درست برنامه، جدا کردن مسئولیت‌ها به تابع‌های مختلف و تمیزی کد اهمیت زیادی دارند. سعی کنید در پیاده‌سازی تمرین، توابع مختلفی تعریف کنید که هر کدام تنها یک کار انجام می‌دهند.

## مسیریاب مترو

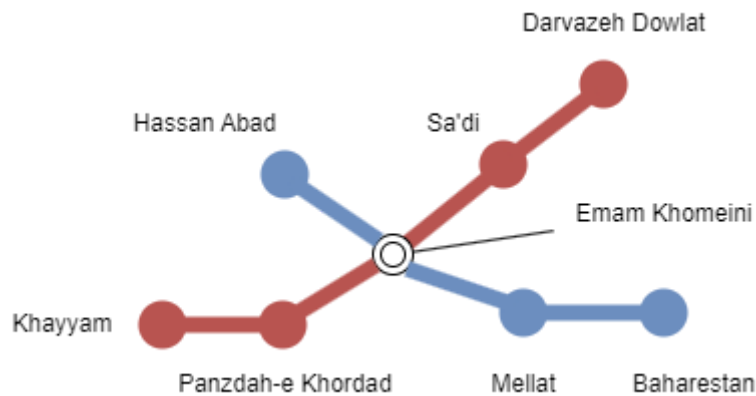
هر روز هزاران و شاید حتی میلیون‌ها مسافر از متروی تهران برای رسیدن به مقصدشان استفاده می‌کنند. همواره تعداد زیادی مسافر جدید به مترو اضافه می‌شود که راه و چاه ایستگاه‌ها را بلد نیستند. از طرفی امروزه خواندن نقشه‌ی مترو برای عموم مردم زمان‌بر بوده و اکثر این مسافران حوصله یا وقت کافی برای خواندن آن را ندارند. شهرداری تهران قصد ساختن برنامه‌ای را دارد که با وارد کردن نام ایستگاه مقصد، خطی که مسافر باید سوار شود را به همراه هزینه و زمان سفر به او نشان دهد. برای ساده شدن مسئله فرض کنید که نقشه مترو همیشه شامل یک تقاطع بوده که این تقاطع نقطه شروع هر مسافر است.



یک نقشه نمونه از ورودی برنامه. مسافر همواره در تقاطع واقع در مرکز تصویر ایستاده است.

## یافتن مسیر

تمام خطوطی که این تقاطع را شامل می‌شوند، یک نقطه‌ی شروع و یک نقطه‌ی پایان دارند. در ابتدای برنامه، لیست ایستگاه‌های سمت شروع و پایان هر خطِ مترو تا بر اساس آن بتوان مقصد هر مسافر را پیدا کرد. به نقشه زیر توجه کنید:



در این نقشه، ایستگاه‌های خط قرمز به سمت پایان [Sa'di, Darvazeh Dowlat] و به سمت شروع، [Khayyam, Panzadah-e Khordad] می‌باشند. مسافر همواره در تقاطع نقشه ایستاده و برنامه باید در جواب سوال مسافر، به او بگوید که به سمت شروع یا پایان کدام خط سوار شود. برای مثال اگر مسافری در نقشه فوق بخواهد به سمت ایستگاه Mellat برود، باید به سمت پایان خط آبی سوار شود.

## محاسبه زمان سفر

علاوه بر مسیریابی، مسافران زمان تقریبی سفر خود را نیز باید بدانند. فاصله زمانی رسیدن مترو در ایستگاه تقاطع (ایستگاه مبدا مسافر) به همراه فاصله زمانی ایستگاه‌های هر خط به شما داده می‌شود. وظیفه‌ی برنامه این است که به کمک داده‌های ورودی و زمان فعلی، زمان رسیدن مسافر به ایستگاه مقصد را به شکل ساعت و دقیقه نمایش دهد. برای مثال اگر هر بیست دقیقه یک مترو به ایستگاه تقاطع بیاید، فاصله Emam Khomeini تا Sa'di ده دقیقه، Sa'adi تا Darvazeh Dowlat پنج دقیقه و زمان فعلی ۱۶:۱۵ باشد، مترو ساعت ۱۶:۲۰ وارد ایستگاه Emam Khomeini شده و ۱۶:۳۵ به ایستگاه Darvazeh Dowlat می‌رسد. لازم به ذکر است که زمان حرکت اولین قطار از ایستگاه تقاطع (ایستگاه فعلی) برابر با ۶:۰۰ است و مسافر تنها در زمان‌هایی که مترو باز است سفر می‌کند (حالتی که مترو تعطیل شود و مسافر مجبور به انتظار تا فردا باشد در این برنامه در نظر گرفته نخواهد شد). علاوه بر آن در صورتی که زمان ورود قطار دقیقاً برابر با زمان فعلی بود، فرض کنید مسافر امکان سوار شدن در آن قطار را دارد.

## محاسبه قیمت سفر

قیمت سفر بر اساس تعداد ایستگاه‌هایی که مسافر طی می‌کند و توسط تابع زیر محاسبه می‌شود:

$$[1000 \times \log(10 \times n)]$$

که  $n$  برابر تعداد ایستگاه‌هایی است که رد می‌کند (به جز ایستگاه مبدأ). مثلاً هزینه مسافرت تا ایستگاه دروازه دولت برابر  $[1000 \times \log(20)] = 1302$  تومان خواهد بود.

## ورودی

در اولین خط ورودی  $n$  می‌آید که نشان دهنده تعداد خطوط در تقاطع فعلی است. در ادامه آن زمان فعلی در فرمت hh:mm می‌آید. مراحل بعدی برای هر خط مترو تکرار خواهد شد ( $n$  بار تکرار می‌شود). به ازای هر خط مترو، در خط اول ابتدا نام آن خط می‌آید. در ادامه آن یک عدد می‌آید که نشان می‌دهد هر چند دقیقه یک بار مترو در این خط می‌رسد (تضمین می‌شود این عدد عاملی از ۶۰ باشد). سپس در خط بعدی تعداد ایستگاه‌های آن به سمت شروع خط آمده و در ادامه آن، نام ایستگاه‌ها به آن سمت می‌آید؛ تضمین می‌شود نام ایستگاه‌ها شامل کاراکتر اسپیس نخواهند بود. قبل از نام هر ایستگاه، فاصله زمانی ایستگاه قبلی با ایستگاه کنونی در واحد دقیقه به شکل یک عدد صحیح نوشته می‌شود. در خط بعدی تعداد ایستگاه‌ها به سمت پایان و در ادامه آن نام ایستگاه‌ها به سمت پایان می‌آید؛ در این خط نیز قبل از نام هر ایستگاه فاصله زمانی آن با ایستگاه قبلی می‌آید. واضح است که در بین نام ایستگاه‌ها، نام ایستگاه فعلی یا همان ایستگاه تقاطع نمی‌آید.

پس از آنکه اطلاعات تمام خطوط به برنامه داده شد، درخواست مسافر که همان نام ایستگاه مقصد است، در یک خط وارد می‌شود. تضمین می‌شود که نام ایستگاه وارد شده معتبر و میان ایستگاه‌های موجود در برنامه خواهد بود. علاوه بر آن، نام ایستگاه مقصد و مبدأ، که همان ایستگاه تقاطع است، یکسان نخواهد بود.

## خروجی

در خط اول خروجی مسیری که مسافر باید طی کند را به صورت زیر چاپ کنید:

```
Towards [end | start] of [Line Name] in [n] station(s)
```

عبارت [end | start] با توجه به جهت حرکت مسافر جایگزین خواهد شد. در صورتی که به سمت شروع خط حرکت کند، عبارت start و در غیر این صورت عبارت end چاپ می‌شود. به جای Line Name نیز نام خط که در ورودی دریافت کردید باید چاپ شود. در نهایت به جای  $n$  باید تعداد ایستگاه‌های مانده تا مقصد را چاپ کنید.

در خط دوم زمان رسیدن به مقصد و در خط سوم نیز هزینه این سفر چاپ می‌شود. فرمت زمان باید بیست و چهار ساعته و به صورت hh:mm باشد؛ دو رقم چپ برای ساعت و دو رقم راست برای دقیقه. هزینه سفر نیز باید به صورت یک عدد صحیح و بدون رقم اعشار چاپ شود.

## مثال

### ورودی نمونه اول

```
2 13:25
Ghermez 5
2
5 Panzdahe_e_Khordad 5 Khayyam
2
5 Sa'adi 10 Darvazeh_Dowlat
Abi 10
1
15 Hassan_Abad
2
10 Mellat 5 Baharestan
Baharestan
```

### خروجی نمونه اول

```
Towards end of Abi in 2 station(s)
13:45
1302
```

### ورودی نمونه دوم

```
2 16:08
Zard 12
4
10 Doctor_Habibollah 5 Ostad_Mo'in 7 Meydan_e_Azadi 4 Bimeh
2
4 Towhid 8 Enghelab
Abi 6
3
6 Daneshgah_e_Sharif 3 Tarasht 5 Sadeghiye
2
7 Navab 5 Meydan_h_Hor
```

Tarasht

**خروجی نمونه دوم**

Towards start of Abi in 2 station(s)

16:21

1302

# نکات و نحوه تحویل

- کد خود را در قالب یک فایل با نام A1-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 A1-810101000.cpp باشد که شامل کد شما است.
- برنامه شما باید در سیستم عامل لینوکس و با مترجم g++ با استاندارد c++20 ترجمه و در زمان معقول برای ورودی های آزمون اجرا شود.
- برای پیاده سازی ساختار داده زمان می توانید از کتابخانه chrono استفاده کنید. اجباری در استفاده از آن نبوده و می توانید از ساختار داده های خودتان نیز استفاده کنید.
- 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.

## نمرات

- تمیزی کد
  - رعایت کردن نام گذاری صحیح و انسجام<sup>1</sup>
  - عدم وجود کد تکراری
  - رعایت دندانه گذاری<sup>2</sup>
  - عدم استفاده از متغیرهای گلوبال
  - استفاده صحیح از متغیرهای ثابت<sup>3</sup> و Magic Value-ها
  - استفاده صحیح از وکتور و string به جای آرایه و char\*
  - توابع کوتاه که فقط یک کار را انجام می دهند
- درستی کد
  - آزمون های خودکار

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

---

<sup>1</sup> Consistency

<sup>2</sup> Indentation

<sup>3</sup> Constant