



مقدمه

هدف از این تمرین آشنایی شما با ورودی و خروجی و کار با رشته‌ها و بردارها در زبان C++ به عنوان مهارت‌های پایه‌ای برنامه‌نویسی در این زبان است. این مهارت‌ها پیاده‌سازی پروژه‌های بزرگتر را امکان‌پذیر می‌کنند. در این تمرین شما یک برنامه‌ی هماهنگی جلسات را پیاده‌سازی می‌کنید. علاوه بر درستی برنامه، تمیزی کد و جدا کردن برنامه به توابع مختلف اهمیت زیادی دارند. سعی کنید که قبل از انجام تمرین حتماً ویدیوهای بخش تمیزی کد موجود در صفحه‌ی درس را مشاهده کنید.

هماهنگی جلسات

در این تمرین شما به پیاده‌سازی یک سامانه‌ی هماهنگی جلسات می‌پردازید. در این سامانه ابتدا تعدادی بازه‌ی زمانی به عنوان انتخاب‌ها ثبت می‌شود و سپس افراد مختلف رای خود را در رابطه با زمان‌هایی که برایشان مناسب است اعلام می‌کنند. در نهایت نیز سامانه لیستی از بهترین بازه‌های زمانی برای برگزاری جلسه را در خروجی چاپ می‌کند.

بازه‌های زمانی

در واقعیت بازه‌های زمانی به صورت زمان شروع و پایان تعیین می‌شود اما در این تمرین شما برای راحتی کار، آن‌ها را با یک عدد صحیح یکتا که از صفر شروع می‌شود مشخص می‌کنید. در خط اول ورودی با عدد صحیح n تعداد این بازه‌های زمانی مشخص می‌شود و بازه‌های با شناسه‌های یکتا از 0 تا $n-1$ در ادامه‌ی برنامه استفاده می‌شوند.

شرکت‌کنندگان و رای‌هایشان

نام تمام شرکت‌کنندگان در یک خط به برنامه داده می‌شود. در خط‌های بعدی رای‌های داده شده ذکر می‌شوند. هر رای می‌تواند سه حالت «می‌توانم»، «نمی‌توانم»، «اگر لازم باشد» داشته باشد که به ترتیب با YES و NO و IF_NECESSARY در ورودی مشخص می‌شوند. برای هر رای ابتدا شناسه‌ی بازه‌ی زمانی موردنظر مشخص می‌شود سپس یکی از سه انتخاب برای آن رای مشخص می‌شود. برای مثال در ورودی نمونه، علی به بازه‌ی زمانی با شناسه 0 رای «نمی‌توانم» و به بازه‌ی زمانی با شناسه 1 رای «می‌توانم» داده است.

دقت داشته باشید که هر نفر باید برای همه‌ی بازه‌های زمانی رای ثبت کرده باشد؛ پس اگر در ورودی رای‌های شرکت‌کننده‌ای ذکر نشده یا یک شرکت‌کننده به یک یا چند بازه‌ی زمانی رای نداده است باید رای «اگر لازم باشد» برای آن شرکت‌کننده در آن بازه‌ی زمانی ثبت شود. برای مثال در ورودی نمونه در نظر گرفته می‌شود که علی به بازه‌های زمانی شماره ۲ و ۳ رای «اگر لازم شد» داده است. همچنین در نظر گرفته می‌شود که اصغر به همه‌ی بازه‌های زمانی رای «اگر لازم باشد» داده است.

بررسی مطلوبیت بازه‌های زمانی

مطلوبیت بازه‌های زمانی به این صورت سنجیده می‌شود که بازه‌ی زمانی که کمترین تعداد رای «نمی‌توانم» را داشته باشد مطلوب‌ترین بازه‌ی زمانی است و اگر دو بازه‌ی زمانی تعداد رای «نمی‌توانم» یکسانی داشته باشند، آن بازه‌ی زمانی که رای «اگر لازم باشد» کمتری داشته باشد مطلوب‌تر است. اگر دو بازه‌ی زمانی همه‌ی رای‌هایشان یکسان باشد آن بازه‌ی زمانی که شناسه‌ی کمتری دارد مطلوب‌تر است.

ورودی و خروجی

قالب ورودی

در خط اول ورودی عدد صحیح n می‌آید که نشان‌دهنده‌ی تعداد بازه‌های زمانی است. در خط بعدی ورودی نام افراد رای‌دهنده که با فاصله از هم جدا شده‌اند ذکر می‌شود. در خط‌های بعدی ورودی رای‌ها مشخص می‌شود.

قالب خروجی

خروجی کد شما شامل حداکثر ۳ بازه‌ی زمانی است که به ترتیب از بیشترین به کمترین مطلوبیت چاپ شده‌اند. قالب چاپ شدن این بازه‌ها در بخش ورودی و خروجی نمونه مشخص است. **دقت داشته باشید** که هر جا لازم است شرکت‌کنندگان را چاپ کنید به همان ترتیبی که در ورودی آمده‌اند چاپ کنید.

ورودی و خروجی نمونه

| قالب خروجی | قالب ورودی |
|---|---|
| Time slot <time_slot_id>: YES: <participants...> NO: <participants...> IF_NECESSARY: <participants...> ### ... | <time_slots_count> <participant1> <participant2> ... <participant>:<time_slot_id> <status> |

| خروجی | ورودی |
|--|--|
| Time slot 2: YES: fateme NO: IF_NECESSARY: ali akbar asghar ### Time slot 3: YES: NO: IF_NECESSARY: ali fateme akbar asghar ### Time slot 1: YES: ali | 4 ali fateme akbar asghar ali:0 NO 1 YES akbar:1 NO fateme:2 YES 0 NO 1 IF_NECESSARY |

| | |
|--|---|
| | NO: akbar IF_NECESSARY: fateme asghar ### |
|--|---|

نحوه تحویل

- کد خود را در قالب یک فایل با نام A1-SID.cpp در صفحه‌ی eLearn درس بارگذاری کنید که SID شماره دانشجویی شماست؛ برای مثال اگر شماره‌ی دانشجویی شما ۸۱۰۱۹۹۰۰۰ باشد، نام پرونده کد شما باید A1-810199000.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، 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 از درستی کامل قالب خروجی برنامه خود اطمینان حاصل کنید و از دادن خروجی‌هایی که در صورت پروژه ذکر نشده است اجتناب کنید. برای مقایسه‌ی خروجی خود حتماً از توضیحات **پیوست ۱** استفاده کنید تا مشکلی در زمان تحویل در اجرای تست‌های شما پیش نیاید.
- در طول این تمرین ممکن است با مشکلاتی روبه‌رو شوید که راه حل آن‌ها را نمی‌دانید؛ در این صورت، جست‌وجوگرهایی مانند google و سایت‌هایی مانند **stackoverflow** و **cplusplus.com** ممکن است به شما کمک کنند.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.

پیوست ۱

مقایسه خروجی برنامه با خروجی مورد انتظار با چشم شاید برای برنامه‌های کوچک که خروجی کمی تولید می‌کنند و روند اجرای کوتاهی دارند میسر باشد، اما این کار برای برنامه‌های بزرگ‌تر با مسیر اجرای پیچیده کاری دشوار است. برای این کار می‌توان از ابزارهایی که در سیستم عامل لینوکس در دسترس است استفاده کرد. برای این کار با استفاده از مجموعه دستورات زیر خروجی را در فایل دلخواه ذخیره می‌کنیم.

```
g++ -std=c++11 source.cpp
```

```
./a.out < in.txt > out.txt
```

حال فرض کنیم که خروجی درست در فایل true_out.txt قرار داشته باشد، برای مقایسه‌ی این خروجی با خروجی out.txt که توسط برنامه‌ی ما تولید شد از ابزار **diff** استفاده می‌کنیم.

```
diff out.txt true_out.txt
```

اگر دو فایل یکسان باشند دستور **diff** هیچ خروجی‌ای تولید نمی‌کند. وگرنه، تفاوت‌های دو فایل را نمایش می‌دهد.

برای مطالعه‌ی بیشتر درباره‌ی diff می‌توانید از [این لینک](#) استفاده کنید.