



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. توجه کنید که برای گرفتن نمره‌ی این تمرین (و تمرین‌های آینده) باید علاوه بر بارگذاری فایل برنامه‌ی خود، در تحویل برخت با دستیاران هم حاضر شوید. (زمان‌های مختلف تحویل روی صفحه درس قرار می‌گیرد که می‌توانید زمان مناسب خود را از میان آن‌ها انتخاب کنید) 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](https://stackoverflow.com) و cplusplus.com ممکن است به شما کمک کنند.

تحویل‌های تصادفی (بر اساس یک داستان واقعی)

استاد درس مهندسی نرم‌افزار در ترم جاری از دانشجویان خواسته برای انجام تکلیف‌های درس گروه‌های سه نفره تشکیل دهند. در این درس تعدادی تکلیف به دانشجویان داده می‌شود که باید به طور گروهی انجام شوند. پس از انجام هر تکلیف، تعدادی از دانشجویان برای تحویل برخت انتخاب می‌شوند و یک روز پیش از تاریخ تحویل، فهرست آن‌ها اعلام می‌شود. نحوه انتخاب باید طوری باشد که هر یک از دانشجویان حداقل یک بار در ترم تحویل داشته باشد، اما بعضی دانشجویان بیش از یک تحویل خواهند داشت.

استاد درس قصد دارد از ابتدای ترم برنامه تحویل‌ها را تنظیم کند (بدون آن که به دانشجویان اعلام کند). بنابراین صورت مسئله این است که با داشتن فهرست دانشجویان درس و فهرست اعضای گروه‌ها برنامه تحویل را تولید کنیم. یعنی مشخص کنیم برای هر تکلیف، چه دانشجویانی باید در تحویل برخت شرکت کنند. برنامه تولید شده باید شرط‌های زیر را برآورده کند.

- شرط ۱. برای هر تکلیف هر گروه بیش از یک تحویل نداشته باشد.
- شرط ۲. ترتیب تحویل گروه‌ها در طول زمان به طور تصادفی تعیین شود و ارتباطی با ترتیب فهرست گروه‌ها و ترتیب اعضای گروه‌ها نداشته باشد (مثلاً به همان ترتیب یا عکس آن نباشد).
- شرط ۳. تمامی دانشجویان حداقل یک‌بار تحویل داشته باشند.

هدف شرط ۲ این است که دانشجویان نتوانند چیزی در مورد ترتیب تحویل‌ها پیش‌بینی کنند و عملاً مجبور باشند در انجام تمام تکلیف‌ها مشارکت داشته باشند و خود را برای تحویل احتمالی آماده کنند. به همین خاطر ضروری است در این برنامه‌ریزی از تولید اعداد تصادفی استفاده شود.

به عنوان مثال، فرض کنید تعداد تکلیف‌ها ۵ است و ۵ گروه سه نفره با مشخصات زیر داریم:

Team ID	Member 1	Member 2	Member 3
Team1	S11	S12	S13
Team2	S21	S22	S23
Team3	S31	S32	S33
Team4	S41	S42	S43
Team5	S51	S52	S53

یک برنامه تحویل مطلوب می‌تواند به این شکل باشد:

Assignment 1	S13	S22	S31	S53
Assignment 2	S11	S23	S42	S53
Assignment 3	S21	S33	S41	S51
Assignment 4	S12	S23	S33	S52
Assignment 5	S11	S32	S43	S51

مثالی از یک برنامه تحویل نامطلوب با نقض شرط ۲ (با قدری دقت می‌توان نظم موجود در برنامه را تشخیص داد):

Assignment 1	S11	S22	S33	S41
Assignment 2	S52	S13	S21	S32
Assignment 3	S43	S51	S12	S23
Assignment 4	S31	S42	S53	S11
Assignment 5	S22	S33	S41	S52

لازم به ذکر است برای تمام تکلیف‌ها تعداد تحویل‌ها یکسان است. برای مقید کردن این تعداد فرض کنید:

- تعداد گروه‌ها k
- تعداد تکلیف‌ها a
- تعداد تحویل‌ها برای هر تکلیف S

برای این که برنامه‌ریزی با شرایط ذکر شده ممکن باشد، اولاً S باید حداقل $\lceil 3k/a \rceil$ باشد تا بتوان از همه دانشجویان تحویل گرفت. برای این که بتوان حداقل یک تحویل اضافه داشت، این مقدار با عدد یک جمع خواهد شد. از طرف دیگر اگر S از k بیشتر باشد حتماً برای هر تکلیف حداقل یک تیم بیش از یک بار تحویل دارد که با شرط ۱ ناسازگار است. بنابراین می‌توانید فرض کنید در تمامی ورودی‌ها شرط $k \geq S \geq \lceil 3k/a \rceil + 1$ برقرار است.

قالب ورودی

در خط اول ورودی به ترتیب اعداد صحیح k و a و s می‌آیند که با فاصله از هم جدا شده‌اند. سپس در k خط بعد، در هر خط شناسه‌ی افراد گروه می‌آیند که با فاصله از هم جدا شده‌اند. فرض کنید $k \geq 1$ و $a \geq 1$. شناسه‌ی افراد از حروف بزرگ و کوچک انگلیسی و اعداد تشکیل شده است و در وسط شناسه فرد فاصله وجود ندارد.

قالب خروجی

خروجی کد شما باید دارای a خط باشد که در خط i ام، s شناسه وجود دارد که با فاصله از هم جدا شده‌اند است. این شناسه‌ها نشانگر دانشجویانی هستند که باید تکلیف i ام را به استاد تحویل دهند.

ورودی و خروجی نمونه

ورودی	خروجی
5 5 4 S11 S12 S13 S21 S22 S23 S31 S32 S33 S41 S42 S43 S51 S52 S53	S13 S22 S31 S53 S11 S23 S42 S53 S21 S33 S41 S51 S12 S23 S33 S52 S11 S32 S43 S51

توجه کنید که خروجی شما نباید دقیقاً معادل خروجی نمونه باشد، اما لازم است که قالب آن درست باشد و ۳ شرط ذکر شده در صورت مسئله در پاسخ شما برقرار باشند.

نحوه تحویل

- برای این که نحوه‌ی ترتیب دهی شما به تحویل‌ها را ارزیابی کنیم و با کد شما مقایسه کنیم، در این تمرین شما باید یک متن کوتاه در حد یک پاراگراف تهیه کنید که در آن توضیح دهید با چه روشی ترتیب تصادفی تحویل‌ها را تعیین و از ایجاد نظم قابل پیش‌بینی جلوگیری می‌کنید. طول این بخش باید حداکثر ۷ خط باشد.
- کد خود را در قالب یک فایل با نام A1-SID.cpp در کنار فایل pdf حاوی متن نحوه‌ی ترتیب‌دهی خود گذاشته و آن‌ها را در قالب یک فایل زیپ با نام A1-SID.zip فشرده کنید (نام فایل گزارش اهمیتی ندارد). این فایل زیپ را در صفحه eLearn درس بارگذاری کنید. SID شماره دانشجویی شماست؛ برای مثال اگر شماره‌ی دانشجویی شما ۸۱۰۱۹۹۰۰۰ باشد، نام پرونده کد شما باید A1-810199000.cpp باشد و نام پرونده بارگذاری شده A1-810199000.zip که شامل پرونده کد و pdf گزارش 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 آزمون های خودکار سنجیده می شود؛ بنابراین از درستی قالب خروجی برنامه خود اطمینان حاصل کنید.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.
- با وجود این که تابع `std::random_shuffle` ممکن است راه حلی مناسب برای به هم ریختن ترتیب یک بردار باشد، برای رعایت شرایط مساوی در بررسی طراحی کد، استفاده از این تابع و توابع پیش تعریف شده در دیگر کتابخانه ها ممنوع است و استفاده از آن باعث از دست دادن نمره ی بخش طراحی تمرین می شود.