



مقدمه

هدف از این تمرین آشنایی شما با رشته‌ها و کار با فایل‌ها است. در این تمرین شما یک برنامه‌ی دفترچه خاطرات ساده را پیاده‌سازی می‌کنید که خاطرات هر روز را ذخیره می‌کند و می‌تواند یادداشت‌های یک روز یا خلاصه‌ی روزهای خوب یا پراتفاق را به کاربر نشان دهد.

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.

دفترچه خاطرات

این برنامه با استفاده از تعدادی دستور که از ورودی استاندارد^۱ می‌گیرد، عملیات مورد انتظار را انجام می‌دهد. به این صورت که اگر کلمه‌ی اول یک خط ورودی یک کلیدواژه‌ی^۲ دستور باشد، آن‌گاه آن دستور اجرا می‌شود و در غیر این صورت، آن خط ورودی، متن یک یادداشت است که باید برای آن روز ذخیره شود.

کارکرد معمول برنامه به این صورت است که کاربر ابتدا شروع یک روز را با دستور شروع روز مشخص می‌کند و سپس متن اتفاقات روز را می‌نویسد که باید به یادداشت‌های آن روز اضافه شود. در این میان ممکن است کاربر دستور دیدن خلاصه‌ی پراتفاق‌ترین روز یا بهترین روز، یا دستور دیدن یادداشت‌های یک روز خاص را بدهد که برنامه باید جواب را در خروجی استاندارد^۳ چاپ کند.

برای پیدا کردن بهترین روز، این برنامه یک فایل از کلمه‌های مثبت به نام `positive-words.txt` را خوانده و مجموع تعداد تکرار هر یک از این کلمات را در یادداشت‌های هر روز می‌شمارد. سپس روزی را که بیشترین تعداد تکرار کلمات مثبت را دارد به عنوان بهترین روز انتخاب می‌کند.

دستورها

۱. شروع روز

`start_day day/month/year`

این دستور شروع یک روز را مشخص می‌کند. یادداشت‌هایی که از این به بعد در ورودی می‌آیند باید برای این روز ثبت شوند. این دستور چیزی در خروجی استاندارد چاپ نمی‌کند.

^۱ stdin

^۲ keyword

^۳ stdout

ورودی نمونه

ورودی	خروجی
start_day 24/9/2020 Today was boring Nothing happened	

۲. نشان دادن یادداشت‌های یک روز خاص

show_day day/month/year

با گرفتن این دستور، برنامه باید یادداشت‌های مربوط به آن روز را در خروجی استاندارد چاپ کند.

ورودی و خروجی نمونه

ورودی	خروجی
start_day 24/9/2020 First day of school I woke up late and it was a disaster start_day 25/9/2020 boring show_day 24/9/2020	First day of school I woke up late and it was a disaster

۳. نشان دادن پرتفاق‌ترین روز

show_the_longest_day

با گرفتن این دستور، برنامه باید در بین تمامی روزهایی که تا به حال مطلبی برای آن‌ها ذخیره شده، روزی که طول کاراکترهای یادداشت‌های مربوط به آن بیشتر از بقیه است را انتخاب کند و خلاصه‌ای از آن در خروجی استاندارد چاپ کند.

ورودی و خروجی نمونه

ورودی	خروجی
start_day 24/9/2020 Today was a long day I do not even know where to begin start_day 25/9/2020 boring show_the_longest_day	24/9/2020 Today was a long day...

۴. نشان دادن بهترین روز

show_the_best_day

با گرفتن این دستور، برنامه باید در بین تمامی روزهایی که تا به حال مطلبی برای آن‌ها ذخیره شده، روزی که در یادداشت‌های مربوط به آن، بیشترین تعداد تکرار کلمات مثبت موجود است را انتخاب کند و **خلاصه‌ای** از آن در خروجی استاندارد چاپ کند. اگر این عدد برای چند روز یکسان باشد قدیمی‌ترین روز انتخاب می‌شود. توجه کنید که در شمردن تعداد تکرار کلمات مثبت، آن کلمه باید عیناً در متن آمده باشد. به عنوان مثال عبارت happy در unhappy یا great در greatest نباید شمرده شوند.

ورودی و خروجی نمونه

خروجی	positive-words.txt	ورودی
24/9/2020 Today was an amazing...	amazing cool great happy fun	start_day 24/9/2020 Today was an amazing day I do not even know where to begin start_day 25/9/2020 boring show_the_best_day

فرمت خلاصه

در خلاصه ابتدا در یک خط تاریخ روز با فرمت `day/month/year` می‌آید. سپس اگر طول متن یادداشت‌های آن روز کمتر از ۲۰ کاراکتر باشد، تمامی متن در خلاصه می‌آید و در غیر این صورت ۲۰ کاراکتر اول متن به علاوه سه نقطه می‌آیند.

نمونه

خلاصه	یادداشت‌های روز ۲۴/۹/۲۰۲۰
24/9/2020 Today was an amazing...	Today was an amazing day I do not even know where to begin
24/9/2020 Boring day	Boring day

نحوهٔ تحویل

- کد خود را در قالب یک فایل با نام A1-SID.cpp در صفحهٔ eLearn درس بارگذاری کنید که SID شمارهٔ دانشجویی شماست؛ برای مثال اگر شمارهٔ دانشجویی شما ۸۱۰۱۹۸۹۹۹ باشد، نام پروندهٔ شما باید A6-810198999.cpp باشد که شامل کد شما است.
- برنامهٔ 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.
- تمیزی کد، شکستن مرحله به مرحله مسئله و طراحی مناسب، در کنار تولید خروجی دقیق و درست، بخش مهمی از نمرهٔ شما را تعیین خواهد کرد.
- درستی برنامهٔ 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. مطلبی در راستای آشنایی بیشتر شما با همین موضوع در صفحهٔ درس بارگذاری می شود.
- 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 ساختمان داده‌ی `std::map` برای نگه داشتن یادداشت های هر روز مناسب است، با توجه به این که هنوز درس داده نشده است، برای رعایت شرایط مساوی در بررسی طراحی کد، استفاده از این ساختمان داده ممنوع است و استفاده از آن باعث از دست دادن نمره‌ی بخش طراحی تمرین می شود.

نکات پایانی

- تضمین می شود که ورودی های آزمون ترتیب و داده‌ی درست دارند و باعث بروز خطا نمی شوند. مثلاً `show_day` برای روزی که `start_day` برای آن صدا زده نشده صدا زده نمی شود و... همچنین در ورودی ها متن یادداشت ها شامل علائم نگارشی نخواهند بود.
- هر سوالی درباره‌ی صورت تمرین دارید را در فروم مربوط به تمرین در صفحه‌ی eLearn درس پی‌رسید تا از سوال های تکراری جلوگیری شود.
- یک نمونه ورودی و خروجی کامل تر، در کنار `positive-words.txt` در قالب یک فایل زیپ از طریق صفحه‌ی eLearn درس در اختیار شما قرار داده می شود که می توانید برای اطمینان از درستی برنامه‌ی خود از آن ها استفاده کنید.
- فایل `positive-words.txt` را در پوشه‌ی فایل اجرایی کدتان قرار دهید. این فایل هنگام اجرای آزمون های خودکار در کنار فایل اجرایی حاصل از کامپایل برنامه‌ی شما قرار می گیرد. اما ممکن است محتوای آن هنگام تحویل تغییر کند.
- هر خط ورودی که کلمه‌ی اول آن یکی از دستورهای بالا نباشد، متن یادداشت حساب می شود و باید به یادداشت های مربوط به روز فعلی اضافه شود.

- دستور نشان دادن بهترین روز که در این تمرین آمده، نوعی تحلیل احساسات⁴ است که موضوعی بسیار پیچیده است و روشی که برای آن در این تمرین داده شده است بسیار ساده و از تکنیک‌های دنیای واقعی دور است. اگر به این موضوع علاقه دارید، پیشنهاد می‌کنیم [این لینک](#) را مطالعه کنید.

⁴ Sentiment Analysis