

باسمه تعالی
هجدهمین دوره‌ی آموزشی المپیاد کامپیوتر
امتحان عملی اصلی سوم

شنبه ۲ شهریورماه ۱۳۸۷

وقت: ۵ ساعت

نصیری شرق

توضیحات

- در خط اول هر یک از برنامه‌های تان نام خود را با فرمت روبه‌رو قرار دهید: `//name: YourNameHere`
مثال: `//name: Hadi Saaee`
دقت کنید که اولین فاصله‌ی خالی این سطر، بعد از : می‌باشد.
- توصیه می‌شود قبل از ترک جلسه‌ی امتحان حتماً یک‌بار (دیگر) تست‌های برگه^۱ را به برنامه‌های تان بدهید.
- وقت امتحان پنج ساعت است و پس از اتمام وقت، برق سالن قطع خواهد شد. حداقل زمان حضور در جلسه نیز یک ساعت می‌باشد. در صورتی که در حین امتحان برق قطع شد، سر جای خود نشسته و در کمال آرامش به حل تئوری و الگوریتم مسئله بپردازید. ضمناً پیش‌نهاد می‌شود به‌صورت مرتب برنامه‌ی خود را ذخیره^۲ کنید تا در صورت رفتن برق، اطلاعات زیادی را از دست ندهید.
- در ده دقیقه‌ی ابتدایی به هیچ سوالی پاسخ داده نمی‌شود. در این زمان همه‌ی سوالات را بخوانید. سوالات شما (صرفاً جهت رفع ابهام) در یک ساعت اول جواب داده خواهند شد. پس از این مدت به هیچ سوالی پاسخ داده نمی‌شود.
- نام کاربری شما در این امتحان `exam3` و رمز عبور شما `keikosandis` می‌باشد. در طول امتحان فقط از همین کاربر استفاده کنید.
- برنامه‌های خود را با نام‌های `memol.cpp` و `workers.cpp` نوشته و در شاخه‌ی `home` کامپیوتر خودتان (`/home/exam3/`) ذخیره کنید. از قراردادن پسوندی نظیر `c`، `cc` و `txt` و غیره برای فایل اصلی برنامه‌های خود و نیز استفاده از حروف بزرگ یا ذخیره‌سازی در سایر شاخه‌ها جداً خودداری فرمائید. دقت کنید که برای این کار لازم نیست شما `mkdir` کنید.
- برنامه‌ی شما دارای محدودیت زمانی و حافظه می‌باشد. بدیهی است در صورت رعایت نشدن این محدودیت‌ها، برنامه‌ی شما کلاً پا خواهد شد. دقت کنید که کامپایلر مورد استفاده در تصحیح، ممکن است تا ۳ مگابایت^۳ به حافظه مصرفی برنامه شما بیافزاید. از این رو، شما نباید روی تمام سقف مجاز حافظه مصرفی حساب کنید. برای مثال اگر حداکثر حافظه ۳۲ مگابایت است، ممکن برنامه‌ی شما با مصرف ۳۰ مگابایت نیز کشته شود.
- مجدداً یادآوری می‌شود که اشتباه در نام‌گذاری برنامه یا نحوه خواندن و نوشتن ورودی و خروجی، اشتباه در ذخیره‌سازی و قراردادن نام برنامه‌نویس در بالای برنامه‌ها، نمره‌ی صفر را در پی خواهند داشت. بعداً هیچ‌گونه اعتراضی در این زمینه پذیرفته نخواهد شد.

^۱ Sample inputs

^۲ Save

^۳ بسته به سرفایل‌های الصاق‌شده و سایر عوامل

۱ مُمَل (memol) ۱۰۰ نمره

مُمَل یک شخصیت کوچولو و محبوب کارتونی است.^۴ یک روز صبح، مُمَل می‌خواست به خانه‌ی دخترمهربان^۵ برود؛ اما در میانه‌ی راه متوجه شد که باران شب گذشته، رودخانه‌ای به عرض d واحد را در مسیر همیشگی‌اش به وجود آورده است. برای راحتی کار، رودخانه را یک بازه‌ی باز روی محور x ها (افقی) بگیرد که از نقطه صفر (جایی که مُمَل هست) تا نقطه‌ی d روی محور را پوشانده است.

خوش‌بختانه در این رودخانه n عدد سنگ ریز و محکم در فاصله‌های s_1 تا s_n از لبه‌ی رودخانه (نقطه‌های s_1 تا s_n محور x ها) قرار دارند. هم‌چنین m قورباغه‌ی چاق نیز در نقاط f_1 تا f_m قرار گرفته‌اند.

پس از کمی جست‌وجو، برای عبور از این رودخانه، مُمَل یک تکه چوب دراز به طول k پیدا کرد. می‌دانیم با استفاده از این چوب و سنگ‌هایی که درون رودخانه وجود دارند، اگر مُمَل در نقطه‌ی x باشد و یک سنگ در فاصله‌ی t از آن نقطه قرار داشته باشد و $t \leq k$ ، آن وقت مُمَل می‌تواند چوب‌اش را دراز کرده و روی سنگ بگذارد و با حرکتی شبیه قهرمان‌های پرش با نیزه^۶ به نقطه‌ای با فاصله‌ی $2t$ از x ، در راستای سنگ برود. یعنی اگر سنگ جلوی (سمت راست) x بود (نقطه‌ی $x+t$ در آن صورت مُمَل می‌تواند به نقطه‌ی $x+2t$ برود و اگر سنگ سمت چپ x بود (نقطه‌ی $x-t$) در آن صورت مُمَل به خانه‌ی $x-2t$ می‌رود.

توجه کنید که چون سنگ‌ها به شدت لغزنده هستند، مُمَل فقط می‌تواند روی قورباغه‌ها و زمین بایستد! هم‌چنین چون چوب خیلی سنگین است، مُمَل نمی‌تواند روی زمین (از نقطه‌ی صفر به سمت چپ) راه برود، اما می‌تواند با چند پرش رفت و برگشت در نقطه‌ای عقب‌تر از نقطه‌ی اولیه‌اش (مثلاً -2) قرار بگیرد. و نیز می‌دانیم فقط سنگ‌ها (و نه زمین) می‌توانند به عنوان تکیه‌گاه (مرکز) پرش قرار بگیرند.^۷

هدف نهایی مُمَل نیز قرار گرفتن در نقاط d یا جلوتر از آن (بیش‌تر از d) است که در آن صورت چوب را رها کرده و به سمت خانه‌ی دختر مهربان می‌دود. اما از آن‌جا که هر پرش (با چوب، به مثابه نیزه!) برای مُمَل، یک دقیقه طول می‌کشد، مُمَل می‌خواهد در کم‌ترین زمان ممکن از رودخانه عبور کند. به مُمَل کمک کنید و برنامه‌ای بنویسید که با خواندن d ، تعداد و محل سنگ‌ها و قورباغه‌ها و نهایتاً طول چوبش (k) مشخص کند که مُمَل حداقل پس از چند دقیقه به آن طرف رودخانه می‌رسد.

ورودی

در سطر اول ورودی، به ترتیب از چپ به راست چهار عدد d (عرض رودخانه)، سپس n (تعداد سنگ‌ها)، بعد m (تعداد قورباغه‌ها) و نهایتاً k (طول چوب مُمَل) آمده است.

در سطر دوم ورودی n عدد (نه لزوماً مرتب) آمده است که هر کدام محل یکی از سنگ‌ها را نشان می‌دهد.

در سطر سوم ورودی m عدد (نه لزوماً مرتب) آمده است که هر کدام محل یکی از قورباغه‌ها را نشان می‌دهد.

خروجی

در صورتی که مُمَل می‌تواند بالاخره از رودخانه عبور کند، در یک سطر کم‌ترین تعداد پرش لازم را بنویسید. در غیر این صورت اگر هرگز مُمَل نمی‌تواند به نقطه‌ی d یا نقاط جلوتر از آن برسد، در تنها سطر خروجی عبارت No Solution (دقیقاً با N و S بزرگ) را بنویسید.

محدودیت‌ها

- تمام اعداد ورودی صحیح و مثبت‌اند.

^۴ شاید شما یادتون نیاد!

^۵ شخصیت محبوب مُمَل

^۶ ولی به جای از صفر تا ۹۰ درجه، از صفر تا ۱۸۰ درجه!

^۷ چون نوک چوب تیز است، در زمین نرم و نیز بدن قورباغه‌ها (اوووج!) فرو می‌رود!

- تمام سنگ‌ها و قورباغه‌ها در نقاط صحیح داخل رودخانه (بازه‌ی $[1, d - 1]$) قرار دارند.
- هیچ دو سنگ، دو قورباغه یا یک سنگ و یک قورباغه‌ای در یک مکان قرار ندارند.
- $1 \leq n \leq 2000$.
- $1 \leq m \leq 50$.
- $1 \leq k \leq 50$.
- همواره $1 \leq d \leq 2000000$ و در ۵۰ درصد تست‌ها، $1 \leq d \leq 2000000000$.
- زمان مجاز هر اجرا ۲ ثانیه و حافظه‌ی مصرفی مجاز حداکثر ۶۴ مگابایت می‌باشد.

ورودی و خروجی نمونه

Sample Input	Sample Output
4 2 1 1 1 3 2	2

Sample Input	Sample Output
4 1 1 2 1 2	No Solution

۲ کارگرها (workers) ۱۰۰ نمره

۳ تعمیرکار ماشین در مرکز «بهینه‌سازی مصرف سوخت» کار می‌کنند. این ۳ تعمیرکار W_1 ، W_2 و W_3 نام دارند. امروز صبح، در جلوی این مرکز، n خودروی سواری صف کشیده‌اند تا به‌ترتیب موتورشان تنظیم شود. می‌دانیم تنظیم کردن موتور هر خودرو برای هر یک از ۳ تعمیرکار همواره دقیقاً یک دقیقه طول می‌کشد منتهی دستمزدی که تعمیرکار W_i برای تنظیم خودروی C_j دریافت می‌کند برای i ها و j های مختلف متفاوت بوده و در یک جدول داده شده است. نکته‌ی جالب این مرکز، رابطه‌ی حسادت (یا به عبارت به‌ترتیبی) میان تعمیرکارهاست! بدین صورت که تعمیرکار W_i ($1 \leq i \leq 3$) تنها در صورتی (در ابتدای یک دقیقه) حاضر می‌شود موتور یک خودرو را تعمیر کند که تعمیرکارهای با اندیس کمتر (در صورت وجود) نیز در آن دقیقه مشغول باشند. به عنوان مثال امکان ندارد که در یک دقیقه فقط تعمیرکارهای W_1 و W_3 مشغول کار باشند؛ چرا که شرط کار کردن تعمیرکار W_3 ، فعال بودن هر دو W_1 و W_2 است. هم‌چنین برای احترام صف و رعایت ترتیب حضور و قرارگیری خودروها در صف، برای هر دو خودروی C_i و C_j (که $1 \leq i \leq j \leq n$)، موتور خودروی C_i باید همواره در زمانی زودتر یا همزمان با موتور خودروی C_j تنظیم شود. به عبارت دیگر، در ابتدای هر دقیقه، $1 \leq k \leq 3$ خودرو از ابتدای صف وارد مرکز شده و (با چینی می‌کنیم) توسط تعمیرکارهای اول تا k ام در یک دقیقه موتورشان تنظیم می‌شود.

با این وصف، از شما خواسته شده تا برنامه‌ای بنویسید که برنامه‌ریزی مناسبی برای کارکردن تعمیرکارها و انتساب خودروها به آن‌ها ارائه کند تا با دریافت کم‌ترین دستمزد (و نه لزوماً در کوتاه‌ترین زمان) موتور تمام n خودرو تنظیم شود. دقت کنید که زمان پایان تمام کارها می‌تواند بین $\lceil \frac{n}{3} \rceil$ و n دقیقه باشد و اصلاً هم برای ما اهمیتی ندارد!

ورودی

در سطر اول ورودی، n ، تعداد ماشین‌های حاضر در صف آمده است. در n سطر بعدی (سطرهای $2 \leq i \leq n+1$ ورودی)، در سطر i ام، ۳ عدد آمده است که به‌ترتیب (از چپ به راست) دست‌مزد دریافتی تعمیرکار اول، دوم و سوم برای تنظیم موتور خودروی $(i-1)$ ام است. فرض کنید خودروها به‌ترتیب قرارگیری‌شان در صف در ورودی آمده‌اند و اولین خودرو در صف^۸ خودرویی است که دست‌مزدهای آن در سطر دوم ورودی نوشته شده است.

خروجی

در تنها سطر خروجی کم‌ترین دست‌مزد دریافتی توسط تعمیرکارها برای تنظیم موتورهای n خودروی ورودی را بنویسید.

محدودیت‌ها

- $0 \leq n \leq 200,000$ و تمام دست‌مزدها صحیح، مثبت و کوچک‌تر از $200,000$ هستند.
- زمان مجاز هر اجرا ۲ ثانیه و حافظه‌ی مصرفی مجاز حداکثر ۶۴ مگابایت می‌باشد.

ورودی و خروجی نمونه

Sample Input	Sample Output
5 9 2 1 1 2 3 2 9 9 8 7 2 4 3 3	10

^۸ که الزاماً در اولین دقیقه وارد مرکز می‌شود ولی الزاماً توسط تعمیرکار اول تنظیم نمی‌شود!

