

افسانه‌ی مخزن گمشده



در سرزمین «گیت‌موریا»، استاد باستانی دیوارِ نسخ می‌گوید: «شاگرد دانا! اگر می‌خواهی دروازه‌های CI/CD را بگشایی، باید لوحی بسازی که همه‌ی جادوها را خودکار اجرا کند. نام لوح تو باید دقیقاً این باشد: HW0-0-<std_id>.sh. فقط همین یک لوح تحویل بده. در یک دایرکتوری کاملاً خالی آن را اجرا کن تا قدرتش آشکار شود.»

فصل اوّل: بیداری مخزن

شاگرد باید در دنیای خالی (پوشه‌ای تهی) نفس بکشد و با ورد «git init» روح گیت را فرا بخواند. سپس نام و ایمیل خویش را برای این مخزن تعیین کند تا هر ردّ قدمش در تاریخ بماند:

- user.name: «Workshop Student»
- user.email: «student@example.com»

او سه اثر نخستین پدید می‌آورد:

- سنگ‌نوشت «src/app.py» که در آغاز فقط یک پیام ساده چاپ می‌کند: hello
- دفتر «docs/notes.md» با واژه‌ی آغازین: Notes
- طومار «CHANGELOG.md» با سرفصل: Changelog

سپس نخستین مُهر در تاریخچه حک می‌شود: پیام کامیت باید دقیقاً «chore: initial commit» باشد. (نگهبانان تاریخچه روی دقت حروف و فاصله‌ها وسواس دارند!)

فصل دوم: شاخه‌ی درودی

شاگرد، شاخه‌ای می‌آفریند با نام «feature/greet». در این شاخه، بر سنگ‌نوشت برنامه، جادوی «سلام» می‌نگارد: تابعی به نام greet که پیام «Hello» را بر زبان می‌آورد. پس از افزودن این جادو، کامیتی بر دل تاریخ می‌نشیند با پیام دقیق: «feat: add greet».

فصل سوم: ادغام دو زمان

اکنون شاگرد باید به شاخه‌ی اصلی بازگردد—نامش «main» است (اگر در جهان شاگرد «master» باشد، پذیرفته است). او فرمان ادغامی می‌خواند که تاریخ را بافته نگه دارد: مرج بدون «fast-forward». پیام این ادغام باید دقیقاً این باشد: «merge: feature/greet». این مرج باید حقیقتاً «مرج» باشد؛ تاریخ باید نشان دهد که این گره، دو پدر دارد—نه این‌که ساده در یک خط حل شود.

فصل چهارم: آزمون صبر—لوپ صدگانه

دروازه‌ی CI/CD تنها با «ریتم» باز می‌شود. شاگرد باید طومار «CHANGELOG.md» را ۱۰۰ بار تکان دهد؛ هر بار، یک خط تازه بیافزاید، درست به این ریخت: i Auto line که در آن از ۱ تا ۱۰۰ بالا می‌رود و هیچ عددی جا نمی‌ماند. پس از هر افزودن، باید مُهری جداگانه در تاریخ بزند: chore: auto commit #i این ۱۰۰ مُهر باید یکی‌یکی، پشت‌سر هم، با شماره‌های دقیق، بی‌کم‌وکاست نقش ببندند. (نگهبانان دآوری، کمین کرده‌اند؛ اگر یکی از شماره‌ها جا بماند یا شکل پیام‌ها حتی اندکی تغییر کند، دروازه قفل می‌ماند.)

فصل پنجم: خاتم نسخ

وقتی جادوهای پیشین به کمال رسید، شاگرد باید «خاتم» بسازد: تگی «annotated» با نام «v1.0». پیام این خاتم آزاد است، اما «annotated» بودنش واجب. این مهر، نشان پایان مناسک است.

سوگند نگهبانان

نگهبانان، لوح تو را تنها زمانی می‌پذیرند که:

۱. نام لوح دقیقاً «HWO-0-<std_id>.sh» باشد (مثلاً HWO-0-403100000.sh).

۲. لوح در پوشه‌ای کاملاً خالی اجرا شود.

۳. همه‌ی نام‌ها و پیام‌ها دقیقاً مطابق این روایت باشند:

- chore: initial commit
- feature/greet (نام شاخه)
- feat: add greet
- merge: feature/greet (ادغام بدون fast-forward)
- Auto line i (در محتوا، برای $i = 1..100$)
- chore: auto commit #i (برای همان i)
- v1.0 (تگ annotated)

۴. در پایان، نگهبان تک‌تک مهرها را می‌شمرد:

- وجود مرج واقعی (نه fast-forward) با پیام دقیق
- وجود ۱۰۰ کامیت پیاپی با پیام‌های دقیق و «CHANGELOG.md» با ۱۰۰ خط دقیق
- وجود تگ annotated با نام «v1.0»

۵. در لوح، اگر کامنتی باشد، باید به زبان انگلیسی باشد (چون زبان کاهنان قاعده دارد).

فرجام: نمره‌ی سرنوشت

در تالار داوری، استاد، همه‌ی لوح‌ها را در یک صندوق می‌ریزد و یکی یکی می‌خواند. هر لوح بر پایه‌ی

«std_id» شناخته می‌شود و نمره‌ای از ۱۰۰ می‌گیرد. لوح کامل، در دروازه را می‌گشاید و نقش می‌بندد:
«Score: 100 / 100 — دروازه‌های CI/CD بر تو گشوده شد.»

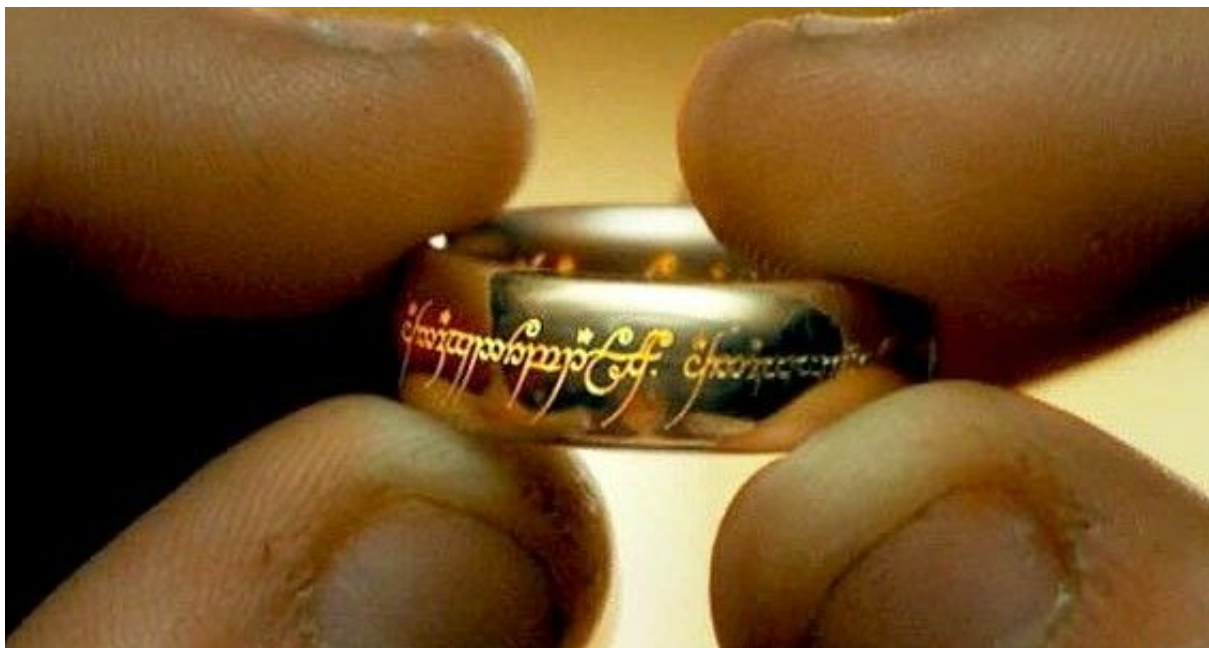
اگر در هر گامی لغزشی رخ دهد—پیامی کم‌وبیش متفاوت، شاخه‌ای غایب، یک مرج ساده به‌جای مرج واقعی، یا حتی یک شماره‌ی جافتاده در حلقه—نگهبانان با زنگی بلند هشدار می‌دهند و نمره فرو می‌نشیند.

یادداشت پایانی استاد باستانی دیوارِ نسخ

«شاگرد! این تمرین را چون آیینی انجام بده، نه چون تکلیف. با هر مُهر، احترامِ تاریخچه را نگاه دار؛ با هر شاخه، قصه‌ای مستقل بساز؛ با هر ادغام، روایت‌ها را به‌درستی پیوند بده. آنگاه که خاتم v1.0 را نهادی، بدانی که نخستین پله را استوار برفتی.»

رمز حلقه

- محدودیت زمان 0.5 ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت



هنگامی که گاندالف متوجه شد حلقه‌ای که در دست بیلبو بگینز است در واقع همان حلقه تاریکی است که توسط سائورون ساخته شده، برای نابود کردن آن شروع به مطالعه درباره‌اش کرد.

در حین مطالعات، او متوجه شد که هنگامی که حلقه در آتش قرار می‌گیرد، نوشته‌هایی بر آن ظاهر می‌شوند. او باید **رمز این متن** را بگشاید تا راهی برای نابود کردن سائورون بیابد.

دستورالعمل رمزگشایی متن به صورت زیر است:

بر حلقه، کلماتی وجود دارد که با کاراکتر - از یکدیگر جدا شده‌اند. دو یا چند کلمه را **هم‌ریشه** می‌نامیم اگر مجموعه‌ی حروف آن‌ها با هم یکسان باشند. به عنوان مثال سه کلمه eat ، ate و tea با یکدیگر هم‌ریشه‌اند.

گاندالف باید کلمات هم‌ریشه را تشخیص دهد و در گروه‌های جداگانه قرار دهد. از آنجا که او از شدت

مطالعه زیاد خسته شده است، از شما کمک می‌خواهد تا با نوشتن یک برنامه، به او در رمزگشایی حلقه کمک کنید.

ورودی

ورودی شامل تنها یک خط است که در آن رشته‌ای از کلمات قرار دارد. این کلمات با کاراکتر - از یکدیگر جدا شده‌اند.

شرایط زیر برقرار است:

$$0 \leq m \leq 200$$

$$0 \leq n \leq 50$$

$$1 \leq k \leq 10$$

که در آن:

- m طول کل رشته است، n تعداد کلمات است، k تعداد گروه‌بندی‌های کلمات هم‌ریشه است.

خروجی

در خروجی، در هر خط باید کلمات هم‌ریشه را چاپ کنید. هر کلمه بین دو علامت ستاره (*) قرار می‌گیرد. برای مثال: *word*

نکات:

- ترتیب چاپ کلمات باید مطابق با ترتیب اولین وقوع آن‌ها در رشته ورودی باشد.
- اگر کلمه‌ای چند بار تکرار شده باشد، فقط بار اول آن در نظر گرفته می‌شود.

نمونه ورودی

bat-eat-tea-tan-ate-nat-tea

نمونه خروجی

bat
eat *tea* *ate*
tan *nat*

نبرد میناس تیریث

- محدودیت زمان 1 ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

آتش جنگ در سرزمین «گندور» زبانه می‌کشد. شهر «مینس تیریث» در حال از دست رفتن است. حاکم شهر مجنون شده‌است و جنگجویان درحال شکست هستند. درخواست کمک این سرزمین توسط یاران قدیمی، سرزمین «روهان» پاسخ داده می‌شود.



جنگجویان روهان به سرعت به میدان نبرد می‌تازند و در این حین متوجه مقاربت مردم گندور به شکست می‌شوند. «سارون» دشمن شیطانی درحال برنده‌شدن است و آن‌ها باید این نبرد را سریع‌تر تحت سیطره خود دریاورند.

هر کدام از سربازان هردو جبهه، تا قبل از برخورد به یک‌دیگر، یک مختصات اولیه (در فضای یک‌بعدی) و یک سرعت ثابت دارند. نکات زیر در برخورد سربازها را در نظر بگیرید:

- اگر دو سرباز گوندور به هم برخورد کنند به زمین می‌افتند و نیاز به کمک دارند.
- اگر دو سرباز سارون به هم برخورد کنند به زمین می‌افتند و فرصت خوبی برای کشتن هردوی آن‌ها

است.

- اگر یک سرباز سارون و یک سرباز گوندور به هم برخورد کنند، سرباز گوندور نیاز به کمک دارد.
- توجه کنید که هرگاه هر دو نفری که به هم برخورد کنند، هر دو از فضای تک‌بعدی تحت آزمایش ما خارج می‌شوند.
- هرگاه بیش از دو سرباز در آن واحد به هم برخورد کنند، نخست دو سربازی که در ابتدا مختصات منفی‌تری دارند را لحاظ می‌کنیم.
- اهمیت هر سه نوع برخورد به یک مقدار است.

جنگجویان روهان به دلیل حساسیت بالا باید به‌موقع به هرکدام از برخوردها برسند. بخاطر همین «گاندالف» به کمک جادوی خود در بدو ورود روهانی‌ها به صحنه نبرد، سرعت و مختصات تمام سربازان را به شما می‌دهد تا شما ترتیب برخوردها را محاسبه کنید و به روهانی‌ها بدهید و در این صورت آن‌ها می‌توانند در این جنگ پیروز شوند.

قابل توجه می‌باشد که به دلیل حساسیت بالای نبرد باید این کار در اسرع وقت اجرایی شود.

ورودی

نخست عدد n که نشان‌دهنده تعداد سربازان است وارد می‌شود و سپس در n خط بعدی، در هر خط دو عدد که عدد نخست نشان‌دهنده x_i و عدد بعدی نشان‌دهنده v_i است.

$$0 \leq n \leq 10000$$

$$-2n \leq x_i, v_i \leq 2n$$

خروجی:

در هر خط x_i, x_j هایی که باهم برخورد می‌کنند نمایش داده می‌شود. ترتیب خط‌ها نسبت به زمان برخورد و در صورت هم‌زمانی x_i کوچکتر مشخص می‌شود. همچنین در یک خط نیز، اول عدد کوچکتر و

سپس بزرگتر نوشته می‌شود.

ورودی نمونه ۱:

6
1 2
0 3
5 4
6 2
-1 5
2 -1

خروجی نمونه ۱:

1 2
-1 0
5 6

▼ Details

تحلیل تست

هم‌اکنون این تست را تحلیل می‌کنیم. در لحظه شروع جفت متحرک‌های زیر پس از مقدار زمان گفته‌شده به هم برخورد می‌کنند.

لحظه برخورد	متحرک دوم	متحرک نخست
0.5s	0	1-
0.67s	1	1-
0.5s	2	1-
6s	5	1-

2.33s	6	1-
1s	1	0
0.5s	2	0
never	5	0
6s	6	0
0.33	2	1
never	5	1
never	6	1
never	5	2
never	6	2
0.5	6	5

نخست، 1, 2 برخورد می‌کنند، سپس این دو از فضا خارج شده و سپس دو جفت باقی‌مانده 0, 1- و 5, 6 در یک آن برخورد می‌کنند، ولیکن بخاطر مختصات اولیه کمتر، گزینه نخست، زودتر چاپ می‌شود.

ورودی نمونه ۲:

4
-2 2
-1 1
1 -1
2 -2

خروجی نمونه ۲:

-2 -1

1 2

▼ Details

تحلیل تست

مشخص است هر ۴ متحرک همزمان باهم به مختصات ۰ می‌سند، لیکن بدلیل مختصات اولیه کمتر نخست زوج مشخص شده چاپ می‌شود و هر دو از فضا خارج می‌شوند و سپس گزینه‌های بعدی بررسی می‌شوند. <details/>

پیمان بیداری حلقه

- محدودیت زمان 1 ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت



در اعماق کوه‌های مه‌آلود و پس از نابودی حلقه‌ی یگانه، هنوز سایه‌هایی مبهم بر سرزمین میانه گسترده است. اکنون افسانه‌ای کهن بازگو می‌کند: یکی از حلقه‌های قدرت - «حلقه‌ی سنجش» - هنوز درون دخمه‌های سنگی نهفته مانده است؛ حلقه‌ای که تنها با آزاد شدن جریانی از انرژی ناب بیدار می‌شود و تاریکی را می‌شکند.

شما، وارث امید اهالی میانه‌سرزمین، مقابل صفی از سنگ‌های انرژی ایستاده‌اید. از میان آن‌ها باید کوتاه‌ترین زنجیره‌ی ممکن را انتخاب کنید؛ بخشی پیوسته از این سنگ‌ها که مجموع نیروی‌شان حداقل به سطحی برسد که حلقه را بیدار کند. اگر چنین زنجیره‌ای بیابید، شاید سپیده‌دم بر سرزمین میانه بتابد؛ و اگر نه، حلقه در خواب مرگ‌آور خود باقی خواهد ماند.

یک آرایه مثبت از اعداد طبیعی و یک عدد طبیعی S داده شده است. طول کمینه زیرآرایه پیوسته‌ای را

پیدا کنید که مجموع عناصرش بزرگتر مساوی S شود.

ورودی

نخست عدد S که نشان‌دهندهٔ عدد هدف است وارد می‌شود و سپس در خط بعدی عدد n که نشان‌دهنده تعداد اعضا آرایه است داده می‌شود، و در آخرین خط n عدد طبیعی که ناصر آرایه هستند و با فاصله از هم جدا شده اند داده می‌شود.

$$1 \leq S \leq 10^8$$

$$1 \leq n \leq 10^5$$

$$1 \leq arr[i] \leq 10^8$$

خروجی:

در خروجی تعداد عضو کوچکترین زیرآرایه ای که چنین شرایطی را داشته باشد، نمایش دهید و اگر چنین زیرآرایه ای وجود نداشت، مقدار 0 را خروجی دهید.

ورودی نمونه 1:

```
7
6
2 3 1 2 4 3
```

خروجی نمونه 1:

```
2
```

ورودی نمونه 2:

100

4

1 2 3 4

خروجی نمونه 2:

0

حلقه‌ی آینه‌ای

- محدودیت زمان 0.5 ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت



گاندالف دوباره با حلقه‌ای دیگر روبه‌رو شده است. این بار او متوجه می‌شود که حلقه دارای خاصیتی *آینه‌ای* است؛ یعنی اگر دنباله‌ای از اعداد را روی آن حک کنیم، ممکن است با نگاهی از دو سوی حلقه، دنباله به صورت یکسان دیده شود.

به دنباله‌ای از اعداد می‌گوییم **آینه‌ای (پالیندروم)** اگر از ابتدا و انتها یکسان خوانده شود. به عنوان مثال، دنبالهٔ

[1, 2, 3, 2, 1]

آینه‌ای است، ولی

$$[1, 2, 3, 4, 1]$$

آینه‌ای نیست.

اما گاندالف متوجه می‌شود که اگر تمام اعضای برابر با یک عدد خاص را از دنباله حذف کند، ممکن است دنباله باقی‌مانده آینه‌ای شود! او می‌خواهد بداند آیا می‌تواند با حذف همه رخ داده‌های یک عدد ثابت (یا بدون حذف هیچ عددی)، دنباله را آینه‌ای کند یا خیر.

به عنوان مثال:

- دنباله $[1, 2, 3, 2, 1]$ بدون حذف هیچ عددی آینه‌ای است.
- دنباله $[1, 2, 3, 1]$ با حذف همه عددهای 3 تبدیل به $[1, 2, 1]$ می‌شود که آینه‌ای است.
- اما دنباله $[1, 2, 3, 4]$ با حذف هیچ عددی آینه‌ای نخواهد شد.

گاندالف از شما می‌خواهد برنامه‌ای بنویسید که مشخص کند آیا این خاصیت برای دنباله برقرار است یا نه.

ورودی

ورودی شامل چندین تست‌کیس است.

در خط اول عددی طبیعی (T) آمده که تعداد تست‌کیس‌ها را مشخص می‌کند. برای هر تست‌کیس:

- در خط اول عدد n طول دنباله.
- در خط دوم n عدد طبیعی به شکل زیر آمده‌اند.

$$a_1, a_2, \dots, a_n$$

$$1 \leq T \leq 10^4,$$

$$1 \leq n \leq 2 \times 10^5,$$

$$1 \leq a_i \leq 10^9,$$

$$\sum n \leq 2 \times 10^5$$

خروجی

برای هر تست کیس، در یک خط چاپ کنید:

- "YES" اگر دنباله با حذف صفر یا یک نوع عدد خاص، آینه‌ای می‌شود.
- "NO" در غیر این صورت.

مثال

ورودی نمونه ۱

```
3
5
1 2 3 2 1
4
1 3 2 1
4
1 2 3 4
```

خروجی نمونه ۱

```
YES
YES
NO
```

توضیح نمونه

- در تست اول، دنباله از ابتدا آینه‌ای است.
- در تست دوم، با حذف تمام عددهای 3، دنباله به $[1, 2, 1]$ تبدیل می‌شود که آینه‌ای است.
- در تست سوم، هیچ عددی وجود ندارد که با حذف آن دنباله آینه‌ای شود.

نکات

- حذف باید شامل تمام رخداد های یک عدد باشد (نه فقط یک مورد).
- اگر دنباله از ابتدا آینه‌ای است، پاسخ "YES" است.
- لازم نیست بدانید کدام عدد حذف می‌شود، فقط وجود چنین عددی کافی است.