

بازی کلاسی

یک دبیرستان برجسته در شهر تاشکند سالگرد تأسیس خود را با برگزاری یک بازی بین دانش‌آموزان و معلمان جشن می‌گیرد. تعداد N دانش‌آموز در کلاس درس نشسته‌اند که از 0 تا $N - 1$ شماره‌گذاری شده‌اند. هر دانش‌آموز یک تکه کاغذ در دست دارد که شامل آرایه‌ای از اعداد صحیح است. در ابتدا هر کاغذ خالی است، بنابراین شامل یک آرایه‌ی تهی است.

دانش‌آموزان در حال انجام این بازی با مدیر مدرسه و M معلم که از 0 تا $M - 1$ شماره‌گذاری شده‌اند هستند. این بازی در M مرحله انجام می‌شود که از 0 تا $M - 1$ شماره‌گذاری شده‌اند. در مرحله‌ی j ، معلم j وارد کلاس درس می‌شود و رویدادهای زیر اتفاق می‌افتد:

- تعدادی دانش‌آموز (که ممکن است صفر هم باشند) دست خود را بالا می‌برند. تضمین می‌شود که در هر مرحله حداقل یک دانش‌آموز دست خود را بالا نمی‌برد و هر دانش‌آموز می‌تواند در کل بازی حداکثر یک بار دست خود را بالا ببرد.
- معلم به برگه‌هایی که دانش‌آموزان در حال حاضر در دست دارند نگاه می‌کند و می‌تواند برگه‌های دانش‌آموزانی را که در این مرحله دست خود را بالا نبرده‌اند اصلاح کند. برای هر یک از این دانش‌آموزان، معلم می‌تواند محتوای برگه‌ی آن‌ها را با یک آرایه‌ی جدید از اعداد صحیح (که می‌تواند تهی باشد) جایگزین کند. به طور کلی، هر عدد صحیح باید بین 0 و 63 باشد و آرایه می‌تواند حداکثر 63 عنصر داشته باشد.
- پس از این اصلاحات، معلم j کلاس را ترک می‌کند و دانش‌آموزان برگه‌های خود را بر اساس یک جایگشت مخفی P با هم عوض می‌کنند. یعنی هر دانش‌آموز i ($0 \leq i < N$) برگه‌ی خود را به دانش‌آموز $P[i]$ می‌دهد که در آن $P[0], P[1], \dots, P[N - 1]$ شامل N عدد صحیح متمایز از 0 تا $N - 1$ هستند. مقادیر P در تمام مراحل بازی ثابت هستند، اما معلمان و مدیر مدرسه آن‌ها را نمی‌دانند.

وقتی تمام M مرحله تمام شد و آخرین تبادل بر اساس P انجام شد، مدیر وارد می‌شود. مدیر فقط با نگاه کردن به برگه‌های دانش‌آموزان باید برای هر دانش‌آموز i ($0 \leq i < N$)، شماره مرحله‌ی j را که در آن دانش‌آموز i دست خود را بالا برده است تعیین کند یا نتیجه بگیرد که دانش‌آموز i هرگز دستش را بالا نبرده است.

معلمان نمی‌توانند با معلمان دیگر یا مدیر مدرسه ارتباط برقرار کنند، مگر از طریق آرایه‌هایی از اعداد صحیح که روی برگه‌ها نوشته شده است. هر معلم می‌داند که در چه مرحله‌ای وارد کلاس درس می‌شود.

وظیفه‌ی شما این است که برای معلمان و مدیر مدرسه، یک استراتژی طراحی و پیاده‌سازی کنید که به درستی تشخیص دهد که آیا و چه زمانی هر دانش‌آموز دست خود را بالا برده است. نمره‌ی شما به حداکثر طول آرایه‌ی نوشته شده روی تمام برگه‌ها بستگی دارد: هر چه حداکثر طول کوتاه‌تر باشد، نمره‌ی بالاتر یا مساوی حاصل می‌شود.

جزئیات پیاده‌سازی

شما باید دو تابع را فراخوانی کنید، یکی برای معلمان و یکی برای مدیر مدرسه.

تابعی که باید برای معلمان فراخوانی می‌کنید این است:

```
std::vector<std::vector<int>> process_step(
    int N, int M, int R,
    std::vector<int> T,
    std::vector<std::vector<int>> A)
```

- N : تعداد دانش‌آموزان.
 - M : تعداد مراحل و همچنین تعداد معلمان.
 - R : شماره مرحله‌ی فعلی (از 0 تا $M - 1$).
 - T : یک آرایه‌ی (احتمالاً تهی) شامل اندیس‌های دانش‌آموزانی که در این مرحله دست خود را بالا می‌برند، که به ترتیب صعودی مرتب شده‌اند.
 - A : آرایه‌ای به طول N که برگه‌ها را توصیف می‌کند و در آن $A[i]$ ($0 \leq i < N$) آرایه‌ای از اعداد صحیح روی برگه‌ای است که توسط دانش‌آموز i در ابتدای مرحله نگهداری می‌شود.
 - این تابع دقیقاً M بار در هر بازی، به ترتیب $R = 0, 1, \dots, M - 1$ فراخوانی می‌شود.
- این تابع باید یک آرایه‌ی B به طول N برگرداند که نشان‌دهنده‌ی برگه‌ها پس از اصلاحات معلم و با همان قالب A است.
- به ازای هر دانش‌آموز i که دستش را بالا برده است (یعنی i در T ظاهر می‌شود) برگه نباید تغییر کند، بنابراین $B[i] = A[i]$.
 - برای هر i با شرط $0 \leq i < N$ ، طول $B[i]$ نباید از 63 بیشتر شود و هر عدد صحیح در $B[i]$ باید بین 0 و 63 باشد.

تابعی که باید برای مدیر مدرسه فراخوانی کنید به شرح زیر است:

```
std::vector<int> determine_steps(
    int N, int M, std::vector<std::vector<int>> A)
```

- M ، N : مشابه مورد بالا.
 - A : آرایه‌ای به طول N که در آن $A[i]$ آرایه‌ای از اعداد صحیح روی برگه است و توسط دانش‌آموز i پس از تمام M مرحله نگهداری می‌شود.
 - این تابع دقیقاً یک بار در هر بازی، پس از آخرین فراخوانی تابع `process_step` فراخوانی می‌شود.
- این تابع باید یک آرایه‌ی D به طول N برگرداند. برای هر i با شرط $0 \leq i < N$ ، باید شرایط زیر را داشته باشد:
- اگر دانش‌آموز i در مرحله‌ی j دست خود را بالا برده باشد: $D[i] = j$
 - اگر دانش‌آموز i در کل بازی هرگز دستش را بالا نبرده باشد: $D[i] = -1$

برنامه‌ی شما مجاز به ذخیره یا انتقال هیچ اطلاعاتی از یک فراخوانی `process_step` به فراخوانی دیگر نیست، مگر از طریق مقدار بازگشتی `process_step`. اگر برنامه‌ی شما سعی در انجام این کار داشته باشد ممکن است امتیاز شما در CMS نادرست باشد و پس از پایان مسابقه کاهش یابد. توجه داشته باشید که ارزیاب ممکن است چندین نمونه از برنامه‌ی شما را به طور همزمان اجرا کند. این بدان معناست که فراخوانی‌های `process_step` و `determine_steps` از یک بازی، ممکن است در نمونه‌های مختلف اجرا شوند و فراخوانی‌های `process_step` و `determine_steps` از بازی‌های مختلف، ممکن است در یک نمونه اجرا شوند. هر نمونه شامل حداکثر 5 بازی است.

محدودیت‌ها

- $2 \leq N \leq 63$
- $1 \leq M \leq 63$
- هر دانش‌آموز در کل بازی **حداکثر یک بار** دستش را بالا می‌برد. به عبارت دیگر، برای هر دانش‌آموز i حداکثر یک مرحله‌ی j وجود دارد که در آن i دستش را بالا می‌برد.
- در هر مرحله، حداقل یک دانش‌آموز دست خود را بالا نمی‌برد.

زیرمسئله‌ها و امتیازدهی

Subtask	Score	Additional Constraints
1	4	$M = 1$
2	6	$N = 2$
3	9	$P[i] = i$ for all $0 \leq i \leq N - 1$.
4	25	At most one student raises their hand at each step.
5	56	No additional constraints.

برای هر نمونه‌ی آزمایشی، اگر مقدار برگشتی هر فراخوانی به تابع `process_step` نامعتبر باشد یا مقدار برگشتی هر فراخوانی به تابع `determine_steps` نادرست باشد، امتیاز شما برابر 0 می‌شود (که در CMS به صورت Output `isn't correct` گزارش می‌شود).

در غیر این صورت، فرض کنید C حداکثر طول هر آرایه در هر B برگردانده شده توسط `process_step` باشد. امتیاز یک نمونه در یک زیرمسئله با امتیاز S به صورت $S \times X$ محاسبه می‌شود که در آن X طبق جدول زیر به C بستگی دارد:

Condition	X
$C \leq 2$	1.00
$C = 3$	0.75
$C = 4$	0.55
$5 \leq C \leq 13$	$0.50 - 0.03 \cdot (C - 5)$
$14 \leq C \leq 63$	$0.19 \cdot \frac{64-C}{64-14} + 0.04$

مثال

سناریویی با $N = 4$ دانش‌آموز، $M = 2$ معلم و جایگشت $P = [0, 3, 1, 2]$ را در نظر بگیرید. در ابتدا تمام کاغذها خالی هستند.

ارزیاب ابتدا تابع زیر را فراخوانی می‌کند:

```
process_step(4, 2, 0, [1], [[], [], [], []])
```

دانش‌آموز 1 دستش را بالا برده است، بنابراین برگه‌اش قابل اصلاح نیست. معلم 0 تصمیم می‌گیرد که [10] را روی برگه‌ی دانش‌آموز 0 بنویسد، [1, 63, 4] را روی برگه‌ی دانش‌آموز 3 بنویسد، و برگه‌ی دانش‌آموز 2 را خالی بگذارد. برای انجام این کار، تابع باید $B = [[10], [], [], [1, 63, 4]]$ را برگرداند.

بعد از رفتن معلم 0 دانش‌آموزان برگه‌های خود را بر اساس P عوض می‌کنند. پس از تعویض، برگه‌ها $A = [[10], [], [1, 63, 4], []]$ هستند.

حالا ارزیاب تابع زیر را فراخوانی می‌کند:

```
process_step(4, 2, 1, [0, 3], [[10], [], [1, 63, 4], []])
```

دانش‌آموزان 0 و 3 دست خود را بالا برده‌اند، بنابراین برگه‌های آنها قابل اصلاح نیست. معلم 1 تصمیم می‌گیرد که [0, 40] را روی برگه‌ی دانش‌آموز 1 بنویسد و [1, 50] را روی برگه‌ی دانش‌آموز 2 بنویسد. برای انجام این کار، تابع باید $B = [[10], [0, 40], [1, 50], []]$ را برگرداند.

بعد از رفتن معلم 1، برگه‌ها دوباره طبق P عوض می‌شوند. بعد از تعویض، برگه‌ها $A = [[10], [1, 50], [], [0, 40]]$ هستند.

در نهایت ارزیاب تابع زیر را فراخوانی می‌کند:

```
determine_steps(4, 2, [[10], [1, 50], [], [0, 40]])
```

این تابع باید آرایه‌ی $D = [1, 0, -1, 1]$ را برگرداند زیرا دانش‌آموزان 0 و 3 دست خود را در مرحله‌ی 1 بالا برده‌اند، دانش‌آموز 1 دست خود را در مرحله‌ی 0 بالا برده است و دانش‌آموز 2 هرگز دستش را بالا نبرده است. در این مثال $C = 3$ است.

ارزیاب نمونه

قالب ورودی:

```
N M
Q[0] Q[1] ... Q[N-1]
P[0] P[1] ... P[N-1]
```

در اینجا Q آرایه‌ای به طول N است که در آن $Q[i] = j$ است، اگر دانش‌آموز i در طول مرحله‌ی j دست خود را بالا ببرد، و یا $Q[i] = -1$ است، اگر دانش‌آموز i در کل بازی هرگز دست خود را بالا نبرد.

قالب خروجی:

پس از هر بار فراخوانی تابع `process_step`، ارزیاب نمونه محتوای برگه‌ها را نمایش می‌دهد. فرض کنید K طول B و $L[i]$ طول $B[i]$ باشد (برای هر $0 \leq i < K$). ارزیاب نمونه B را با یک خط خالی در انتها به شکل زیر چاپ می‌کند:

```
K
L[0] B[0][0] B[0][1] ... B[0][L[0]-1]
L[1] B[1][0] B[1][1] ... B[1][L[1]-1]
:
L[K-1] B[K-1][0] B[K-1][1] ... B[K-1][L[K-1]-1]
```

سپس ارزیاب نمونه، برگه‌ها را بر اساس جایگشت P عوض می‌کند. اگر $K \neq N$ باشد، ارزیاب نمونه یک پیام خطا چاپ می‌کند و به جای آن خاتمه می‌دهد.

پس از فراخوانی `determine_steps`، خروجی ارزیاب نمونه به صورت زیر خواهد بود:

```
C H
D[0] D[1] ... D[H-1]
```

در اینجا H طول آرایه‌ی D است که توسط `determine_steps` برگردانده شده است.