

لعبة في القسم

تحتفل إحدى المدارس البارزة في طشقند بذكرى تأسيسها من خلال تنظيم مباراة بين الطلاب والمعلمين. يوجد N طالبًا جالسين في الفصل الدراسي، مرقمين من 0 إلى $N - 1$. يحمل كل طالب ورقة تحتوي على قائمة من الأعداد الصحيحة. في البداية، تكون كل ورقة فارغة، لذا فهي تحتوي على قائمة فارغة.

يلعب الطلاب اللعبة مع M معلمًا، مرقمين من 0 إلى $M - 1$ ، بالإضافة إلى مدير المدرسة. تلعب اللعبة على M خطوة، مرقمة أيضًا من 0 إلى $M - 1$.

في الخطوة j ، يدخل المعلم j إلى القسم وتحدث الأحداث التالية:

- يرفع بعض الطلاب (وربما لا أحد) أيديهم. ومن المؤكد أنه في كل خطوة يوجد طالب واحد على الأقل لم يرفع يده، ويجوز لكل طالب رفع يده مرة واحدة على الأكثر خلال اللعبة بأكملها.
- ينظر المعلم إلى الأوراق التي يحملها الطلاب حاليًا، بإمكانه تعديل الأوراق التي يحملها الطلاب الذين لم يرفعوا أيديهم في هذه الخطوة. بالنسبة لهؤلاء الطلبة، يمكن للمعلم استبدال محتويات أوراقهم بقائمة جديدة (قد تكون فارغة) من الأعداد الصحيحة. حيث يجب أن يكون كل عدد صحيح من القائمة محصورًا بين 0 و 63، ويمكن أن تحتوي القائمة على 63 عددًا على الأكثر.
- بعد هذه التعديلات، يغادر المعلم j ويقوم الطلاب بتبادل أوراقهم وفقًا لتبديل سري P . أي أن كل طالب i ($0 \leq i < N$) يعطي ورقته للطلاب $P[i]$ ، حيث $P[0], P[1], \dots, P[N - 1]$ هي N أعداد صحيحة مختلفة محصورة بين 0 و $N - 1$. تبقى قيم P ثابتة خلال جميع خطوات اللعبة، لكن المعلمين ومدير المدرسة لا يعرفونها.

بمجرد انتهاء جميع الخطوات M وإجراء التبادل الأخير وفقًا لـ P ، يدخل مدير المدرسة. وبالنظر فقط إلى الأوراق التي يحملها الطلاب، يجب على المدير أن يحدد، لكل طالب i ($0 \leq i < N$)، رقم الخطوة j التي رفع فيها الطالب i يده، أو أن يستنتج أن الطالب i لم يرفع يده أبدًا.

لا يمكن للمعلمين التواصل مع المعلمين الآخرين أو المدير، إلا من خلال قوائم الأعداد الصحيحة المكتوبة على الأوراق. يعرف كل معلم رقم الخطوة التي يخطوها عند دخوله الفصل الدراسي.

مهمتك هي ابتكار وتنفيذ استراتيجية للمعلمين والمدير تحدد بشكل صحيح ما إذا كان كل طالب قد رفع يده و في أي خطوة رفعها. ستعتمد علامتك على الحد الأقصى لطول أي قائمة مكتوبة على ورقة — كلما كان الطول الأقصى أصغر، كانت العلامة أعلى أو مساوية.

تفاصيل التنفيذ

يجب عليك تنفيذ إجراءات، أحدهما للمعلمين والآخر للمدير.

الإجراء الذي يجب عليك تنفيذه للمعلمين هو:

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

- N : عدد الطلاب.
- M : عدد الخطوات، وكذلك عدد المعلمين.
- R : رقم الخطوة الحالية (من 0 إلى $M - 1$).
- T : قائمة (قد تكون فارغة) تتكون من أرقام الطلاب الذين يرفعون أيديهم في هذه الخطوة، مرتبة بترتيب تصاعدي.
- A : قائمة بطول N تصف الأوراق، حيث $A[i]$ ($0 \leq i < N$) هي قائمة الأعداد الصحيحة الموجودة على الورقة التي يحملها الطالب i في بداية الخطوة.
- يتم استدعاء هذا الإجراء بالضبط M مرة في كل لعبة، بالترتيب $R = 0, 1, \dots, M - 1$.
- يجب أن يُرجع الإجراء قائمة B بطول N ، تمثل الأوراق بعد تعديلات المعلم، بنفس تنسيق A (format).
- بالنسبة لكل طالب i رفع يده (أي، i يظهر في T)، يجب ألا يتم تغيير الورقة، لذا $B[i] = A[i]$.
- بالنسبة لكل i بحيث $0 \leq i < N$ ، لا يجب لطول $B[i]$ أن يتجاوز 63، ويجب أن يكون كل عدد صحيح من $B[i]$ محصوراً بين 0 و 63.

الإجراء الذي يجب عليك تنفيذه للمدير هو:

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

- M ، N : كما هو مذكور أعلاه.
- A : قائمة بطول N ، حيث $A[i]$ هي قائمة الأعداد الصحيحة الموجودة على الورقة التي يحملها الطالب i بعد جميع خطوات M .
- يتم استدعاء هذا الإجراء مرة واحدة فقط في كل لعبة، بعد الاستدعاء الأخير لـ `process_step`.

يجب أن يُرجع الإجراء قائمة D بطول N . لكل i بحيث $0 \leq i < N$ ، يجب أن يتحقق ما يلي:

- $D[i] = j$ إذا رفع الطالب i يده خلال الخطوة j ، أو
- $D[i] = -1$ إذا لم يرفع الطالب i يده على الإطلاق طوال اللعبة.

لا يُسمح لبرنامجك بتخزين أو نقل أي معلومات من استدعاء لـ `process_step` إلى آخر، إلا من خلال قيمة الإرجاع لـ `process_step`. إذا حاول برنامجك القيام بذلك، فقد تكون نتيجتك في CMS غير صحيحة وقد يتم تخفيضها بعد انتهاء المسابقة. لاحظ أن نظام التصحيح قد يقوم بتنفيذ عدة مثيلات (instances) لبرنامجك في وقت واحد. وهذا يعني أن استدعاءات `process_step` و `determine_steps` من نفس اللعبة قد يتم تنفيذها في مثيلات مختلفة، وأن استدعاءات `process_step` و `determine_steps` من ألعاب مختلفة قد يتم تنفيذها في نفس المثيل. تتضمن كل حالة اختبار ما يصل إلى 5 لعبة.

القيود

- $2 \leq N \leq 63$
- $1 \leq M \leq 63$
- يرفع كل طالب يده مرة واحدة على الأكثر خلال اللعبة بأكملها. بعبارة أخرى، لكل طالب i ، هناك خطوة واحدة على الأكثر j يرفع فيها i يده.
- في كل خطوة، هناك طالب واحد على الأقل لا يرفع يده.

المهام الفرعية والنقاط

المهمة الفرعية	النقاط	القيود الإضافية
1	4	$M = 1$
2	6	$N = 2$
3	9	$P[i] = i$ لجميع $0 \leq i \leq N - 1$
4	25	يرفع طالب واحد على الأكثر يده في كل خطوة.
5	56	لا توجد قيود إضافية.

بالنسبة لكل حالة اختبار، تكون نتيجتك هي 0 (يتم الإبلاغ عنها على أنها Output isn't correct في CMS) إذا كانت القيمة المرجعة لأي استدعاء للإجراء process_step غير صالحة أو إذا كانت القيمة المرجعة لأي استدعاء للإجراء determine_steps غير صحيحة.

بخلاف ذلك، لنفترض أن C هو الحد الأقصى لطول أي قائمة في أي B يتم إرجاعها بواسطة process_step. عندئذٍ، تُحسب النتيجة لحالة اختبار في مهمة فرعية ذات نتيجة S على النحو التالي: $S \cdot X$ ، حيث يعتمد X على C وفقًا للجدول التالي:

الشرط	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

لنفترض سيناريو يضم $N = 4$ طالبًا، و $M = 2$ معلمًا، وترتيبًا $P = [0, 3, 1, 2]$. في البداية، تكون جميع الأوراق فارغة.

يقوم المصحح أولاً باستدعاء:

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

لقد رفع الطالب 1 يده، لذا لا يمكن تعديل ورقته. قد يقرر المعلم 0 كتابة [10] في ورقة الطالب 0، وكتابة [1, 63, 4] في ورقة الطالب 3، وترك ورقة الطالب 2 فارغة. للقيام بذلك، يجب أن تُرجع الإجراء $B = [[10], [], [], [1, 63, 4]]$.

بعد مغادرة المعلم 0، يتبادل الطلاب أوراقهم وفقًا لـ P . بعد التبادل، تصبح الأوراق كما يلي: $A = [[10], [], [1, 63, 4], []]$.

يقوم المصحح الآن باستدعاء:

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

لقد رفع الطالبان 0 و 3 أيديهما، لذا لا يمكن تعديل أوراق امتحانهما. قد يقرر المعلم 1 كتابة [0, 40] على ورقة الطالب 1 وكتابة [1, 50] على ورقة الطالب 2. للقيام بذلك، يجب أن تُرجع الدالة $B = [[10], [0, 40], [1, 50], []]$.

بعد مغادرة المعلم 1، يتم تبادل الأوراق مرة أخرى وفقاً لـ P . بعد التبادل، تصبح الأوراق $A = [[10], [1, 50], [], [0, 40]]$.

أخيراً، يقوم المصحح باستدعاء:

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

يجب أن تُرجع هذه الإجراء القائمة $D = [1, 0, -1, 1]$ نظراً لأن الطالبين 0 و 3 رفعاً أيديهم في الخطوة 1، والطالب 1 رفع يده في الخطوة 0، والطالب 2 لم يرفع يده مطلقاً. في هذا المثال، $C = 3$.

نموذج للمصحح

تنسيق الإدخال:

N M

Q[0] Q[1] ... Q[N-1]

P[0] P[1] ... P[N-1]

هنا، Q عبارة عن قائمة بطول N ، حيث $Q[i] = j$ إذا رفع الطالب i يده خلال الخطوة j ، أو $Q[i] = -1$ إذا لم يرفع الطالب i يده مطلقاً خلال اللعبة بأكملها.

تنسيق الإخراج:

بعد كل استدعاء لـ `process_step`، يقوم برنامج تصحيح العينة بإخراج محتويات الأوراق. لنفترض أن K هو طول B و $L[i]$ هو طول $B[i]$ (لكل $0 \leq i < K$). يقوم برنامج التصحيح النموذجي بطباعة B بالصيغة التالية، متبوعة بسطر فارغ:

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]

ثم يقوم برنامج التصحيح النموذجي بتبديل الأوراق وفقاً للتبديل P . إذا كان $K \neq N$ ، فإن برنامج التصحيح النموذجي يطبع رسالة خطأ وينتهي العملية بدلاً من ذلك.

بعد استدعاء `determine_steps`، يُخرج برنامج التصحيح النموذجي:

C H

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

هنا، H هو طول القائمة D التي تُرجعها الدالة `.determine_steps`.